Summer 2013

Oracle Cost Based Optimizer Correlations

Robert Apple
Regis University

ORACLE COST BASED OPTIMIZER CORRELATIONS

A THESIS

SUBMITTED ON 8th OF AUGUST, 2013

TO THE DEPARTMENT OF INFORMATION TECHNOLOGY

OF THE SCHOOL OF COMPUTER & INFORMATION SCIENCES

OF REGIS UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF MASTER OF SCIENCE IN

SOFTWARE ENGINEERING AND DATABASE TECHNOLOGIES

BY

Robert Apple

Robert Apple

APPROVALS

Ernest Eugster

Dr. Ernest Eugster, Thesis Advisor

Donald J. Ina

Don Ina, Faculty of Record

Nancy Birkenheuer

Nancy Birkenheuer, Program Coordinator

Abstract

Database systems use optimizers on queries to select execution pathways that are supposed to provide optimal performance. The Oracle database version of this technology is called the Cost Based Optimizer (CBO). Researchers have studied whether Oracle optimizer estimates could be correlated to execution speeds with a high degree of confidence, but have found that correlating optimizer cost estimates with actual execution speed to be problematic and unreliable. If possible, however, such correlations would be helpful to developers who are tasked with query creation and optimization. Although much has been written on databases, the academic literature on optimizers was sparse. To fill the gap, this researcher developed a quantitative research methodology to test query optimization on an Oracle 11g database. Correlations between cached, non-cached, partitioned and non-partitioned table structures and indexes were performed. The findings suggest that confident correlations between optimizer cost estimates and execution speeds are not yet possible. Suggestions for further research were provided.

Keywords: Cost Based Optimizer, cardinality, index toggle, correlation, confidence, Oracle

Acknowledgements

A special thanks is given to my wife Heather for her unfailing devotion and patience as this thesis unfolded. Also, thank you to Dr. Nancy Birkenheuer for her enthusiasm and optimism, for providing confidence that hard work pays off, and that this thesis could be completed. Last but not least, to the unending patience of Dr. Ernest Eugster. His “red pen” became my worst enemy as well as my best friend, a special thank you.

Table of Contents

List of Figures

Chapter 1 -- The Oracle Cost Based Optimizer Dilemma

One of the most significant challenges faced by database developers is ensuring fast and predictable data retrieval to justify the contributions that databases can make to an organization's mission. Database vendors developed the Cost Based Optimizer (CBO) which provides a way to understand and tune queries. But, correlating CBO cost estimates with actual execution speed can be problematic and unreliable. CBO estimates, for example, could produce fast results in one circumstance, and similar estimates could then produce slow results in similar circumstances. In addition, although trade literature on the relational database management system (RDBMS) is vast, few academic studies could be found that had relevance. This thesis attempts to redress this dilemma by proposing new correlations between CBO estimates and execution speeds.

This chapter discusses the inner workings of Cost Based Optimizers and their implications. It introduces the different views that researches have had on the sources of CBO inconsistent results. This chapter also introduces the quantitative research design that this researcher adopted to test correlations, using the Oracle RDBMS in an experiment.

Relational Database Management Systems -- Size and Speed

Since Dr. E. J. Codd published his paper "A Relational Model of Data for Large Shared Data Banks" in 1970 proposing a new database model to replace hierarchical and network models, the RDBMS has become more critical than ever to coordinate data management and leverage enterprise data for competitive advantage. This is reflected in the statistics. According to market researcher International Data Corporation, the worldwide market for RDBMS grew an estimated 7.2% in 2011, reaching $22 billion (Thomas, 2011).

RDBMS systems have also grown explosively in their size with databases in some organizations reaching petabyte proportions and beyond. For example, the e-Bay database engine passed the 10 petabyte size in 2010. Additional systems owned by e-Bay for data warehousing passed the 4.5 petabyte threshold, and growth of as much as 100% per year was predicted on both systems (Ratzesberger, 2010). In another development, the United States National Security Agency which is building a data center in Utah will host several RDBMS. This data store is predicted to store yottabytes of data (Trenholm, 2009).

This explosive growth in database use and size, however, has brought with it two challenges. The first challenge was ensuring fast response time when processing data. For example, consider the usage requirements of e-Bay. Burleson (2006), commenting on that system, indicated that it:

- Contained over 212 million registers users
- Contained over two petabytes of Data
- Supported 26 billion SQL executions per day
- Was 99.94% available
- Displayed over one billion page views per day (p. 1)

It is easy to understand that with 26 billion SQL executions per day being issued against this online market, if delays happen because response times were slow, usage would be hampered. For example, research by the Akamai Technologies Inc (Yang & Faris, 2009) showed that users generally expected websites to load in two seconds, and 40% of online shoppers would actually leave the site if page loads took longer than three seconds. The financial fallout for a company like e-Bay would be catastrophic if 40% of its users decided to leave because web-pages took longer than three seconds to load.

The second challenge is related to ensuring the predictability of speed requirements, or more precisely, query results being returned inside a specific time tolerance. Hu, Sundara and

Srinivasan (2007) gave a practical example of this need by talking about a global positioning system (GPS) providing driving directions. Regardless of the size of the database, the response times must be fast, otherwise a driver risks not making the correct turn when needed. Furthermore, the response time must be predictably consistent. Otherwise, a driver approaching a turn quickly may receive instructions too late or too early as a result of acceleration or deceleration of the vehicle.

Queries and Speed (Preliminaries)

In all major RDBMS systems, Structured Query Language (SQL) is the primary tool for interfacing with a database and extracting data (Rob & Coronel, 2007). SQL is declarative in nature (Chamberlin & Boyce, 1974). This means that all functions related to speed and predictability, when extracting data, must reside in the RDBMS system. SQL queries are passed from a user or machine interface to the RDBMS where they are parsed, optimized, compiled and executed. After the query is executed, data is returned to the end user, another program, or another database.

Cost Based Optimizer and Other Speed Enhancing Tools

Query execution speed is the time it takes between a query being submitted to the RBDMS and results being returned to the submitter after query execution (Oracle, 2011). In RDBMS systems, a primary objective of query design is increasing overall query execution speed (Burleson & Danchenkov, 2005).

To aid in the objective of writing queries that perform well, database vendors have provided tools that are transparent to both developers and end users. One of these tools, specifically related to query processing and applied during the query optimization phase, is the Cost Based Optimizer (CBO). The CBO is a fundamental tool associated with all aspects of database queries. The database offerings of Oracle, Microsoft and IBM all utilize CBO functionality (Oracle, 2005; Microsoft, 2009; Fechner, 2006). In an Oracle system, for example, every query that runs through the database first passes through the CBO (Lewis, 2006a). There is no way to disable or circumvent the CBO; it is essential and interwoven with all Oracle queries. Yagoub and Gongloor (2007) stated this precisely when they said: "All SQL statements use the optimizer, which is a part of the Oracle database that determines the most efficient means of accessing the specified data" (p. 3).

A CBO has been incorporated in Oracle database products since 1992 when it was released in Oracle 7 (Colgan, 2005). Since that time, several product refinements have been made with the primary goal of increasing accuracy during optimization which would result in increased query speed (Lejeune, Buch, & Palmer, 2003). According to Lewis (2006), an additional goal has been the increase of the CBO ability to correlate CBO estimates with time execution speed.

Because query speed is a primary objective of query development, additional speed enhancing tools have been provided to aid in database development, debugging and performance tuning. Some of these tools include the SQL Performance Analyzer (Yagoub & Gongloor, 2007), the SQL Tuning Advisor (Hall, 2012), and the Oracle Explain Plan (EP) (Colgan, 2011a). These tools work with SQL query statements and all utilize the CBO at a fundamental level. This researcher used Oracle's EP in a case study to correlate CBO estimates with query execution speeds.

The CBO Process

As the fundamental component of all query optimization, the CBO automatically analyzes all SQL code prior to query execution. During this phase, the queries' potential effort requirements of CPU, I/O, and data cardinality are evaluated inside the RDBMS and assigned numerical numbers signifying the predicted "cost" incurred if that query were executed. The purpose of the CBO is to calculate the cost of multiple query execution plans and to select a plan for the query in question that maximizes performance while minimizing cost (Colgan, 2011a). Higher numbers indicate a larger load on the system, while smaller numbers means less (Oracle, 2005). For example, if a query can be executed with two differing costs, the smaller one should have a faster execution response because it is the more efficient (Lewis, 2006c).

Even though SQL is a declarative language and programmatic access to underlying database structures like source code is not possible, developers must still work to streamline SQL queries (Burleson & Danchenkov, 2005). Use of Oracle's EP, for example, aids in this effort by allowing developers a way to analyze a query in such a way as to see high-level representations of the execution pathway chosen by the CBO, the cost of each step, and the overall cost of the chosen plan (Colgan, 2011a).

The Oracle Explain Plan

The EP tool has been available since the CBO was first introduced in the Oracle 7 RDBMS in 1992 (Colgan, 2011a) and has been suggested as a primary tool for all development activities that require streamlining SQL queries (Niemiec, 1999; Colgan, 2011a). The EP uses several special tables in the RDBMS that are populated when specific SQL commands are entered, and then through either the normal SQL+ interface provided by Oracle (Colgan, 2011a), or through other popular RDBMS access tools such as TOAD provided by Quest Software (OracleFaqs, 2012), developers can see the EP results. The output from the EP is in tree form with each leg showing specific information generated by the CBO. Figure 1.1 gives a simple example of a SQL query EP generated in the TOAD tool:

Plan
SELECT STATEMENT ALL_ROWS Cost: 2 Cardinality: 1
1 FAST DUAL Cost: 2 Cardinality: 1

Figure 1.1 **-- Simple EP example**

This tree was created from the EP of the query: “select ‘Hello World’ from dual;”. It contains two rows: the first is a summary that indicates this is a “SELECT” statement with the optimizer mode of "ALL_ROWS" being utilized, the CBO cost estimate of 2, and the cardinality of 1. The second row is indented with a "1" showing this is the first actually executed line that performs a "FAST" scan of the dual table, having the cost of 2 and the cardinality of 1.

The EP tool can provide more sophisticated information, depending on the complexity of the query involved. Lewis (2006) provided the following EP example from the SQL+ interface with an explanation as shown in Figures 1.2 and 1.3 respectfully:

```
select
        outer.*
from    emp outer
where   outer.dept_no in (
                select  dept_no
                from    dept
                where dept_group = 1
        )
and     outer.sal > (
                select  avg(inner.sal)
                from    emp inner
                where
                        inner.dept_no = outer.dept_no
                and     inner.dept_no in (
                                select  dept_no
                                from    dept
                                where dept_group = 1
                        )
        );
```

Figure 1.2 **– EP Example**

This figure shows a query with three embedded sub-queries: one found directly under the "where" statement, another associated with the outer.sal variable and a ">" operator, and another associated with the inner.dept_no variable embedded two levels in the where section. This query only accessed two tables: the "dept" and the "emp" table. Figure 1.3 shows the CBO has decided the Hash Join is the chosen method for joining these two tables.

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
|Id |  Operation                    |Name     |Rows  |Bytes |Cost    |
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
| 0 |  SELECT STATEMENT             |         |  500 |51500 |     98 |
| 1 |    HASH JOIN                  |         |  500 |51500 |     98 |
| 2 |      VIEW                     |VW_SQL_1 |    6 |  156 |     59 |
| 3 |        SORT GROUP BY          |         |    6 |   78 |     59 |
| 4 |          HASH JOIN            |         |10000 | 126K |     38 |
| 5 |            TABLE ACCESS FULL  |DEPT     |    3 |   15 |      2 |
| 6 |            TABLE ACCESS FULL  |EMP      |20000 | 156K |     35 |
| 7 |      HASH JOIN                |         |10000 | 751K |     38 |
| 8 |        TABLE ACCESS FULL      |DEPT     |    3 |   15 |      2 |
| 9 |        TABLE ACCESS FULL      |EMP      |20000 |1406K |     35 |
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Figure 1.3 **– Explain plan**

The ability of the CBO to rewrite queries into equivalent queries that generate less cost is a fundamental component of the CBO functionality. Lewis (2006) explained that this EP output, taken from an 8i Oracle version, demonstrated that the "optimizer has turned the outer subquery into a simple hash join in lines 7, 8, and 9." Lewis went on to explain that "looking at line 2 [illustrates that] the optimizer has also unnested a subquery—and inside that subquery, the optimizer has also turned our silly inner subquery construct into a simple hash join [in lines 4, 5, and 6]" (p. 242). This example shows some of the power the optimizer has, and some of the significant changes it can make to queries as it searches the execution domain for an execution plan that produces the best cost.

The EP of the CBO can handle extremely complex queries, not just those with a few table joins containing significant joins and filter conditions. Figure 1.4 provides a much more complex query:

```
SELECT DISTINCT OTF.DEA_PID_DEAL DEAL_PID, OTF.DEAL_NAME DEAL_NAME,
OTF.BRANCHCODE
  FROM (  SELECT DEA.DEA_NME_DEAL DEAL_NAME,
              FAC.FAC_NME_FACILITY FACILITY_NAME,
              FP.EFP_CDE_FEE_TYPE FEE_TYPE,
              FP.EFP_CDE_CURRENCY CURRENCY,
              FP.EFP_DTE_EFFECTIVE DATE_FEE_RECEIVED,
              SUM (ACP.APP_AMT_BOOK) FEE_AMT,
              ACP.APP_CDE_PORTFOLIO PORTFOLIO,
              ACP.APP_CDE_EXPENSE EXPENSE_CODE,
              DEA.DEA_PID_DEAL DEA_PID_DEAL,
              DEA.DEA_CDE_BRANCH BRANCHCODE
         FROM VLS_EVENT_FEE_PYMT FP  JOIN VLS_DEAL DEA
              ON DEA.DEA_PID_DEAL = FP.EFP_PID_DEAL
              JOIN VLS_FACILITY FAC  ON FAC.FAC_PID_FACILITY = FP.EFP_RID_OWNER
              JOIN VLS_ACR_CYC_PMT_PT ACP
              ON FP.EFP_RID_FEE_PAYMNT = ACP.APP_RID_ACR_CY_PMT
       GROUP BY DEA.DEA_NME_DEAL,
              FAC.FAC_NME_FACILITY,
              FP.EFP_CDE_FEE_TYPE,
              FP.EFP_CDE_CURRENCY,
              FP.EFP_DTE_EFFECTIVE,
              ACP.APP_CDE_PORTFOLIO,
              ACP.APP_CDE_EXPENSE,
              DEA.DEA_PID_DEAL,
              DEA.DEA_CDE_BRANCH
       UNION ALL
        SELECT DEA.DEA_NME_DEAL DEAL_NAME,
              ' ' FACILITY_NAME,
              'FHAD' FEE_TYPE,
              G.GLE_CDE_CURRENCY CURRENCY,
              MAX (G.GLE_DTE_TRANS_EFF) DATE_FEE_RECEIVED,
              SUM (G.GLE_AMT_ENTRY) FEE_AMT,
              G.GLE_CDE_PORTFOLIO PORTFOLIO,
              G.GLE_CDE_EXPENSE EXPENSE_CODE,
              DEA.DEA_PID_DEAL DEA_PID_DEAL,
              DEA.DEA_CDE_BRANCH BRANCHCODE
         FROM    VLS_GL_ENTRY G  JOIN VLS_DEAL DEA
              ON G.GLE_PID_DEAL = DEA.DEA_PID_DEAL
        WHERE    G.GLE_CDE_GL_SHTNAME = 'MISK'
              AND G.GLE_CDE_ACCTG_OPER = 'OR'
              AND G.GLE_TXT_DESCRIPTON <> 'Deprication'
       GROUP BY DEA.DEA_NME_DEAL,
              G.GLE_CDE_CURRENCY,
              G.GLE_CDE_PORTFOLIO,
              G.GLE_CDE_EXPENSE,
              DEA.DEA_PID_DEAL,
              DEA.DEA_CDE_BRANCH) OTF
ORDER BY 1
```

Figure 1.4 **– More complex query**

This complex query was formatted in the TOAD 11.0.0.116 version. Three queries are represented: two inner queries linked through a UNION statement, and an outer query that pulls from these. In the first inner query appearing above the UNION statement, four tables are joined using ASCII syntax, and after the UNION statement, the second inner query joins two tables with similar syntax. No filtering conditions exist on the first inner query, yet there are three filter conditions on the second inner query: two equijoins, and one not equal condition. The explain plan of this query is show in Figure 1.5, and the cost is highlighted for illustration purposes.

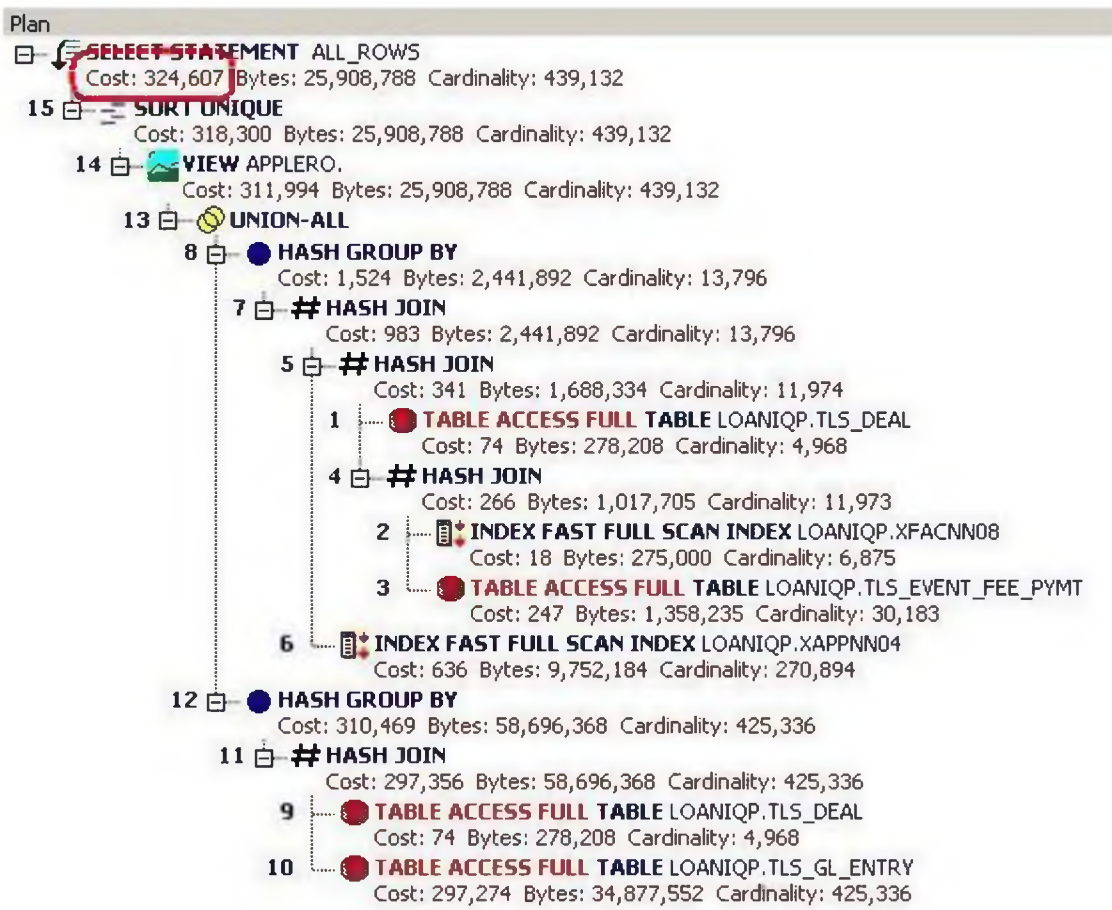


Figure 1.5 **-- Using EP for a more complex query**

The top line in this figure contains a red oval which highlights the overall query cost of 324,607. Developers often rearrange queries to manipulate EP output produced by the CBO which can change cost estimate values. This is what is meant by "streamlining" query performance.

```
SELECT DISTINCT
   OTF.DEA_PID_DEAL DEAL_PID, OTF.DEAL_NAME DEAL_NAME, OTF.BRANCHCODE
FROM (  SELECT /*+ FIRST_ROWS */ DEA.DEA_NME_DEAL DEAL_NAME,
           FAC.FAC_NME_FACILITY FACILITY_NAME,
           FP.EFP_CDE_FEE_TYPE FEE_TYPE,  FP.EFP_CDE_CURRENCY CURRENCY,
           FP.EFP_DTE_EFFECTIVE DATE_FEE_RECEIVED,
           SUM (ACP.APP_AMT_BOOK) FEE_AMT,
           ACP.APP_CDE_PORTFOLIO PORTFOLIO,
           ACP.APP_CDE_EXPENSE EXPENSE_CODE,
           DEA.DEA_PID_DEAL DEA_PID_DEAL,
           DEA.DEA_CDE_BRANCH BRANCHCODE
      FROM VLS_EVENT_FEE_PYMT FP
           JOIN VLS_DEAL DEA
             ON DEA.DEA_PID_DEAL = FP.EFP_PID_DEAL
           JOIN VLS_FACILITY FAC
             ON FAC.FAC_PID_FACILITY = FP.EFP_RID_OWNER
           JOIN VLS_ACR_CYC_PMT_PT ACP
             ON FP.EFP_RID_FEE_PAYMNT = ACP.APP_RID_ACR_CY_PMT
    GROUP BY DEA.DEA_NME_DEAL,
           FAC.FAC_NME_FACILITY,  FP.EFP_CDE_FEE_TYPE,
           FP.EFP_CDE_CURRENCY,  FP.EFP_DTE_EFFECTIVE,
           ACP.APP_CDE_PORTFOLIO,  ACP.APP_CDE_EXPENSE,
           DEA.DEA_PID_DEAL,  DEA.DEA_CDE_BRANCH
    UNION ALL
      SELECT /*+ ORDERED */
           DEA.DEA_NME_DEAL DEAL_NAME,
           ' ' FACILITY_NAME,   'FHAD' FEE_TYPE, G.GLE_CDE_CURRENCY CURRENCY,
           MAX (G.GLE_DTE_TRANS_EFF) DATE_FEE_RECEIVED,
           SUM (G.GLE_AMT_ENTRY) FEE_AMT, G.GLE_CDE_PORTFOLIO PORTFOLIO,
           G.GLE_CDE_EXPENSE EXPENSE_CODE, DEA.DEA_PID_DEAL DEA_PID_DEAL,
           DEA.DEA_CDE_BRANCH BRANCHCODE
      FROM    VLS_GL_ENTRY G JOIN  VLS_DEAL DEA
           ON G.GLE_PID_DEAL = DEA.DEA_PID_DEAL
     WHERE    G.GLE_CDE_GL_SHTNAME = 'MISK'
           AND G.GLE_CDE_ACCTG_OPER = 'OR'
           AND G.GLE_TXT_DESCRIPTON <> 'Deprication'
    GROUP BY DEA.DEA_NME_DEAL,   G.GLE_CDE_CURRENCY,
           G.GLE_CDE_PORTFOLIO,  G.GLE_CDE_EXPENSE,
           DEA.DEA_PID_DEAL,  DEA.DEA_CDE_BRANCH) OTF
ORDER BY 1
```

Figure 1.6 **-- Modified complex query**

Figure 1.6 shows the same query as Figure 1.4, but in this case, an "ORDERED" hint has been placed in the second inner select. This change is circled in red to illustrate. This small change is similar to what a developer would do when manipulating the CBO estimates. The resultant EP in Figure 1.7 does show a slight modification with a different CBO estimate.

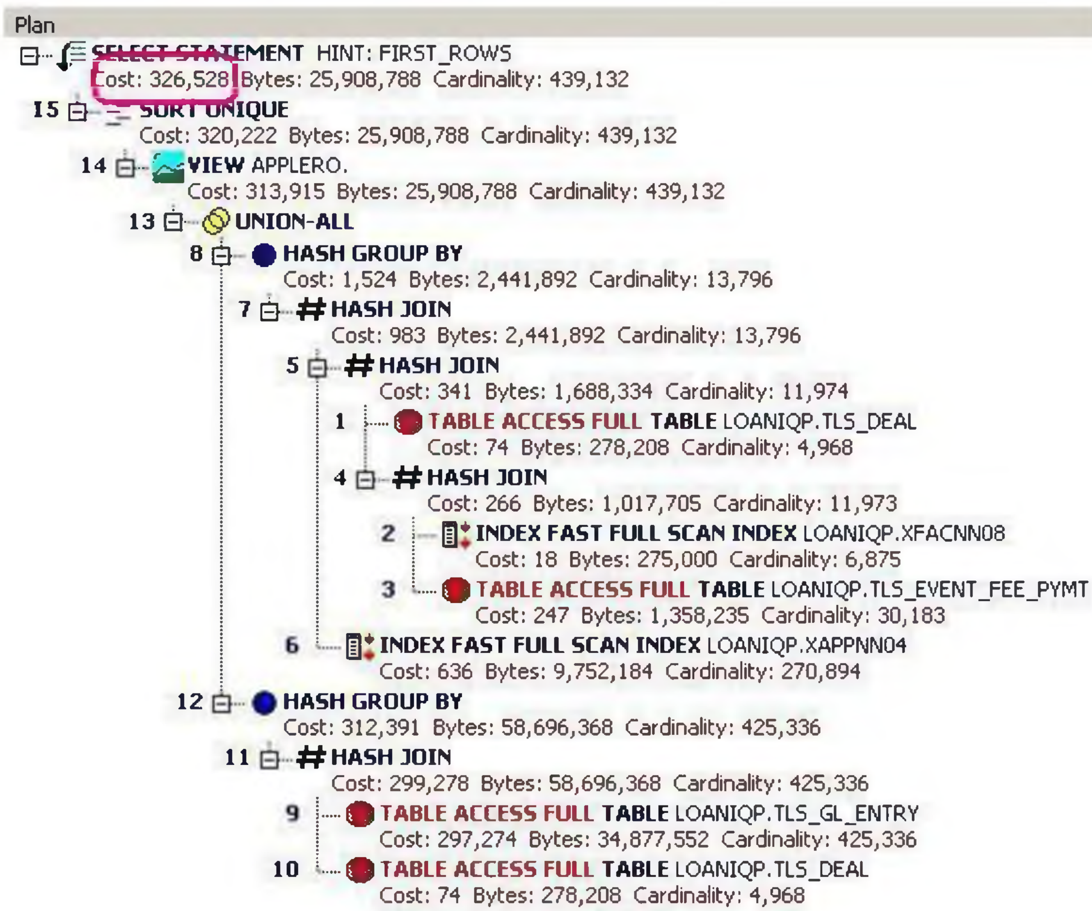


Figure 1.7 **-- Modified query EP**

A small change to the original SQL statement in Figure 1.6 produced a CBO estimate difference. In Figure 1.5, the estimate was 324,607. However, in Figure 1.7, the new estimate was 326,528. From these numbers, it would be reasonable to assume that the query with the smaller cost, namely that shown in figure 1.5, would run faster if both were executed.

Research Problem

In an ideal situation, a developer may create a query that performs according to specific expectations inside a development environment: the execution time is within an acceptable tolerance, and the cost estimate is stable and optimized. Then, when that query is moved to a production environment that includes similar hardware, software, configuration settings and data, it would be expected to perform consistently. For example, if a query incurred the cost of 5000 in development for a specific execution plan, and then if that query is moved to a similar production environment, it would be expected to incur a cost of 5000 and perform in the same amount of time as it did in development.

It would also be expected that consistent execution speeds between development and production environments would aid during post-production support. Through the EP, the developer could know the execution cost of all queries before they are executed. By preserving the execution plans of those queries in production, developers would have a way to predict expected performance (Lewis, 2011). If execution speeds began to decline but the CBO execution plans remained consistent, it would be a signal that performance tuning was necessary (Oracle, 2005).

This idea of plan cost consistency having similar execution performance can also be applied to several queries in the same environment be that development or production. This is slightly different than the idea of moving a query and execution plan from one environment to another and expecting the same execution speed -- this is between two completely different queries in the same RDBMS that have the same CBO cost, but completely different execution plans. For example, if two queries in the same environment each have the same cost of 4500, it is reasonable to assume that because they both use the exact same CBO, they would perform in the same amount of time. Lee and Zait (2008) supported this assessment as they explained that "Cost is a proxy for performance; the lower the cost, the better the performance (e.g. response time) of the query is expected to be" (p. 1368). If the cost for both queries is 4500 and is indeed a consistent proxy for performance, then having exactly the same "expense" would suggest the same execution time.

Lewis (2006b) noted that the CBO cost can be linked to execution time and that consistent costs between multiple queries should execute in similar times. Speaking about the Oracle system, he indicated that "cost is time," meaning that from Oracle 10g forward, a specific column exists in the database, associated with the CBO optimization process, that shows the estimated time to completion. This column is directly related to and proportional with the CBO cost estimates for the query.

Significance

Despite the tremendous efforts made to produce fast and consistent query results, this researcher, among others, has encountered situations where a single query that has more than one possible execution plan performs counter intuitively. In one situation, a plan with a lower cost had a higher execution time than the same query being run with a higher execution cost. This is inconsistent with the data the CBO is supposed to provide--lower cost queries are defined as more efficient (Oracle, 2011), In another situation, this researcher encountered different queries with the same cost estimates that had significantly different execution times--one query running as much as 2000% slower when compared to the first.

Developers and researches have encountered similar concerns across a multitude of environments which included different hardware platforms, software packages, and configuration settings. For example, developers working for a large aerospace company have indicated that query execution speeds are inconsistently correlated to CBO estimates: just because one query has a lower cost estimate, it is common for that query to take significantly more time when compared to different queries that have higher CBO estimates (P. Hardy, personal communication, July 15, 2010).

These types of inconsistencies raise concerns as to whether the CBO estimates are of any value during development, or if there are other things happening which introduce discrepancies between CBO costs and query speed. Kyte (2000) explained that a CBO query cost estimate was an artificial number "arrived at to select a query given a certain environment" (p. 1), and that just because one query had a lower cost, there is no way to compare this to another with a higher cost and see them run as expected. Lewis (2011) countered this argument when he wrote: "The cost of a query represents the optimizer's estimate of how long it will take that query to run—so it is perfectly valid to compare the cost of two queries to see which one the optimizer thinks will be faster" (p. 1). It is clear a disagreement exists as to the purpose, accuracy, and use of the CBO tools. Surprisingly, an endorsement was provided by Kyte in the forwarding section of the book "Cost Based Optimizer Fundamentals" authored by Lewis (2006) in which Lewis indicated clearly that he believed cost should and eventually will reflect a time estimate.

Research Questions and Purpose

The research questions that drive this study are as follows: first, can correlations between CBO estimates and execution speeds be determined; second, can those correlations provide confident predictions for further queries based on CBO estimates; and third, are Lewis and Kyte correct, or are they both wrong. Stated another way, the purpose of this study is to provide a determination of whether developers can use CBO estimates for query tuning and execution speed prediction. If correlations are possible but provide no predictive confidence as to query speed, then developers must still time each query to judge speeds, regardless of the EP cost estimate. Whether this timing step is necessary is addressed in this document, and that determination will be used to address the dilemma between Lewis and Kyte.

Regarding the Research Method

To perform this research, a three phased quantitative approach was adopted which investigated the CBO maturity and also provided a framework for an experiment that empirically looked at query speeds and CBO estimates. The first phase documented trade literature on Oracle RDBMS, with special emphasis to speed enhancement. In a chronological fashion, industry trade documentation was reviewed to show trends have been established up through the 11g offerings where speed enhancements consistently show innovation that directly affect the accuracy and efficiency of the CBO. Lewis (2006) wrote: “One day, perhaps within the next couple of minor releases, you will be able to look at the cost of a query and convert it confidently into an approximate run time, because the optimizer will have produced exactly the right execution plan for your data, on that machine, at that precise moment in time” (p. 3). Considering that Oracle 11g release 2 contains several minor release upgrades as well as a major version change from the Oracle 10g release 2 version available when Lewis wrote this (Portugal, 2009), arrival at the point Lewis mentioned might have already occurred.

With the foundational of Oracle technology in place, the second phase reviewed the academic literature related to CBO time correlations. Although trade literature on RDBMS is vast, the few academic studies that could be found suggest that CBO estimates are moving toward time based correlations, particularly related to progress predictions that involve feedback looping tools that are very close to actual time correlations.

In the third phase, this researcher collected CBO estimates and execution speeds from an Oracle 11g Release 2 database using a variety of techniques to manipulate the CBO plan choices. This researcher used data finding, and linear regression techniques in three specific areas:

1. Queries were compared against themselves in the same RDBMS but with differing cost estimates, and whether those with a lower cost (more efficient) execute consistently faster than those with a higher cost (less efficient) was investigated.
2. Linear regression analysis was applied to the same queries with multiple execution plans and CBO estimates, and correlation confidence was computed between cost estimates and execution speeds.
3. Linear regression analysis was applied to a large sampling of queries, correlating their cost with execution speeds, and correlation confidence was computed.

Summary

This chapter introduced the concept of query optimization through the CBO, and explained that this tool is utilized for all SQL queries processed in RDBMS, the goal being the selection of an optimal execution pathway. Developers often utilize the EP, a tool provided by Oracle, as a way to streamline queries. The EP provides a cost estimate of the plan it selects with the understanding that lower cost plans are expected to perform faster than those with a higher cost. Experience has shown, however, that this is not always the case. Two general trains of thought exist with regards to the use of the CBO: one by Kyte (2000) who indicated the cost estimate is an arbitrary number that has limited relevance to fast or predictable queries, and Lewis (2006) who believed that the CBO estimate would someday become a clear indication of fast and predictable queries. Lewis (2011) maintained this belief years later, still contending that someday CBO estimates will reflect speed predictability. The next chapter presents a trade literature review showing that the Oracle CBO has matured considerably since Lewis made his original assertions in 2006. It lays the foundation for the academic literature review in Chapter 3.

Chapter 2 -- CBO as Speed Promotes Predictability

Chapter 1 introduced the CBO estimate dilemma and how it relates to development problems when working to streamline query statements. It is clear that generational changes with the Oracle RDBMS have always maintained the importance of query speed (Antoshenkov & Ziauddin, 1996; Colgan, 2008). Though query speed has been important, less emphasis appears to have been placed on query speed predictability. As time has passed, however, market influences have driven RDBMS innovation (Carter, 2007; Davies, Shaffer, & L'Her, 2009), and the CBO has received considerable attention (Colgan, 2010). Lewis (2006) recognized this innovative and refining process and believed that as estimates became more reliable in future RDBMS releases, developers would one day be able to correlate the CBO cost to actual execution time.

Based on existing trade press, this chapter examines the historical trends of Oracle RDBMS innovations that directly affected CBO accuracy. The formula for the CBO has changed from Rule Based Optimization to the current CBO, and CBO accuracy has been increased as awareness of data distribution has been built into the formulation. Additional modifications including hardware awareness have also been introduced to improve CBO estimate accuracies. The significance of this chapter revolves around setting a foundation for understanding the academic articles presented in Chapter 3, and the research methodology in Chapter 4.

Speed and CBO Accuracy Related to Time -- An Introduction

Historically, default optimization in Oracle databases was handled by the Rule Based Optimizer (RBO) (Colgan, 2011a). The RBO approach used heuristic rules to create execution plans for SQL queries. Though it provided significant improvement in some situations, the RBO did not take into account data distribution, and could actually slow query execution in some circumstances. For example, suppose a table had a column that could only accept two values: 'Y' or 'N'. If this column were indexed, the RBO would utilize that index arbitrarily if a query were to incorporate this column in the WHERE clause. Now suppose the column contained 'Y' in 99% percent of the circumstances. In this situation, additional I/O would be required by the query filtering on that 'Y' since for every lookup in the index, an additional lookup would then be required to trace the index back to the table source for the original row values in question. Had the query simply scanned the table and omitted the index in the first place, I/O could have been minimized, and the query would have probably performed faster. This type of concern is one of the major problems associated with using heuristically driven optimization: it doesn't take into account the distribution of the actual data (Ahmed, Lee, Witkowski, Das, Su & Zait. 2006).

To improve performance, Oracle developed the CBO which was first released with Oracle 7 in 1992. One advantage the CBO had over the RBO was that it took data distribution into account. With the release of Oracle 9i, the Oracle Corporation urged users to move away from the RBO and start using the CBO for all query optimization. Oracle made the CBO the default optimizer in its 10g offerings in 2003. Today, the RBO exists for backward compatibility reasons only (Colgan, 2008).

Though not dominated by heuristic logic as the RBO was, the CBO still used heuristic approaches in the Oracle 10g and 11g offerings (Colgan, 2008). Ahmed, Lee, Witkowski, Das, Su and Zait (2006) found that in specific situations, run time could actually be improved 387% when using heuristics only in special cases called "unnesting transformations" (p. 1036). Additional findings showed that predicate pushdown approaches improved by 23% and placements in group-by statements improved by 21% over the CBO choices. However, these researchers made it clear that in overall performance comparisons, the CBO "outperformed heuristic-based transformations by 20%" (p. 1036). It was because of findings like these that Oracle permanently moved toward the CBO as its default.

CBO Begins to Mature

Researchers have noted the overall goal of CBO was an increase in accuracy during optimization activities, thus increasing query speed (Lejeune, Buch, & Palmer, 2003; Colgan, 2011a). The CBO depends on a specific formula for calculating costs, and those formulas have changed as the CBO has undergone generational refinement.

Figure 2.1 shows some of the differences to CBO as Oracle moved from the 9i to 10g offerings.

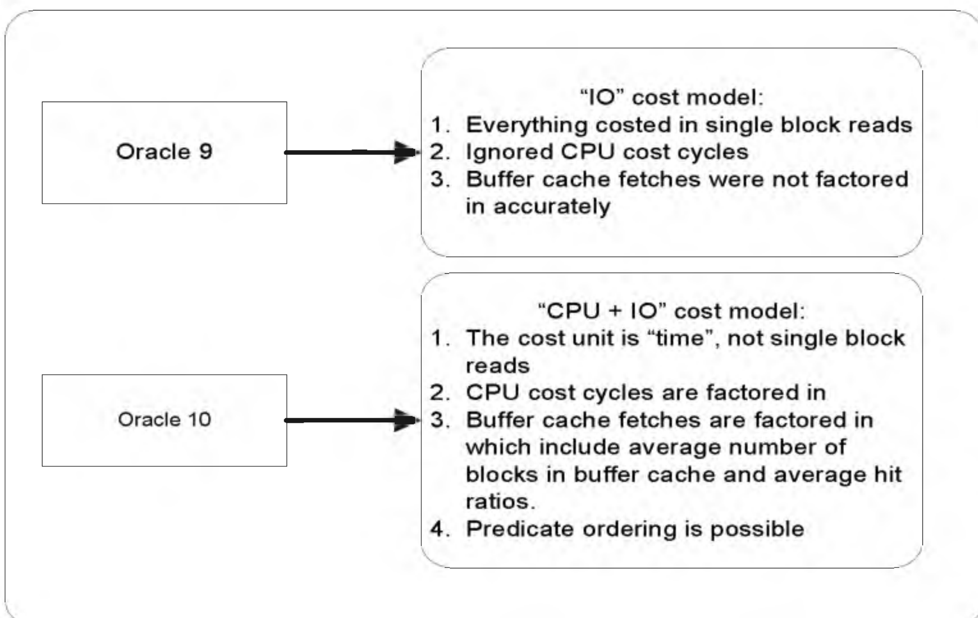


Figure 2.1 **-- High level CBO differences**

The changes to CBO have not gone unnoticed to the academic community. Fan, Jiang, Lumpkin and Sancheti (2003) explained that the 9i RDBMS offering used single block reads as its costing model, "largely [ignoring] CPU costs or [using] imprecise constants to estimate it" (p. 9). This led to inaccuracies in estimates where high CPU cycle counts would be necessary for such requirements as processing functions contained in the queries themselves. Furthermore, data required to supply query results would either be contained on disk or in memory, and memory data (called buffer cache data) would not be factored into CBO estimates accurately. This resulted in inaccuracies as retrieving data from memory is significantly faster than retrieving data from disk storage.

The 10g offering corrected some of these 9i inaccuracies by expanding the CBO model to include CPU cycles. The idea of "time" was also introduced as a cost unit of measurement (Fan, Jiang, Lumpkin & Sancheti, 2003). This is a significant change as it began to link the idea of CBO estimates with query speed predictions (Lewis, 2006). Also, the time factor more accurately took into account the speed differences between data retrieved from disk as opposed to memory, all linking CBO estimates more closely with overall performance, and ultimately with response time itself (Lewis, 2006).

CBO Formula for Oracle 10g and 11g

Lewis (2006, p. 4) provided the current formula the CBO uses for determining cost:

$$\mathrm{Cost} = \left[\frac{\#\mathrm{SRds} * \mathrm{sreadtim} + \#\mathrm{MRds} * \mathrm{mreadtim} + \dfrac{\#\mathrm{CPUCycles}}{\mathrm{cpuspeed}}}{\mathrm{sreadtim}} \right]$$

***Figure 2.2* -- CBO Formula**

The explanation of the variables is as follows:

- "#SRds = number of single block reads
- #MRds = number of multi block reads

- #CPUCycles = number of CPU Cycles
- sreadtim = single block read time in milliseconds
- mreadtim = multi block read time in milliseconds
- cpuspeed = CPU cycles per second" (p. 4)

The sreadtim, mreadtim, and cpuspeed are hardware dependent, and require specific information gathering for their calculation. In explaining how Oracle RDBMS calculates these values, Burleson (2007) noted that the DBMS_STATS.GATHER_SYSTEM_STATS procedure is typically run during a normal workload day on the database instance in question. As work progresses, system sampling occurs where averages are calculated for each of these values, and often stored internally in the system. Oracle recommends that this system gathering activity happen preferably during a peak workload period, and these statistics should be gathered only one time (Colgan, 2008). If a hardware change occurs -- for example, data is moved from internal storage to a SAN solution--then this data gathering activity should be repeated (Colgan, 2008).

In situations where workload statistics are unavailable, Oracle provided a way for gathering non workload statistics; but as Burleson (2007) observed: “the optimizer uses the workload statistics in hopes of getting the 'best' execution plan for the SQL” (p. 1). The current values for these statistics can be extracted from the system using the query in Figure 2.3:

```
SELECT pname "Statistic",
pval1 "Value Set"
FROM  sys.aux_stats$;
```

Figure 2.3 **-- System Statistics SQL**

It should be noted that having inaccurate system statistics will cause the CBO to malfunction. Though it will still produce a cost for each query it processes, that cost will have little reflection on actual results (Colgan, 2011b). It should also be noted that the changes mentioned above, though introduced in earlier Oracle release versions, have perpetuated forward into the current Oracle 11g offerings. The CBO formulas have remained unchanged since Oracle 10g, and the procedures for gathering system statistics still remain in force (Oracle, 2011)

Cardinality

Researchers have universally asserted that cardinality is the single most important factor for CBO accuracy (Lewis, 2006; Galindo-Legaria, Joshi, Waas, & Wu, 2003; Kyte, 2009). Colgan (2011b) made it clear that the accuracy of cardinality is fundamental to everything the CBO does, and since every SQL query is affected by the CBO (Yaboug and Gongloor, 2007), accurate cardinality information has a greater effect on CBO accuracy and speed than any other single factor. This fundamental and even foundational aspect of cardinality means that many of the most important and innovative improvements made to the Oracle RDBMS during generational changes between 9i to 10g, and particularly from 10g to 11g are associated with maintaining clear statistical information relative to the cardinality of the data (Colgan, 2011b). As noted in Figure 2.2 , the CBO uses the number of single block reads, multi block reads, and CPU cycle counts in its calculation. For each query the CBO must analyze, those numbers come from the cardinality of the underlying data.

What Is Cardinality?

Cardinality in a database system refers to the number of values inside a column that the optimizer expects to obtain for a given operation (Kyte, 2005). According to Date (2003), databases are made up of rows and columns, and each column, composed of heterogeneous data, may possess multiple values. Through a system of statistical gathering (Colgan, 2011c), the optimizer looks at the data in question, and depending on the operation in question, will calculate the cardinality accordingly. This cardinality is directly factored into the CBO estimates.

Lewis (2006) provided an example of this cardinality concept. He explained that during a conference, he "managed to draw an audience of 1,200 people. How many of them do you think were born in December [he asks]? If you've decided that the answer is about 100, then you've just performed a perfect imitation of the CBO" (p. 41). This conclusion, however, rested on the following assumptions which he pointed out:

- 12 months exist in the year
- Birth dates are assumed evenly distributed through a given year
- One-twelfth would therefore be born during December
- Since if there are 1,200 people, 100 are expected to be born in December.

In situations where data is not evenly distributed, the CBO can make mistakes because cardinality numbers can become skewed. This concern, addressed later in this paper, is overcome with the use of database histograms (Lewis 2006).

How Cardinality is Discovered (Statistics)

Cardinality on database objects is possible through statistics gathering (Oracle, 2005). According to the Oracle Database Performance Tuning Guide 10g Release 2 (10.2):

> “Optimizer statistics are a collection of data that describe more details about the database and the objects in the database. These statistics are used by the query optimizer to choose the best execution plan for each SQL statement. Optimizer statistics include the following:
>
> - Table statistics

  - Number of rows
  - Number of blocks
  - Average row length
- Column statistics
  - Number of distinct values (NDV) in column
  - Number of nulls in column
  - Data distribution (histogram)
- Index statistics
  - Number of leaf blocks
  - Levels
  - Clustering factor
- System statistics
  - I/O performance and utilization
  - CPU performance and utilization" (p. 14-1)

Prior to Oracle 10g, statistics had to be gathered manually by the DBA, causing problems for the CBO if forgotten or estimated incorrectly. Statistics were gathered automatically in versions 10g and above through an automatic maintenance job that utilizes the GATHER_STATS_JOB procedure (Oracle, 2005). As this maintenance job runs, if there are missing statistics or if the underlying data that contributed to a statistical reading has changed significantly (a situation called stale statistics), then these items are given priority during the next maintenance window. Because of the importance of accurate statistics, it serves as an innovative improvement for the RDBMS to automatically gather statistics as a default function.

Focusing priority on situations where stale statistics hamper performance also supports the importance of cardinality accuracy.

Cardinality Concerns

Clearly, without accurate cardinality, the CBO is left to perform estimations that have little if any relevance to actual performance concerns. To avoid this, two specific items were introduced by Oracle to address problems that might arise: dynamic sampling and histograms.

Dynamic Sampling

Dynamic sampling, introduced in Oracle 9i release 2 and maintained and improved in more recent versions (Colgan, 2010), is a tool used by the CBO "to improve server performance by determining more accurate estimates for predicate selectivity and statistics for tables and indexes" (Oracle, 2005, p. 298). When underlying data have missing statistics, have stale statistics, or have statistics that for various reasons might lead to significant errors in estimation, dynamic sampling can be used. This feature is controlled by the OPTIMIZER_DYNAMIC_SAMPLING parameter, and can be suppressed if wanted.

Oracle (2005) has indicated situations where dynamic sampling should not be used. For example: "[when] a query normally completes quickly (in less than a few seconds), you will not want to incur the cost of dynamic sampling" (p. 298). The reference for "cost" in this context means the time that the optimizer will spend performing the dynamic sampling itself, not the "cost" value of the CBO estimate itself. Oracle continued: "however, dynamic sampling can be beneficial under any of the following conditions:

- A better plan can be found using dynamic sampling.
- The sampling time is a small fraction of total execution time for the query.
- The query will be executed many times" (p. 298).

Histograms

Lewis (2006) explained that data skewing can throw the CBO off target because the optimizer often looks for normal distribution patterns in the data. Drawing from the example he gave above, if the majority of the people in his 1,200 group were born in January instead of December, then the cardinality estimates would be completely wrong. Because of this concern, Oracle's RDBMS utilize histograms on the database columns.

An Oracle histogram is a statistical representation of data skewing present in a table or index, preserved in the RDBMS for access by the CBO, and kept current automatically by the RDBMS when the CBO recognizes such skewing exist (Lewis, 2006; Colgan, 2011b). There are two types of histograms maintained by the Oracle system: height-balanced and frequency.

In the height-balanced histogram, "the column values [of a table or index] are divided into bands so that each band contains approximately the same number of rows" (Oracle, 2005, p. 301). This data separation into bands provides information of where endpoints fall when doing range scans. The frequency histogram places "each column value…[of a table or index into]…a single bucket of this histogram. Each bucket contains the number of occurrences of that single value" (Oracle, 2005, p. 302).

Both of these histograms are created automatically as the Oracle system requires, and each have internal Oracle views that allow their perusal. This automatic creation of histograms is an additional example of the RDBMS system maturing as time passes, and is directly related to CBO accuracy improvement.

9i, 10g, and 11g Statistic Enhancements

Colgan (2008) explained that between the 9i and 10g products, dynamic sampling was enhanced to provide greater system stability in the advent of poor statistics. For example, the

10g offering provided an auto_sample_size function that determined if sampling smaller sets was adequate, and if so, what the sampling size should be. Because statistic gathering itself can tax a system's resources, parameters have been included in the DBMS_STATS package to determine if index statistics need to be collected at all. The 10g offering also allowed for saving statistics. If, for some reason, CBO estimates change significantly after statistics gathering and plan stability deteriorates, it is possible to restore the statistics from a previously saved state and restore CBO plan selection accordingly.

Colgan (2010) further explained that in the 11g offerings, additional sampling enhancements, both at the dynamic level as well as using the DBMS_STATS package, were provided. The DBMS_STATS package now allows for statistics to be copied between tables, and will also capture extended statistics which includes data correlated between columns in the same table. A new sampling algorithm was created to provide more accurate sampling. Furthermore, CBO changes have been made to the group-by placement, allowing the optimizer to perform group-by functions prior to performing some joins. These are all significant changes as the enhancements to statistical information, including improved algorithms, makes the CBO even more accurate in its assessments. This accuracy was what Lewis (2006) spoke about when he projected that someday the CBO estimates would be accurate enough to correlate to execution time.

Indexes -- A Significant Push for Speed

Oracle indexes were created as a way to significantly improve the speed of Oracle SQL queries (Burleson, n.d). As disk storage costs have decreased, Oracle has continued to include new and innovative indexing algorithms in its offerings. However, Leishman (2007) indicated that Oracle really only has two basic index patterns, b-tree and bitmap. All other indexes are based off of these models. These two index types are further identified in a later section.

What an Index Is (Blocks and Avoiding Waste)

Leishman (2007) explained that all Oracle data is stored on computer disk in a small unit called a “block”, and when read, the entire block is read during each oracle I/O operation. This block may come in various sizes, “but is usually one of 4kb, 8kb, 16kb, or 32kb” (p. 1). Because data rows in each block are usually smaller than a single block, as data is read, the Oracle system will gather more information than is usually necessary. The relevant data will be contained somewhere within. Once a relevant block is retrieved, Oracle must then search inside the block’s data for the relevant rows to satisfy a query. This is a time consuming process. If a faster way to find the block containing relevant data were possible, and if the address in that block of the relevant data were known, it would greatly improve the overall speed with which relevant data could be found. This is the purpose of indexing.

B-TREE

Balanced-tree (b-tree) index structures are the most common index type used in computer science and Oracle. Leishman (2007) explained that in a RDBMS system, b-tree structures are not based on data rows, they are based on blocks and the column values those blocks hold. Each b-tree node contains a column value for a table and points to a lower branch that contains more precise values. As one moves through the nodes and arrives at a leaf node, the precise block address of the row being searched, as well as the location in the block where that row is located can be found. This final physical address is called a ROWID.

Leishman (2007, p. 1) provided an example of how a b-tree works. If one was looking for the name Galileo in a b-tree telephone book, the following steps would be followed:

1. You read page 1 and discover that "page 6 starts with Fermat and that page 7 starts with Hawking."
2. You read page 6 and learn that "page 350 starts with Fysche and that page 351 starts with Garibaldi."
3. You read page 350 and find this is a "leaf block." Here you find Galileo's address (this would be the block address), and his phone number (this is where in the block he is found).

Because this system allowed multiple branch blocks for each choice, it is possible to "find specific row[s] in a million row table. In reality, index blocks often fit 100 or more rows, so b-trees are typically quite shallow" (Leishman, 2007, p. 1).

It should be noted that b-tree indexing is the primary form of indexing in Oracle RDBMS systems (Burleson, 2010), but that indexing is not limited to storing single column values: multiple values can be stored, something called a "compound index." The value used for sorting purposes in the index is called the leading edge (Oracle, 2005). In situations where all of the values required to satisfy a query are found in a compound index, the CBO might choose to satisfy the query by pulling values exclusively from the index. This approach is called a "fast full scan" and is another example of some of the innovative ideas Oracle utilizes through the CBO to improve performance efficiency.

Situations exist where b-tree structures do not perform well. One of these happens if the underlying table has low-cardinality columns—meaning "columns with less than 200 distinct values" (Burleson, 2010, p. 1). For example, a table called EMPLOYEES contains a column GENDER. Each employee will be either male or female, but if the table has a significant number of rows, a b-tree index would be of little help when it only has at most two branch blocks. B-tree indexes are also not able to support SQL queries using Oracle's built-in functions (Burleson, 2010). In each of these situations, other forms of indexing are necessary: bitmap and functional based.

Bitmap Index

Bitmap indexes address the concern of low-cardinality in the underlying data. A bitmap index is a two-dimensional array where each row in the table being indexed is represented by a single bit in a column of the index array (Burleson, 2010). In the example above where GENDER is being tracked, a bitmap index would be ideal.

Performing searches in bitmap indexes are fast. Burleson (2010) provided an example of a car manufacturer where car color, make and year were all indexed in a bitmap fashion. Using a specialized bitmap optimizer method and a special merge routine, he said: "Oracle can provide sub-second response times when working against multiple low-cardinality columns" (p. 1), even if there are millions of rows in the underlying table in question.

Function Based Index

In the Oracle RDBMS, when a function is used in the WHERE clause of a SQL statement, the CBO is unable to utilize any associated b-tree indexes (Burleson, 2010). This means that the underlying table must be scanned, and each candidate value of the column in question must have the same function applied and tested for a match against the original query. In large tables in particular, this would be a very costly process as it requires a full table scan (Burleson, 2010). To address this concern, function based indexes were created.

A function based index is a b-tree index that uses the function results of the index column as its search value. The function based index allows the CBO to search matching index values when a matching function is applied to a SQL query in a WHERE clause. This mechanism provides the advantage of fast queries when foresight allows for proper index creation based on known functions.

Tables, Indexes, and Partitions

Dealing with RDBMS systems that are large, such as the petabyte examples discussed in Chapter 1, has proven challenging, not only with regards to storage considerations, but also related to data retrieval speeds when queries performing selections on tables with a large number of rows are not adequately indexed. Oracle Partitioning, introduced in 1997 with Oracle 9i, was designed to improve performance while minimizing I/O considerations (Burleson, 2012).

An Oracle partition is simply a means of taking a large table and perhaps its associated indexes, and splitting it into smaller tables that act independently as though they are different tables all together. These partitions can be stored in different tablespaces, for example, and if a tablespace is taken offline, the remaining portions of the partitioned table remain accessible to the rest of the system. However, when necessary, the partitions can all be logically combined into a single table or index and accessed by the SQL engine as though they were a single unit. This is one of the powers of Oracle partitioning, a divide and concur approach (Burleson, 2012).

Burleson (2012) discussed the partitioning options that were introduced in Oracle 9i as follows:

- "Range Partitioning -- Used when there are logical ranges of data. Possible usage: dates, part numbers, and serial numbers.
- Hash Partitioning -- Used to spread data evenly over partitions. Possible usage: data has no logical groupings.
- List Partitioning -- Used to list together unrelated data into partitions. Possible usage: a number of states list partitioned into a region.

- Composite Range-Hash Partitioning -- Used to range partition first, then spread data into hash partitions. Possible usage: range partition by date of birth, then hash partition by name; store the results into the hash partitions.
- Composite Range-List Partitioning -- Used to range partition first, then spread data into list partitions. Possible usage: range partition by date of birth then list partition by state, then store the results into the list partitions" (p. 1).

Burleson (2012) explained that in Oracle 10g, the CBO became more “partition-aware,” which allowed for partition pruning to enhance query speed throughput. This trend of partition choice enhancement has continued into the Oracle 11g offering as Burleson explained:

- "Interval partitioning for tables [was introduced] -- This is a new 11g partitioning scheme that automatically creates time-based partitions as new data is added…
- System Partitioning [was also introduced] -- The new system partitioning feature in Oracle 11g provides you with the ability to implement and manage new partitions without a specific partition key. Instead, each partition is mapped to a tablespace using the extended partitioning syntax for system partitions. … As such, the main benefit of using the new system partitioning is for manageability purposes” (p. 1).

Oracle (2007) explained in the VLDB and Partitioning Guide 11g Release 1 (11.1) that indexes also benefit with manageability improvements, including availability, performance, and scalability as they are subjected to partitioning schemes. Two particular types of partitioning are available for indexes: global and local.

Global indexes are those that have independent partitioning keys from the associated tables, and don't necessarily need to have the same number of partition segments. In this area, there are two specific offerings: range partitioned and hash partitioned. Oracle (2007) suggested that this type of partition should be used for OLTP applications as it increases flexibility and manageability if changes are needed, without disturbing the associated table structures.

Local indexes are "equi-partitioned" with their underlying tables, meaning "each partition of a local index is associated with exactly one partition of the table" (Oracle, 2007, p. 2-12). This structure enabled Oracle to automatically sync the table partition with its associated index, and provided greater simplicity with their creation, yet at the cost of some of the flexibility afforded the global indexes. Oracle (2007) suggested this type of partition should be used with DSS applications as few changes are often warranted, and keeping things in sync is easier as data is updated into the system, but generally not altered.

Table and index partitioning is a topic that shows a great deal of innovation. Lejeune, Buch and Palmer (2003) argued that when partition pruning became available in Oracle 10g, dramatic performance improvements became possible. Herodotou, Borisov, and Babu (2011) gave examples of how multilevel partitioning, specifically applied to star schemas, allowed for more precision in partition-wise pruning, and resulted in significant speed increases. Optimizers, they argued, are generally "bottom up", something they indicated was very common in traditional database systems, and something that made innovation easier to apply. These examples, and the innovation they afford, show a trend in the Oracle RDBMS: not only is data distribution being taken into account, but the CBO has become more aware of how the data is structurally distributed, not just the cardinality of the data itself.

Plan Optimization -- The Basics of Equivalent Queries and its Power

At this point, several database innovations have been presented to show that trends exist with the Oracle system offerings that continue today (Colgan, 2011b). However, to investigate CBO time correlations, the concept of equivalent queries must be introduced.

Chamberlin and Boyce (1974) argued that SQL is a declarative language, and data extraction from the system, including optimization, relies primarily on the RDBMS system itself. Those who write SQL, however, understand that it is often possible to write queries in multiple ways while still "declaring" the same thing. The CBO does something similar, both with how the query is written as well as how the underlying data is retrieved. This process, the “declaration” of a query in multiple ways that will produce the exact same result set, is what is meant by equivalent queries.

Looking at all the ways the data can be retrieved for use, the CBO has an enormous task of calculating the cost values of multiple pathways to decide which choices would be more efficient (Colgan, 2011a). Through this approach, the CBO is supposed to select an optimized execution plan, hopefully ensuring maximum performance of the query in question. Examples of both equivalent queries and data retrieval are presented here to make this concept more clear.

Equivalent Queries -- An Example

Figure 2.4 shows two tables: CUSTOMER and ACCOUNT. If the customer.account_status field contains only one element of the pair (ACTIVE, INACTIVE), and the account.rep_stat field contains only one element of the pair (CURRENT, TERMINATED), then Figure 2.4 shows two queries that are functionally equivalent:

```
SELECT  a.customer_name,
        a.custoemr_address,
        b.account_rep_id,
        b.account_rep_name
  FROM  customer a JOIN account b
    ON  (a.cust_id = b.cust_id)
WHERE   a.account_status = 'ACTIVE'
   AND  b.rep_stats <> 'TERMINATED'
```
Simple Query 1

```
SELECT  a.customer_name,
        a.customer_address,
        b.account_rep_id,
        b.account_rep_name
  FROM  customer a JOIN account b
    ON  (a.cust_id = b.cust_id)
WHERE   a.accout_status <> 'INACTIVE'
   AND  b.rep_stat = 'CURRENT'
```
Simple Query 2

Both of these queries will return exactly the same results although written differently. If one of these two queries were executed against an Oracle RDBMS, the CBO has the option to "re-write" that query to something different if it decided the cost would be lower with that alternative path. This transformation process is a primary function of the CBO--choosing which query will perform the best (Ahmed, 2010). It is also possible that the CBO would choose yet a third option, as shown in Figure 2.5.

```
SELECT  a.customer_name,
        a.customer_address,
        b.account_rep_id,
        b.account_rep_name
  FROM  customer a JOIN account b
    ON  (a.cust_id = b.cust_id)
WHERE   a.account_status = 'ACTIVE'
   AND  b.rep_stat = 'CURRENT'
```
Simple Query 3

***Figure 2.5* -- New Equivalent Query**

This query returned identical results from those shown in Figure 2.4. It does not, however, utilize any "not equal" matching criteria, and it illustrates a simple way more than one query can be written to return duplicate results.

Simple Join Choices -- An Example

With each SQL query, the CBO also seeks to optimize the query by choosing the best way to access and process the underlying data. A simple example is presented here. Lewis (2006) explained that two common access methods where two or more tables are joined in a query include nested loops and hash joins. A nested loop happens when the first table is looped through to find data that satisfies specific criteria, and then for each row in the first table, the second table is looped through to find rows that match with the first. Hence, it is a loop within a loop, or a "nested loop". A hash join is more complex. In this situation, the smaller table is often parsed according to criteria, and values that satisfy that criteria have a hash function applied to them and are stored off in memory. The second table is then parsed, and depending on the join condition in the SQL query, the same hash function would be applied and matched to that which was stored off in memory from the first table. In both of these approaches, a nested loop or a hash join, data is accessed in a different manner, but both will produce the same results if utilized with the same SQL query. Both of these approaches can even be mixed and used in the same query when more than two tables are involved. However, the CBO makes a choice as to which joins would be the most efficient in each specific situation. That choice which showed the lower "cost" as predicted by the CBO would likely be chosen (Colgan, 2011a).

Besides equivalent queries, the CBO also has the ability to order and apply query predicates. A predicate is “the syntax used to specify a subset of rows to be returned. Predicates are specified in the WHERE clause of a SQL statement” (OracleFaqs, 2008, p. 1). Figure 2.6 shows this concept:

```
SELECT a.customer_name,
       b.customer_address
  FROM  customer a JOIN cust_detail b
     ON  (a.cust_id = b.cust_id)
WHERE b.signup  = 'RECENT'
    AND b.current = 'INACTIVE'
```

Figure 2.6 **-- Predicate Example**

The two items in this WHERE clause are the link between b.signup and “RECENT” and the link between b.current and INACTIVE . Predicate ordering is the process where each of these items is evaluated by the CBO as to efficiency. During optimization, if the CBO finds that applying one predicate ahead of the other produces a lower cost estimate, that predicate order will likely be preserved during query execution.

Progress Continues

The idea of equivalent queries and multiple execution pathway choices, including predicate ordering, opens up vast possibilities for optimizer choices. Colgan (2005) explained that the purpose of the Oracle Optimizer is to "determine the best strategy for performing each query" (p. 4), and that "cost…is based upon in-depth knowledge about I/O, CPU, and memory resources required by each query operation” (p. 23). This includes statistical information, database objects and hardware server platform information. Colgan (2011b) later explained that these improvements were perpetuated into current Oracle 11g offerings with additional algorithm improvements, expansions and improvements to the functionality associated with cardinality accuracy. These improvements have increased CBO accuracy with estimations as well as the cost approximations used for plan selection.

In the Oracle 10g offering, SQL transformations, including heuristic query rules, are applied “whenever possible” to improve the performance of a query (Colgan, 2005, p. 6). This idea of query transformation can also be used to illustrate the complexity of the CBO tool. For example, Colgan (2005, p. 6-7) explained that some of these transformations include "simple view merging" in which the query is re-written in such a way as to eliminate internal references to views completely; "complex view merging" which allows multiple views that perhaps have GROUP-BY or DISTINCT clauses which are merged, thus simplifying and streamlining the code to a better cost outcome; "subquery flattening" where various subqueries are re-written into the form of semi-joins, anti-joins, or simple joins, and others. The list of possibilities is large: additional choices given by Colgan (2005, p. 9-10) include "transitive predicate generation", "common subexpression elimination", "group pruning for 'CUBE' queries", "outer-join to inner-join conversion", and "predicate pushdown and pullup". Predicate pushdown or pullup, also known as predicate ordering, is of particular interest in this thesis because it is useful in showing that even though the CBO has grown in sophistication as each subsequent version has been released, there are still limitations that the CBO cannot easily overcome.

CBO Limitations

Database researchers have asserted that the CBO has limits. Colgan (2005) explained that the CBO must limit its search domain when searching for the best possible execution plan. For example, if there are five tables in a query, there are potentially five factorial choices for predicate ordering. For each of those choices, there are also multiple choices for join methods as well as heuristically driven transformations. If the permutations of possibilities are so extensive that it takes more time to find an optimized execution plan for the query than it would to have run a sub-optimal plan, it defeats the purpose. Oracle therefore utilizes an “adaptive search strategy" that focuses on the best choices first, and limits out those that are considered sub-optimal (Colgan, 2005, p. 15).

Lewis (2006) explained that when the CBO limits the checking of all possible query plans as a result of this "adaptive search strategy" discussed by Colgan (2005), it can result in sub-optimal plans being chosen over better alternatives, causing performance degradation. This limitation of the CBO does not threaten the possibility of time based correlations with the CBO execution costs. The primary reason is that all plans chosen by the CBO use the same formula across all potential query choices. Because the cardinality is computed the same way on all data, if one choice is limited out, even if it would have been considered optimal in comparison to what replaced it, the reasons for that limitation are consistent between both plans; namely, the same optimizer statistics that generated cardinality results, the same system statistics, and the same formulation that generated the original cost are identical for both execution plans.

Summary

This chapter examined high-level changes that the Oracle CBO has undergone. Query speed has been of primary concern, and to increase that speed, the CBO has grown and matured. Vendor literature showed that CBO capabilities were increased between the 9i and 10g offerings, and significant improvements related to maintaining system statics, cardinality, collecting expanded statistics, and even algorithm improvements were introduced and continued to mature up through the 11g offerings. This has brought greater accuracy to the CBO estimates while improving performance throughout. The CBO has become aware of indexing and partitioning, allowing partition pruning to increase speeds dramatically in some situations. The idea of equivalent queries, and how the CBO is capable of not only accessing logical data in an optimized manner, but also rewriting the join clauses and SQL structures into equivalent, yet more optimized versions, was significant. All of these items support the original projections of Lewis (2006); namely, in a future version of Oracle, Lewis speculated that perhaps it would be possible for the CBO to be correlated to actual execution speeds provided the accuracy of the CBO increased sufficiently.

In closing, it should be noted that Lewis (2011) has maintained his position that the CBO estimate is related to time. He remarked: "The cost of a query represents the optimizer's estimate of how long it will take that query to run--so it is perfectly valid to compare the cost of two queries to see which one the optimizer thinks will be faster" (p. 1). Lewis provided an example of four different queries that all resolved in the CBO as equivalent queries, but syntactically were not. As the CBO results were shown, all performed exactly as expected. Lewis indicated, however, that though his example proved his point, comparing costs across different queries for response time was often compromised by "the optimizer model not being right for [the] data, or the optimizer arithmetic [being] to simplistic or having a bug" (p. 1). This reticence by Lewis seems to suggest he was not completely convinced the CBO had matured sufficiently for clear correlations to be established. The next chapter discusses the academic literature related to CBO estimates.

Chapter 3 – CBO in Academic Literature

In Chapter 2, significant emphasis was placed on CBO enhancements and how these related to query speed. Although the trade literature was extensive, academic literature correlating CBO estimates with actual query execution speeds was sparse. Works published address CBO time correlation concerns, but often focus on the challenges with query consistency between similar database systems, and how lower CBO estimates do not necessarily produce faster response times. This chapter explores these topics from the academic literature, and also shows how correlation topics are beginning to emerge as well as some of the specific challenges faced.

Plan Regression

Burleson (2009) explained that "One perplexing issue with Oracle is the dynamic nature of Oracle SQL, and the optimizer's propensity to change execution plans whenever CBO statistics change. This has been dubbed 'Monday Morning Mayhem' because some SQL explain plans will change for the worst" (p. 1). When a plan changes for the worse as Burleson explained, it is called plan regression.

Researchers have long noted that plan regression is a serious problem. Ziauddin, Das, Su, Zhu and Yagoub (2008) explained that various factors, including optimizer statistics, parameter changes, hardware or software changes, can cause the Oracle Cost Based Optimizer to choose sub-optimal plans. When this happens, business activities that depend on query performance can suffer markedly. Working to address this concern, these researchers indicated that Oracle 11g was equipped with a "SQL Plan Management" tool (p. 1347), tightly integrated with Oracle's query optimizer systems. These researchers indicated that this tool possesses two primary goals: "preventing sub-optimal plans from being executed, [and] allowing new plans to be used if they are verifiably better than previous plans" (p. 1346). They explained that the idea of managing plan regression is something not unique to Oracle alone: Microsoft and IBM have methods to do something similar with their respective SQL Server and DB2 offerings.

A significant point must be noted however, and that is plan regression happens when a plan that has a "lower" cost is actually run in place of a plan with a "higher" cost, but performs worse. This should not happen; the lower cost plan should consistently perform better (Oracle, 2011). If the Oracle RDBMS can not look at the same query with two execution plans that differ and ensure that the one with the lower cost actually performs better, then it appears the CBO is not dependable at all. Most of the academic literature that attempts to correlate CBO estimates with time approaches the topic in a way similar to or related to plan regression. Finding "better" CBO estimates and constraining them to perform in a more optimal way is a common theme in many articles that are relevant to this research.

CBO Innovation to Predict Cardinality

As mentioned in Chapter 2, scholars have argued that cardinality is the single most important aspect associated with CBO accuracy and speed. Though cardinality is calculated automatically through statistics gathering in Oracle 11g (Colgan, 2011b), if tables change or statistics become inconsistent—for example between a development and production database—then CBO accuracy suffers. In their study on the relations among cardinality, query speed inconsistencies and plan regression, Tendulkar and Phalak (2011) explained that predicting query performance quality against high volume database systems was particularly difficult for developers. Often the software had to be deployed to production, and only then could unforeseen performance problems be addressed, making it difficult to conform to company expectations and service level agreement requirements. These researchers suggested that the

SQL Performance Assurance Services (SQL-PASS) Oracle tool addressed some of these concerns. This tool is deployable on development systems, and should allow developers to extrapolate production statistics into new systems through percentage, or through specifying actual row numbers. This statistical change affects cardinality approximations and directly influences the CBO plan selection. CBO scores in 11g have two general characteristics: "CPU Cost – Number of machine cycles required for query execution" (p. 542); and "I/O Cost – Number of physical block reads required for query execution" (p. 542). Tables are not always populated in a consistent manner: some tables grow faster than others, and some are quite static. SQL-PASS contained functionality that addressed this concern, allowing adjustments when seeking to project statistics into the future to facilitate "what if" testing. These researchers were also able to show that SQL-PASS can function in cloud computing environments, making it unnecessary to install on development servers. They argued it was also secure in that it affords users secure access to their projects and statistical changes.

Query Self Tuning – Feedback Loops

Scholars are considering the need for queries that tune themselves during actual execution. For example, Lee and Zait (2008) indicated that Oracle Optimization (OO) utilized data provided to it for query execution pathway choices. Included were statistics related to data space utilization, number of rows involved in queries, distribution of columns, basic cardinality of data, and also system statistics including I/O bandwidth and sub-system storage. The quality of the final query was directly related to the value of the data passed to the OO system.

Two specific items, however, caused problems: poor statistics on the intermediate cardinality of queries, and plan reuse when utilizing bind-variables. Lee and Zait (2008) indicated: "Cost is a proxy for performance; the lower the cost, the better the performance (e.g.

response time) of the query is expected to be" (p. 1368). Interestingly, they pointed out that the Oracle 10g systems did not create new execution plans for query reuse do to bind-variable utilization. Consequently, if the data was skewed, the optimization plan for that query could be extremely inefficient. This problem was remedied in Oracle 11g which showed CBO innovation and maturation as new versions were released.

These researchers also discussed a plan they said could be used in the Oracle 11g release; namely, creating a feedback loop to the OO engine on all queries that are processed to provide statistical information with regards to intermediate values for similar queries. Lee and Zait (2008) also mentioned that providing additional information on bind-variable query plans that should naturally change for data skewing as well as other distribution concerns would be helpful. They argued that this approach could be implemented with two specific goals: 1) "zero input from SQL developers and zero maintenance for the DBA" (p. 1378), and 2) "extremely low overhead from monitoring the SQL statements that are limited to only the statements that may benefit from feedback" (p. 1378). The researchers pointed out that their approach has been soundly validated using the Oracle E-Business Suite Workload tools for data replication and testing.

Time Constrained SQL Queries

Hu, Sundara and Srinivasan (2007) explained that with database technology growing, business demands have expanded the amount of data storage to petabyte sizes. However, despite the size of datasets, circumstances existed that required query results within given parameters, including those that are time driven. The researchers argued that Oracle technology offered "hints" that often allow for fast query results, including 'first-few rows', or 'top-k rows.' However, they proposed a system that took an additional syntactical clause requiring the query to adhere to a “soft” or “hard” time limit, as well as allowing for “partial” or “approximate” results. When query results were time based, the researchers explained that “trading off quality results for predictable response time is quite useful” (p. 1207).

They developed a method that showed how syntactical statements could be developed and intertwined with the CBO, taking advantage of cardinality sets and system statistics. Where hard limits were defined and partial or summarized results were allowable, their system auto adjusted accordingly. These researchers setup a controlled experimental investigation which successfully showed feasibility for their approach: query time could be constrained in a number of cases, and the “use of smaller time limits [did] reduce the overall query execution time” (p. 1217). The researchers planned to further their work, but did mention that the inaccuracies with CBO estimates, especially regarding missed cardinality results, were difficult to overcome.

In a follow-up paper, Hu, Sundara and Srinivasan (2009) explained that developed approaches for time-constrained SQL queries involved sampling to “reduce the amount of data that [needed] to be processed, thereby allowing completion of the query in the specified time constraint” (p. 1104). This approach proved useful and effective when estimations could be allowed. But when performing aggregate functions such as MIN, MAX, MEDIAN, SUM and COUNT, the approach proved more complex and problematic. Hu, Sundara and Srinivasan presented two general concepts: 1) when using time constraints, there needed to be an implicit transformation to a SAMPLE clause on individual tables; and 2) estimates on aggregates needed to consider an additional measure of the “goodness” (meaning accuracy) of the result. They explained how Bernoulli Sampling could be used to assist in the process, and through mathematical theory, confidence intervals for SUM, COUNT, AVG and MEDIAN could be calculated. Hu, Sundara and Srinivasan concluded their work stating that their results were a foundation of "estimation in time-constrained approximate queries" (p. 1107).

To the best knowledge of this researcher, the work of Hu, Sundara and Srinivasan was the first attempt to link CBO estimates to actual execution speeds. However, it doesn't directly address the topic as their focus was directed at constraining the query to a specific time tolerance, not predicting what the time will be prior to query execution. They did, however, make it clear that situations existed where queries must perform within a specific time tolerance, and their approach even suggested that data approximations might be a trade off when results were needed within a specific timeframe.

Keeping Query Speeds Consistent

Waas, Giakaumakis and Zhang (2011) recognized plan regression as a significant problem in current production environments, and that the CBO might even compound the problem by choosing sub-optimal plans when superior plans are not even considered. They gave two primary reasons why plan regression itself occurred: 1) optimizer code-level changes [happened] to the optimizer" (p. 1); and 2) changes were applied to the query, logical or physical changes were made to database structures, changes were made to underlying data, or recomputed statistics were calculated. They explained that when the optimizer parses the space of possible execution plans, hundreds of millions of possibilities exist. An exhaustive search would slow things down significantly, so the optimizer must use various techniques to find the "best found plan" (BFP).

This approximation approach often misses plans that would be better choices; therefore, the BFP does not guarantee the optimal plan. Through a plan regression approach, Oracle can create plan baselines, and their use can be guaranteed to not change until another plan that is verifiably superior is generated (Colgan, 2010). Although helpful in production environments, Waas, Giakaumakis and Zhang argued that this approach did not "help engineers to direct their efforts to actually improve the product" (p. 1). They explained that current developer practices included five steps: 1) create a plan regression suite of benchmark, customer, or workload queries; 2) extract the BFP for each; 3) execute against a test suite and find all new BFP choices; 4) in case of discrepancies, evaluate; and 5) implement the change if needed.

This approach, however, had at least two failings: only BFP changes are tested while other far ranging changes might be ignored, and significant manual intervention is required. Optimizers are generally good at estimating which plans are better than others; however, the researchers argued that these estimates fail considerably when comparing estimated performance between actual performance. This is what plan regression is all about. Waas, Giakaumakis and Zhang provided a general way to sample actual vs. estimated query selections from an expanded BFP space, and statistical relevance between the two were gathered. A measure of "how well a given optimizer models a query's plan space" (p. 3) was presented, and preliminary results reviewed.

It is interesting to this researcher that plan regression could be compounded by optimal plans not even being considered during plan selection. If plan regression can happen, then it seems possible that plans omitted from BFP searches might actually land on both sides of the performance spectrum: some may be regressed, and some not. Waas, Giakaumakis and Zhang indicated that developers must execute plans on a test system to find superior BFP items. By this they meant the queries must actually be executed and timed. Their work illustrated at least three concerns the CBO has when considering cost to execution speed correlations:

1) if the test system is not an exact match of production, meaning identical cardinality on all items in question as well as underlying hardware, then there is no guarantee that results will be similar or even relevant after testing: the execution speeds on both systems might differ because the execution plan on each system might differ at the time of the actual test;

2) no method exists to correlate CBO estimates to actual query speed before actual execution: actual testing must be performed;

3) and this approach is very labor intensive.

All of these items help to show that if CBO estimates could be correlated to actual execution speeds, it would be a helpful benefit, particular when considering this approach for regression testing and remediation.

Progress Indicators

Luo, Naughton, Ellmann and Watzke (2004) approached the concept of query execution speed from the perspective of a progress indicator. They explained: “Progress indicators are a widely used user-interface technique in modern software systems….Typically, a progress indicator has the following two features: 1) It keeps track of the percentage of the task that has been completed; 2) [and] it continuously estimates the remaining task execution time” (p. 1). They also indicated that at the time of this writing, no commercial RDBMS provided such a tool, and they were also “unaware of any published techniques for supporting such a progress indicator” (p. 1). Through a feedback loop and by continuously refining cost estimates during execution which included keeping track of current execution speeds, they established that “progress indicator[s] … were useful for both I/O-intensive and CPU-intensive queries, and that [their approach] adapted both to the optimizer’s estimation errors and to varying runtime system loads" (p. 11). Luo, Naughton, Ellmann and Watzke did indicate that even though their approach was "fairly course", it did open up additional possibilities for progress indicators for such things as load management, automatic administration, and even performance tuning.

Long Running Queries

Krompass, Kuno, Wiener, Wilkinson, Dayal and Kemper (2009) explained that long-running queries "are a known problem in all commercial database products" (p. 132). Often, business intelligence workloads contain many queries where execution times vary widely, "from seconds to hours" (p. 132). Despite this fact, the researchers argued that to their knowledge they have never seen a "thorough classification of long-running queries nor a systematic study of the most effective corrective actions" (p. 132). Performance was often hampered with unreliable cost estimates that resulted in plan regression, which they called "problem queries", resource contention that went unnoticed, and systems that were overloaded as the result of the first two items. The only solution to this was to reduce the number of queries on the system, or to filter out and carefully control the queries that are problems. Krompass, Kuno, Wiener, Wilkinson, Dayal and Kemper went on to propose a taxonomy for distinguishing long-running query types, and propose specific workload management techniques to address concerns by preventing those queries from executing, or scheduling them in such a way as to prevent other work from successfully acquiring necessary system resources. Recognizing that cost estimates can be significantly off, the researchers argued that a "Kill & Requeue" or a "Suspend & Resume" (p. 143) process could reactively address concerns by managing resources during operational execution.

The idea of long-running queries was addressed by Oracle. The Oracle Database Reference 11g Release 1 (11.1) Part Number B28320-03 (2009) explained: "[the view]

V$SESSION_LONGOPS displays the status of various operations that run for longer than 6 seconds (in absolute time). These operations currently include many backup and recovery functions, statistics gathering, and query execution, and more operations for every Oracle release" (p. 956). It is interesting to this researcher that there is a "TIME_REMAINING" field in this view defined as "estimate (in seconds) of time remaining for the operation to complete" (p. 957). For such a field to exist, clear feedback from the database would be necessary, and estimates of "UNITS"—which is a measurement unit field also in the view—would be necessary. This view is available in Oracle 11g when the TIMED_STATISTICS or SQL_TRACE parameters are set to true, and proper statistics have been gathered on objects with either the ANALYZE command or the DBMS_STATS package

Summary

Academic research that has linked CBO estimates to actual execution speeds is sparse, despite some recent advancements. All of these articles in one way or another touch upon the idea of CBO cost time estimates; however, they all miss the core—none of them actually correlate CBO costs to recorded query execution speeds. Yet, some researchers have cast doubt on the feasibility that CBO estimates can be correlated to time. Ziauddin, Das, Su, Zhu and Yagoub (2008) in particular, speaking about plan regression, indicated that tools exist internally with the Oracle system that control potential plan regression, preventing plans that might have a lower cost and will potentially result in an inferior performance, from replacing those that perform better. From the literature, this plan regression reality suggests that the CBO does not always perform as expected. This information casts doubt on the feasibility that CBO correlations to execution speeds are possible, something this thesis investigates. The next chapter introduces the quantitative research method used in this thesis.

Chapter 4 – Methodology

This chapter presents the quantitative research method of this thesis: an Oracle experiment that gathered data to investigate CBO correlations to execution speeds. To perform this task, an Oracle system was used where data structures were created, populated with randomized data, analyzed to promote CBO estimate accuracy, and then tested. This chapter provides the details of each step of this process, including information about the underlying system that was utilized, the software release used for the RDBMS, the DDL structure for the tables and indexes that were used, how the structures were populated with data, and the format and testing process where data was gathered. Each detail is presented in a way to provide completeness so that reproducibility is possible, and also to provide a solid foundation for the presentation of findings and analysis that are detailed in the next chapter.

The Oracle CBO Experiment

This experiment was created to test the three criteria outlined in Chapter 1: namely,

1. Comparing queries against themselves in the same RDBMS but with differing CBO estimates and seeing if those with lower CBO estimates perform faster than those with higher;
2. Performing linear regression on the same queries that have multiple CBO estimates, and computing the correlational confidence between cost and execution speeds;
3. Performing linear regression across a large sampling of queries, and computing their correlational confident between cost and execution speed.

As all of these tests were performed on specific hardware, software, and table structures, those individual components are outlined here.

The System Used

A Linux machine running RedHat 2.6.18 was utilized for this case study. Figure 4.1 shows the information as output from the “uname –a” command, and provides important configuration information:

```
(DEV) oracle@trulodb9001$ uname -a
Linux trulodb9001 2.6.18-194.32.1.el5 #1 SMP Mon Dec 20
10:52:42 EST 2010 x86_64 x86_64 x86_64 GNU/Linux
```

Figure 4.1 **- Machine OS**

This machine was equipped with two CPUs, 3821 megs of memory, and was dedicated exclusively to the Oracle processes running on it. Figure 4.2 shows information as output from the “free –m” command, and provides important memory configuration information:

```
(DEV) oracle@trulodb9001$ free -m
             total       used       free     shared    buffers     cached
Mem:          3821       3676        145          0         72        337
-/+ buffers/cache:       3266        555
Swap:         1695         87       1608
```

Figure 4.2 **- Machine Memory**

Oracle Installation

The Oracle Banner from a SQLPlus connection shows the Oracle version being utilized as 11.2.0.2. The output from the Banner is show in Figure 4.3:

```
SQL> select * from v$version;

BANNER
--------------------------------------------------------------------------
Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - 64bit
Production
PL/SQL Release 11.2.0.2.0 - Production
CORE    11.2.0.2.0      Production
TNS for Linux: Version 11.2.0.2.0 - Production
NLSRTL Version 11.2.0.2.0 - Production
```

***Figure 4.3* - Oracle Banner**

The current patch release is show in Figure 4.4 as a reflection of OPatch output:

```
(DEV) oracle@trulodb9001$ ./opatch lsinventory
Invoking OPatch 11.2.0.1.4

Oracle Interim Patch Installer version 11.2.0.1.4
Copyright (c) 2010, Oracle Corporation.  All rights reserved.


Oracle Home       : /opt/oracle/product/11.2.0.2
Central Inventory : /opt/oracle/Inv
   from           : /etc/oraInst.loc
OPatch version    : 11.2.0.1.4
OUI version       : 11.2.0.2.0
OUI location      : /opt/oracle/product/11.2.0.2/oui
Log file location : /opt/oracle/product/11.2.0.2/cfgtoollogs/opatch/opatch2013-
07-06_14-40-11PM.lo

Patch history file: /opt/oracle/product/11.2.0.2/cfgtoollogs/opatch/
opatch_history.txt

Lsinventory Output file location : /opt/oracle/product/11.2.0.2/cfgtoollogs/
opatch/lsinv/lsinvento

--------------------------------------------------------------------------------
Installed Top-level Products (1):

Oracle Database 11g                                                 11.2.0.2.0
There are 1 products installed in this Oracle Home.


Interim patches (7) :

Patch  10626132     : applied on Mon Mar 14 10:37:44 EDT 2011
Unique Patch ID:  13350217
   Created on 31 Dec 2010, 00:18:12 hrs PST8PDT
   Bugs fixed:
                                      10626132
```

```
Patch  10387939     : applied on Mon Mar 14 10:36:59 EDT 2011
Unique Patch ID:  13350217
   Created on 30 Dec 2010, 22:55:01 hrs PST8PDT
   Bugs fixed:
     10158965, 9940990, 10190642, 10031806, 10228635, 10018789, 9744252
     10010252, 9956713, 10204358, 9715581, 9770451, 10094635, 10121589
     10170431, 9824198, 10071193, 10145612, 10035737, 9845644, 10086980
     10052141, 10039731, 10035521, 10219576, 10184634, 10207092, 10138589
     10209232, 8752691, 9965655, 9819413, 9500046, 10106828, 10220118, 9881076
     9869287, 10040531, 10122077, 10218814, 10261389, 10033603, 9788588
     9735237, 10126219, 10043801, 10073205, 10205715, 9709292, 10105926
     10079168, 10098253, 10005127, 10013431, 10228151, 10092153, 10142909
     10238786, 10260808, 10033071, 9791810, 10052956, 9309735, 10026972
     10080579, 10073683, 10004943, 10019218, 9539440, 10022980, 10061490
     10006008, 6523037, 9724970, 10142776, 10208386, 10113803, 10261680
     9671271, 10084145, 10051966, 10355493, 10227133, 10229719, 10046912
     10228393, 10353054, 10142788, 10221016, 9414040, 10127360, 10310299
     10094201, 9591812, 10129643, 10332589, 10026193, 10195991, 10260870
     10248523, 9951423, 10261072, 10299224, 10230571, 10222719, 10233732
     10113633, 10102506, 10094949, 10077191, 10329146, 8685446, 10048701
     10314582, 10149223, 10245259, 10151017, 9924349, 10245086, 11074393

Patch  10157622     : applied on Mon Mar 14 10:36:07 EDT 2011
Unique Patch ID:  13350217
   Created on 19 Nov 2010, 01:41:19 hrs PST8PDT
   Bugs fixed:
     9979706, 9959110, 10016083, 10015460, 10014392, 9918485, 10157622
     10089120, 10057296, 9971646, 10053985, 10040647, 9978765, 9864003
     10069541, 10110969, 10107380, 9915329, 10044622, 10029119, 9812970
     10083009, 9812956, 10048027, 10036193, 10008467, 10040109, 10015210
     10083789, 10033106, 10073372, 9876201, 10042143, 9963327, 9679401
     10062301, 10018215, 10075643, 10007185, 10071992, 10057680, 10038791
     10124517, 10048487, 10078086, 9926027, 10052721, 9944948, 10028235
     10146768, 10011084, 10027079, 10028343, 10045436, 9907089, 10073075
     10175855, 10178670, 10072474, 10036834, 9975837, 10028637, 10029900,
9949676

Patch  10317487     : applied on Mon Mar 14 10:22:25 EDT 2011
Unique Patch ID:  13409177
   Created on 2 Feb 2011, 02:52:55 hrs PST8PDT
   Bugs fixed:
     10317487

Patch  9438890      : applied on Mon Mar 14 10:21:21 EDT 2011
Unique Patch ID:  13104376
   Created on 2 Nov 2010, 08:44:19 hrs PST8PDT
   Bugs fixed:
     9438890

Patch  10177856     : applied on Mon Mar 14 10:18:42 EDT 2011
Unique Patch ID:  13418693
   Created on 4 Feb 2011, 01:31:59 hrs PST8PDT
   Bugs fixed:
     10177856

Patch  10193846     : applied on Mon Mar 14 10:18:07 EDT 2011
Unique Patch ID:  13212424
   Created on 6 Dec 2010, 01:49:29 hrs PST8PDT
   Bugs fixed:
     10193846
```

Figure 4.4 **- Opatch**

Finally, with regards to the internal Oracle memory configurations, a copy of the Oracle PFile is show in Figure 4.5:

```
SANDBOX.__db_cache_size=922746880
SANDBOX.__java_pool_size=16777216
SANDBOX.__large_pool_size=16777216
SANDBOX.__oracle_base='/opt/oracle'#ORACLE_BASE set from environment
SANDBOX.__pga_aggregate_target=402653184
SANDBOX.__sga_target=1207959552
SANDBOX.__shared_io_pool_size=0
SANDBOX.__shared_pool_size=234881024
SANDBOX.__streams_pool_size=0
*.audit_file_dest='/opt/oracle/admin/SANDBOX/adump'
*.audit_trail='db'
*.compatible='11.2.0.0.0'
*.control_files='/oradata/SANDBOX/data01/SANDBOX/control01.ctl','/oradata/
SANDBOX/data01/SANDBOX/control02.ctl'
*.db_block_size=8192
*.db_domain=''
*.db_name='SANDBOX'
*.diagnostic_dest='/opt/oracle'
*.dispatchers='(PROTOCOL=TCP) (SERVICE=SANDBOXXDB)'
*.open_cursors=300
*.optimizer_mode='ALL_ROWS'
*.pga_aggregate_target=400556032
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=1201668096
*.undo_retention=90
*.undo_tablespace='UNDO01'
```

Figure 4.5 **- PFile**

The Table Structure

Four sets of table structures were created for this case study inside this Oracle RDBMS. All were named similarly at the base level; namely, MASTER_1, MASTER_2, MASTER_3 and MASTER_4. The first three tables in this list each have additional satellite tables that form respective "STAR" schemas. These satellite tables link back to their respective MASTER table, and each was named in such a way as to make identification of its level, corresponding placement, and links easily recognizable. The ERD diagrams for the groupings of MASTER_1, MASTER_2, and MASTER_3 are given here in Figures 4.6, 4.7, and 4.8 respectively.

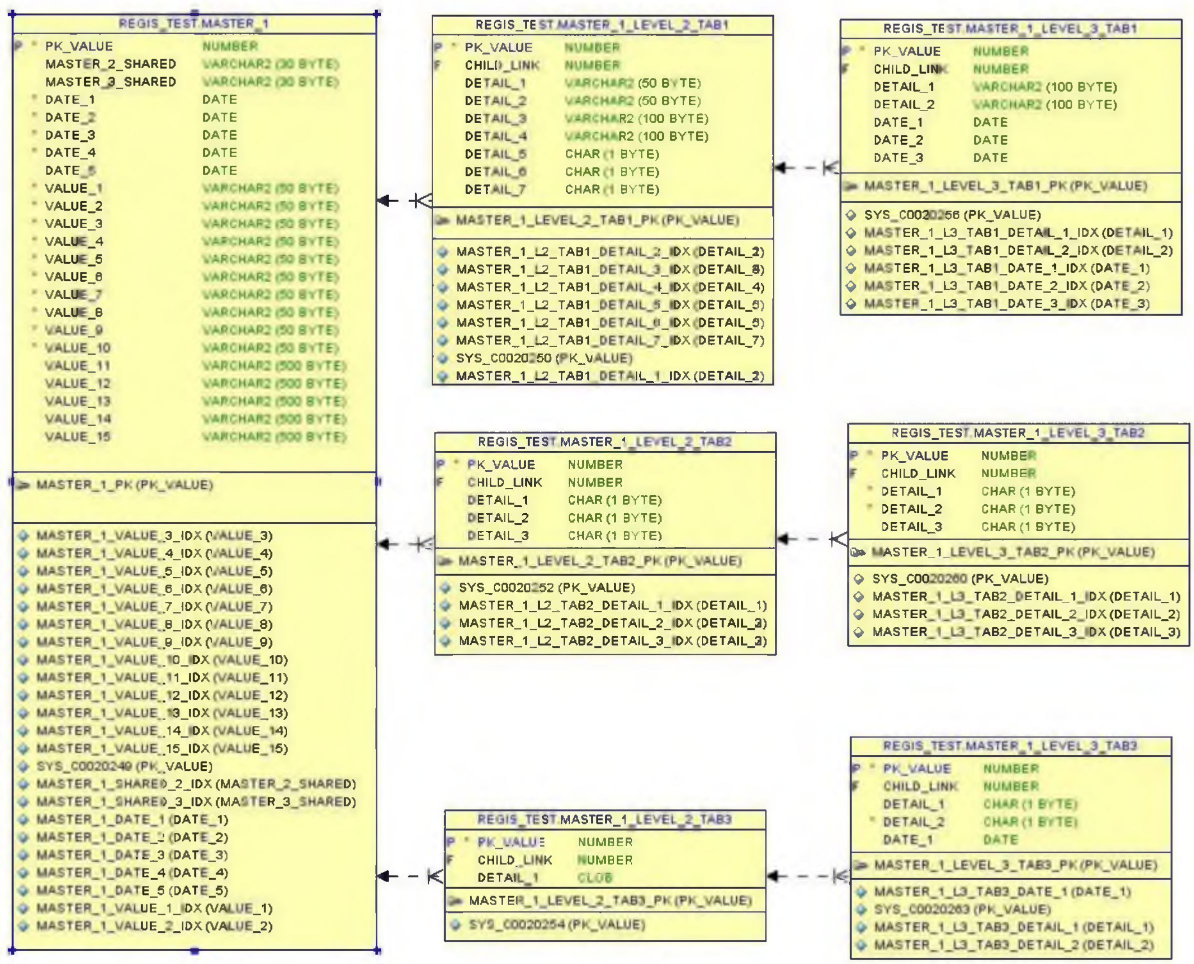


Figure 4.6 - **Master 1 Table Group Structure**

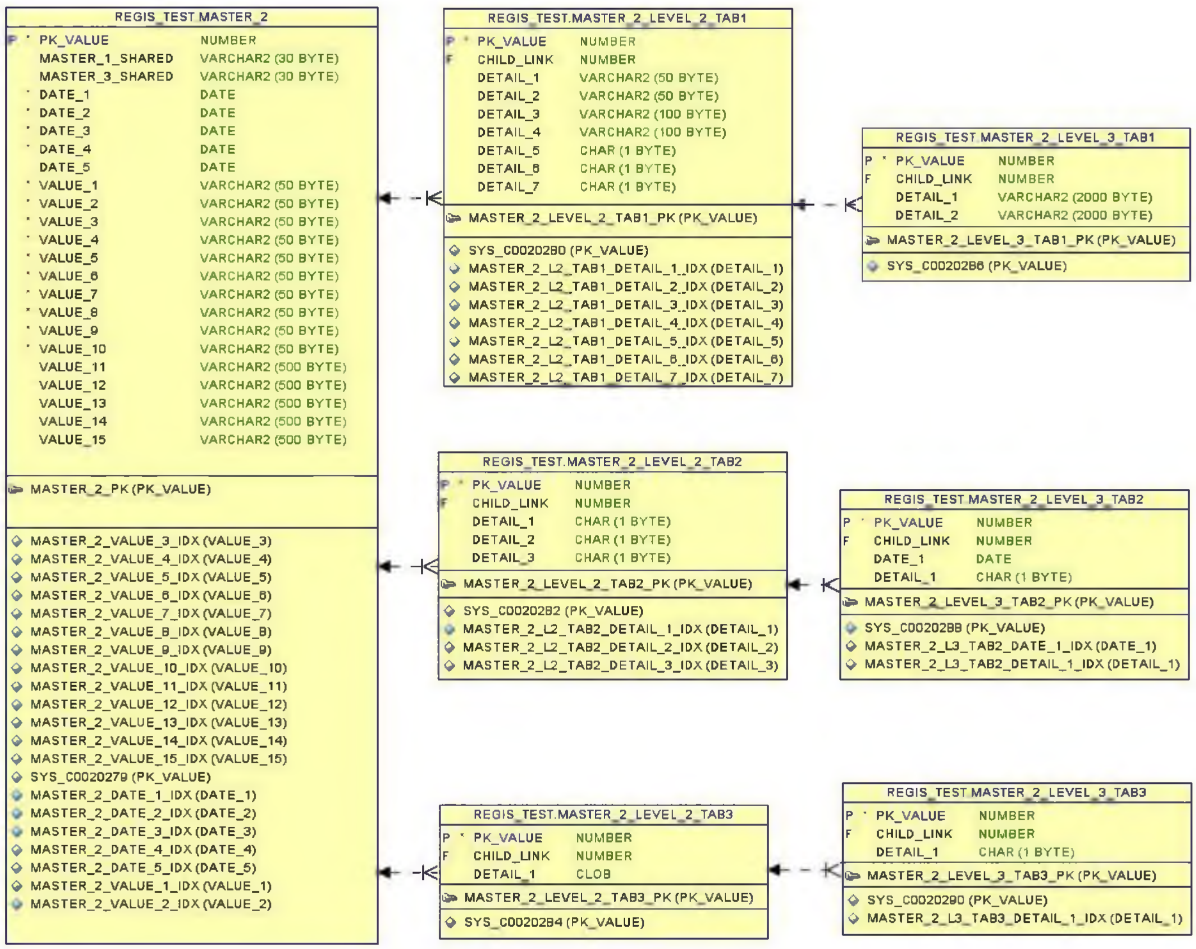


Figure 4.7 - **Master 2 Table Group Structure**

Figure 4.8 - **Master 3 Table Group Structure**

MASTER_4 is a slightly different table; it was created following a range partitioning scheme on the "DATE_1" field, with list sub-partitions on the "CHAR_1" field. Because of the volume of data placed in the MASTER_4 table, it was created in a compressed format. Its basic structure is shown in Figure 4.9:

MASTER_4				
PK_VALUE	NUMBER	NN	(PK)	(IX8)
DATE_1	DATE			(IX3)
DATE_2	DATE			(IX10)
DATE_3	DATE			(IX9)
CHAR_1	CHAR (1 Byte)			(IX2)
CHAR_2	CHAR (1 Byte)			(IX1)
CHAR_3	CHAR (1 Byte)			(IX5)
VALUE_1	VARCHAR2 (10 Byte)			(IX6)
VALUE_2	VARCHAR2 (10 Byte)			(IX4)
VALUE_3	VARCHAR2 (10 Byte)			(IX7)

***Figure 4.9* - Master 4 Table Structure**

Each table contains various columns, and each column in each table was individually indexed. The name PK_VALUE is for the primary key, and each table has a column with this same name. All other column names were similar and created to show their respective function. The PK_VALUE fields were populated with numerically ascending integers, and all other fields were populated with data as will be outlined below. The full DDL script for the creation of the table structure in this case study is included in Appendix 1.

The indexes on each column were created to adhere to the following criteria:

→ DATE columns were all individually indexed in a non-unique standard form.

→ CHAR columns were all individually indexed in a non-unique bitmap form.

→ VALUE fields were all individually indexed in a non-unique standard form.

→ All indexes on MASTER_4 were compressed, follow the same format as the above indexes, and were locally partitioned according to the table structure.

→ All indexes were named in a way to make them easily recognized with regards to their level, the table they were associated with, and their column name.

The Randomization of Data

Data in the tables was generated through a Perl script that randomized data in an arbitrary way. Date values were constrained between certain boundaries, the character fields were either a capital or lowercase letter or number, and value fields were simply a combination of character fields with the same criteria. All value fields possessed no spaces or special characters. Each of the MASTER_1, MASTER_2 and MASTER_3 tables were populated with 1,000,000 rows respectively, and the subsequent tables that had referential links back to these master tables were populated with 10,000 rows each. The MASTER_4 table differs in that it was populated with 30,000,000 rows, and value fields were populated with the first 10 characters from a list of 500 names that were generated arbitrarily. As mentioned above, compression was implemented on this MASTER_4 table as it consumed all of the available disk space, making index creation impossible. The code utilized for data generation is available in Appendix 2.

A Comment about Hints

At one time, "hints" were utilized heavily in development practices for Oracle developers. Niemiec (1999), in a book called "Oracle Performance Tuning," explained: "While the optimizer is incredibly accurate in choosing the correct optimization path and use of indexes for thousands of queries on your system, the optimizer is not perfect. Oracle has provided hints that you can specify for a given query so that the optimizer is overridden, and hopefully better performance is achieved for a given query" (p. 284). Niemiec dedicated several chapters to the understanding and utilization of hints in SQL development practices.

When hints are utilized in queries, they can dramatically change the execution pathway choice of the CBO. For example, Figure 4.10 gives a query that was executed with the "/*+ ORDERED */" hint:

```
SELECT /*+ ORDERED */
       COUNT (*)
  FROM Master_4
 WHERE      value_3 IN
               (SELECT value_3
                  FROM master_4
                 WHERE value_3 IN
                          ('Jenee',
                           'Hassie',
                           'Andrea',
                           'Danyell',
                           'Laree',
                           'Juliette',
                           'Eliene',
                           'Marie',
                           'Terry',
                           'Susie',
                           'Tina',
                           'Kareem',
                           'Amie',
                           'Levi',
                           'Eddie',
                           'Fleta'))
       AND date_1 IN
               (SELECT date_1
                  FROM master_4
                 WHERE date_1 BETWEEN TO_DATE ('01-JAN-2001', 'DD-MON-
YYYY')
                                  AND TO_DATE ('02-MAR-2002', 'DD-MON-
YYYY')
                UNION
                SELECT date_2
                  FROM master_4
                 WHERE date_2 NOT BETWEEN TO_DATE ('01-JAN-2001',
                                                   'DD-MON-YYYY')
                                      AND TO_DATE ('02-MAR-2002',
                                                   'DD-MON-YYYY'))
```

Figure 4.10 **– ORDERED Hint**

When executed against the MASTER_4 table, this query produced a cost of 546,976. However, when the hint was removed and this query was re-run, the execution plan changed markedly, and the cost reduced to 209,918.

The use of hints in current development practices is now discouraged. Kyte (2013) explained: "…do not use hints. They will be your last choice – never the first choice and never standard operating procedure. The problem is – we may well use your hints and your hints are just *wrong*, slower, precluding us from taking a better path. Hints – only useful if you are in the RBO and you want to make use of an access path" (p. 1). It is because of this shift in focus that this experiment does not utilize any hints: they are discouraged from use in development, and since the RBO ceased being the choice for optimization in Oracle 10g (Colgan, 2008), they have been avoided in this thesis.

The Index "Toggle" Method

Oracle 11g offers a way to utilize indexes for a given query, or actually turn them off. Kuntulu (2010) explained: "One of the new features of Oracle 11g, invisible indexes, is a perfect solution to the above problem [of when to use an index or not]. An index can be made invisible to the Optimizer, so that the optimizer can't see it. If an adhoc query requires the usage of the index it can explicitly specify the index as a part of the hint" (p. 1). Since we are not using hints, however, it is also possible to toggle the visibility of the index on or off with a separate DDL command. Syntax to make an index visible and invisible is shown in Figure 4.11:

```
--Turning the Index ON
ALTER INDEX master_4_char_1_idx VISIBLE;

--Turning the Index OFF
ALTER INDEX master_4_char_1_idx INVISIBLE;
```

Figure 4.11 **– Index "TOGGLE"**

Utilizing this functionality, namely Visible or Invisible Indexing, was a core feature of this experiment.

The Data Gathering Method

Queries in this experiment were created against the tables mentioned above with the express goal of avoiding full table scans when all indexes were visible, and utilizing indexes in such a way as to maximize their use overall. For example, the query in Figure 4.10, when run without the ORDERED hint, produced the following execution plan as shown in Figure 4.12:

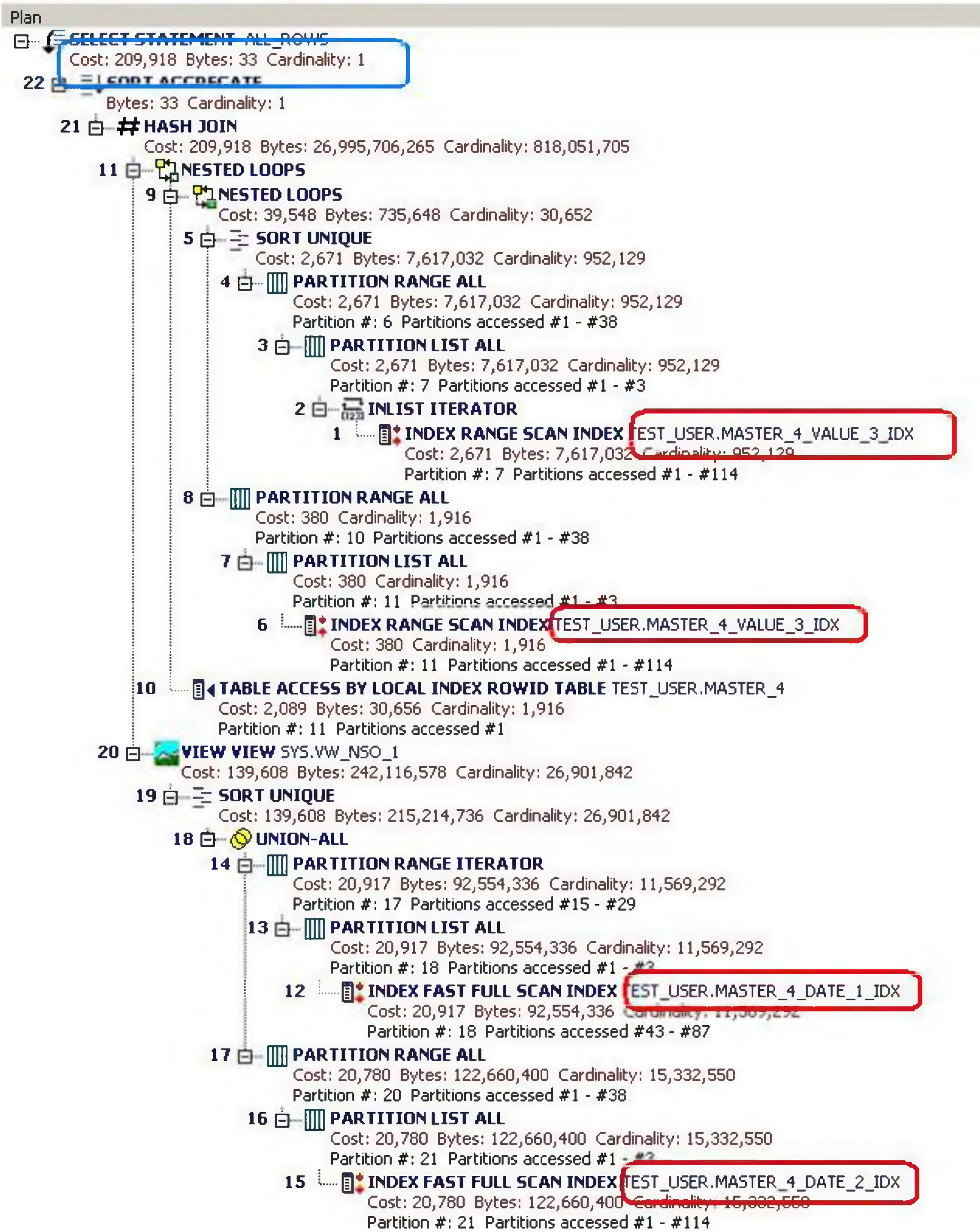


Figure 4.12 – **Execution Plan**

This EP has highlights to show that three separate indexes were utilized, and these indexes all correspond to the three predicate sections of the query in question. This EP also shows that the overall cost of this query was 209,918. Please notice the changes in the EP when one of the indexes, namely the MASTER_4_DATE_1_IDX is toggled off. This is illustrated in Figure 4.13:

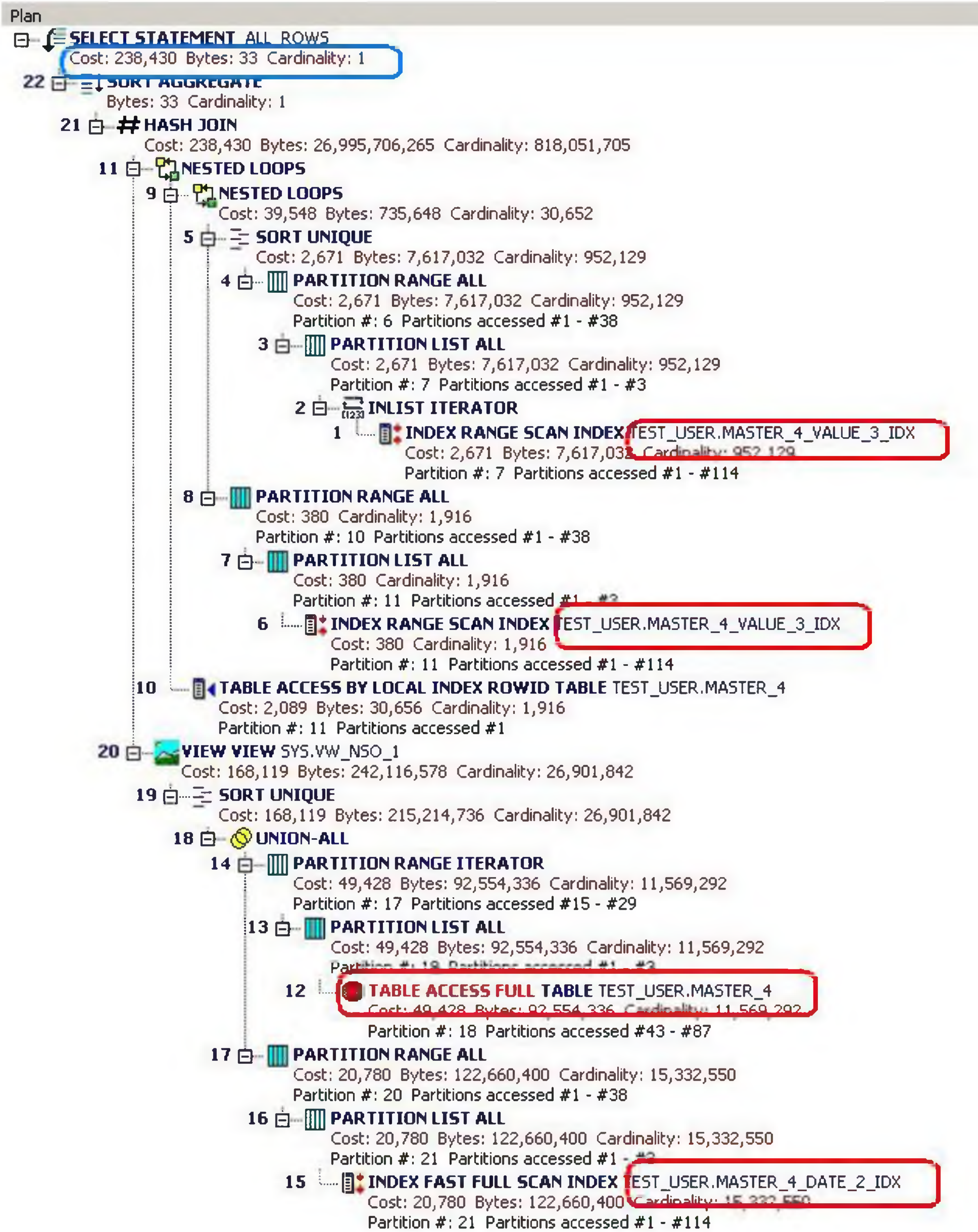


Figure 4.13 - **Revised Execution Plan**

This EP shows that all of the previous indexes were utilized with one exception—in place of the MASTER_4_DATE_1_IDX, a Table Access Full now resides. Also notice that the cost of this new EP is 238,430. This is slightly higher than the cost when the index was utilized before, and indicates that this new EP is less efficient than its predecessor.

Through a method of toggling indexes on and off, this experiment was able to create combinations for a myriad of possible EP pathways. For each "k" number of indexes, the formula for the possible combinations is given by: Combinations = 2^k. So, if there are 3 indexes, for example, there will be 8 combinations of ways they can be toggled on and off. An example in Figure 4.14 will make this clear:

Combinations:	Index_1	Index_2	Index_3
0	On	On	On
1	Off	On	On
2	On	Off	On
3	On	On	Off
4	Off	Off	On
5	Off	On	Off
6	On	Off	Off
7	Off	Off	Off

Figure 4.14 **– Index Toggle Combinations**

Methodology for Gathering Data

With the tables created and populated, and with indexes created on every field in each column, the following steps were followed to gather data:

1. The DBMS_STATS.GATHER_SCHEMA_STATS procedure was run to collect cardinality data, thus ensuring the optimizer had appropriate statistics on data distribution to function correctly.
2. The DBMS_STATS.GATHER_SYSTEM_STATS procedure was run during the initial runs of the data gathering portion of this study, to ensure that adequate statistics were gathered on the underlying hardware system, seek times, and overall system performance.
3. A series of queries were developed that parsed the tables in as random a fashion as could be generated by this researcher, the goal being that each query would utilize only indexes to gather data when executed manually with all indexes visible.
4. These queries were then passed to a Perl script that sequentially did the following:
    a. All indexes associated with the query were turned off to provide a default start.
    b. Each index on/off combination was explored as it related to the query, and the EP cost of the query in question was captured. If that cost was new (meaning not a duplicate of a previously captured cost for that query), the query was then executed for actual execution speed.
    c. The execution speed was captured, and the internal data dictionary of the RDBMS was queried to ensure that the original EP cost was the same as the actual execution cost when the query was run.
    d. Results were stored off to a file for further analysis.
    e. The next combination was tested, and the process was continued until all combinations were checked, or until the script was manually terminated for performance problems.

A total of 27 queries were generated against this database, 24 of which were written against the MASTER_1, MASTER_2, and MASTER_3 tables, or a combination of them and their associated STAR members. Only 3 queries were run against the MASTER_4 table, as the most interesting results were generated there, eventually needing script termination do to performance problems.

Each SQL file passed to the Perl script contained 4 sections. Those are illustrated in Figures 4.15, 4.16, 4.17 and 4.18 respectively:

```
--INDEXES
MASTER_1_CHAR_1_IDX
MASTER_1_CHAR_2_IDX
MASTER_1_CHAR_3_IDX
MASTER_1_DATE_1_IDX
MASTER_1_DATE_2_IDX
MASTER_1_DATE_3_IDX
MASTER_1_DATE_4_IDX
MASTER_1_DATE_5_IDX
MASTER_1_VALUE_1_IDX
MASTER_1_VALUE_2_IDX
```

Figure 4.15 **– INDEX Listing**

The first section of each SQL file contains a listing of the indexes that the associated query accesses. This gave the Perl script a definitive list of indexes to switch on or off as it checked through each combination of possible EP plans. In this example, there were ten indexes, so there were potentially 1024 different EP variants.

```
--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
select count(*)
from master_1
where date_1 in (select date_1 from master_1 where date_1 between to_date('01-JAN-1970','DD-MON-YYYY') and sysdate)
  and date_2 in (select date_2 from master_1 where date_2 between to_date('01-JAN-1971','DD-MON-YYYY') and sysdate)
  and date_3 in (select date_3 from master_1 where date_3 between to_date('01-JAN-1972','DD-MON-YYYY') and sysdate)
  and date_4 in (select date_4 from master_1 where date_4 between to_date('01-JAN-1973','DD-MON-YYYY') and sysdate)
  and date_5 in (select date_5 from master_1 where date_5 between to_date('01-JAN-1974','DD-MON-YYYY') and sysdate)
  and char_1 in (select char_1 from master_1 where char_1 between 'A' and 'Z')
  and char_2 in (select char_2 from master_1 where char_2 between 'A' and 'Z')
  and char_3 in (select char_3 from master_1 where char_3 between 'A' and 'Z')
  and (value_1 like 'a%' or value_1 like 'b%')
  and (value_2 like 'A%' or value_2 like 'B%')
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/
```

Figure 4.16 **– Initial Cost Capture**

This section of the SQL file is used to clear out old EP estimates, and then after executing the EP for the associated query, capture the actual cost of that query.

```
--TIME_COST
set timing on
select /*+ MASTER_1_QUERY*/
count(*)
from master_1
where date_1 in (select date_1 from master_1 where date_1 between to_date('01-JAN-1970','DD-MON-YYYY') and sysdate)
  and date_2 in (select date_2 from master_1 where date_2 between to_date('01-JAN-1971','DD-MON-YYYY') and sysdate)
  and date_3 in (select date_3 from master_1 where date_3 between to_date('01-JAN-1972','DD-MON-YYYY') and sysdate)
  and date_4 in (select date_4 from master_1 where date_4 between to_date('01-JAN-1973','DD-MON-YYYY') and sysdate)
  and date_5 in (select date_5 from master_1 where date_5 between to_date('01-JAN-1974','DD-MON-YYYY') and sysdate)
  and char_1 in (select char_1 from master_1 where char_1 between 'A' and 'Z')
  and char_2 in (select char_2 from master_1 where char_2 between 'A' and 'Z')
  and char_3 in (select char_3 from master_1 where char_3 between 'A' and 'Z')
  and (value_1 like 'a%' or value_1 like 'b%')
  and (value_2 like 'A%' or value_2 like 'B%')
/
exit
/
```

Figure 4.17 **– Execution Speed Capture**

This section of the SQL file performs two specific functions: it captures the actual execution time, but it also implements a comment with the "/*+ MASTER_1_QUERY*/" phrase following the opening "select". This comment has similar syntax to a hint, but is not one. It is utilized in the final process of this capture to cross check that the EP cost as was captured in code shown in Figure 4.16 will be the same during actual execution.

```
--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_1_QUERY%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
```

Figure 4.18 **– Final Cost Cross-Check**

This final section of the SQL file is used to capture the actual cost when the query was run. It is possible that running an EP on a query to estimate its cost might differ from the actual cost when the query is run. This code captures that final cost and stores that information for further analysis.

The code utilized for each query execution is attached in Appendix 3. The format for each is similar and not arranged in any particular order.

Perl Script for Data Gathering

The Perl script for data gathering is attached in Appendix 4. Each SQL file, as mentioned above, was opened by this script, and EP choices were investigated with index on/off toggling. When a new cost value was discovered for a combination, the query was then run with results captured. Values were stored in a character delimited flat file for importation into Microsoft Excel for review and analysis. Trends were then investigated, and the objectives of this case study were investigated.

Summary

This chapter outlined the objectives of the Oracle experiment of this thesis, and presented the methodology used, as well as the structure and tools of the system and software approach for data gathering and analysis. The next chapter examines the data collected.

Chapter 5 – The Optimizer Experiment

Up to this point, this researcher has laid the foundations for examining the design goals and significance of the CBO. The evolution and architecture of the CBO was introduced (Chapter 1), technical literature showing the maturation trends of the Oracle RDBMS which affect CBO speed and accuracy was reviewed (Chapter 2), an academic literature review showing the beginning interests in CBO output as well as possible challenges for CBO correlation was discussed (Chapter 3), and a methodology to test the hypothesis that correlations could be achieved was presented (Chapter 4). This researcher now turns to the analysis of the data collected from the experiment.

Two general database structures were utilized to collect the data: partitioned and non-partitioned. Both produced significantly different results, primarily as a result of the underlying amount of data found in both structures. With regards to the non-partitioned grouping, 23 queries were run, in both cached and explicitly non-cached scenarios, and results were mixed, primarily showing non-correlation, and in some cases, actually showing negative correlation. The speed with which data was returned was significant, and possibly skewed results. However, with regards to the partitioned data, correlation could be found in some situations, but in others, the results varied in such a way as to refute claims made by both Kyte and Lewis regarding the CBO cost having any value at all.

Experimental Preliminaries

The experimental portion of this thesis consisted of running the Data_Gatherer_Script as outlined in Chapter 4, and running statistical analysis on captured data. The script run took approximately 72 total hours between June 8 and June 25, 2013, which included restarts to validate consistency with runs and execution speeds. Data was then imported into an Excel spreadsheet, and statistical data computed. The order of execution is preserved in the following high level statistical data as presented in Figure 5.1:

Type	QUERY	Avg CS	Mdn CS	Mode	Count	Max CS	Min CS	Slope	Correl	Stdev
NF	GENERAL_JOIN_2	147.81	147	152	503	496	92	0.00	-0.14	25.25
NF	GENERAL JOIN 1	70.94	69	70	256	242	44	0.00	0.59	16.63
NF	GENERAL JOIN 3	10.13	6	6	8	34	4	0.00	-0.49	10.25
NF	MASTER 1 JOIN 3	213.88	212	217	119	353	176	-0.01	-0.60	21.22
NF	MASTER 1 QUERY	153.26	150	143	311	430	116	0.00	0.00	21.21
NF	MASTER 2 QUERY	136.47	119	122	174	749	68	0.00	-0.21	71.67
NF	MASTER_3_QUERY	490.40	350	253	80	9050	46	-0.10	-0.27	985.52
NF	MASTER ALL JOIN 4	376.89	228	#N/A	9	1214	202	-0.07	-0.38	326.20
NF	MASTER ALL JOIN 5	262.05	260	267	402	765	205	0.00	-0.01	35.84
NF	MASTER ALL JOIN 6	290.41	287	277	127	434	267	0.00	0.04	18.78
NF	MASTER ALL JOIN 7	284.18	285	295	310	447	234	0.01	0.41	24.66
NF	GENERAL JOIN 4	36.17	40	17	24	143	14	0.01	0.59	26.73
NF	GENERAL_JOIN_5	193.00	193	#N/A	2	240	146	-0.03	-1.00	66.47
NF	GENERAL JOIN 6	85.58	49	43	12	301	5	0.02	0.44	96.68
NF	GENERAL JOIN 7	18.38	9	3	382	37	2	0.00	0.12	15.55
NF	GENERAL JOIN 8	103.44	45	3	16	751	1	0.01	0.23	187.58
NF	GENERAL JOIN 9	56.71	44	3	16	96	1	0.01	0.78	45.16
NF	MASTE 1 JOIN 1	99.26	97	97	58	232	83	0.00	-0.25	18.76
NF	MASTER_JOIN_2	194.10	190	195	30	399	161	-0.01	-0.47	41.10
NF	MASTER 3 QUERY 3	50.00	28	#N/A	3	117	5	0.03	0.98	59.15
NF	MASTER ALL JOIN 1	78.00	77	76	64	169	60	0.00	-0.18	13.38
NF	MASTER ALL JOIN 2	214.80	209	211	256	634	176	0.00	-0.33	44.70
NF	MASTER ALL JOIN 3	299.44	211	219	160	5052	172	-0.05	-0.36	444.78
FL	GENERAL JOIN 2	180.47	171	163	503	1807	123	-0.01	-0.24	84.61
FL	GENERAL JOIN 1	83.35	82	78	256	250	56	0.00	0.57	17.53
FL	GENERAL JOIN 3	15.50	11	11	8	47	7	0.00	-0.49	13.44
FL	MASTER 1 JOIN 3	226.39	225	226	119	467	182	-0.01	-0.50	28.17
FL	MASTER 1 QUERY	156.94	155	156	311	455	120	0.00	0.06	20.46
FL	MASTER 2 QUERY	147.98	136	129	174	1298	87	0.00	-0.21	110.22
FL	MASTER_3_QUERY	536.41	406	395	80	8149	70	-0.10	-0.28	881.26
FL	MASTER ALL JOIN 4	1065.11	1098	#N/A	9	1421	295	-0.08	-0.38	358.80
FL	MASTER ALL JOIN 5	595.74	602	347	402	1326	275	0.00	0.07	263.74
FL	MASTER ALL JOIN 6	577.50	559	354	127	911	334	0.16	0.74	185.89
FL	MASTER ALL JOIN 7	735.23	560	337	310	2104	300	-0.02	-0.06	430.27
FL	GENERAL JOIN 4	54.08	52	28	24	270	24	0.01	0.33	49.33
FL	GENERAL_JOIN_5	180.00	180	#N/A	2	241	119	-0.04	-1.00	86.27
FL	GENERAL JOIN 6	56.17	31	5	12	400	2	0.00	-0.06	109.61
FL	GENERAL JOIN 7	19.01	18	3	382	39	2	0.00	0.11	15.87
FL	GENERAL JOIN 8	229.44	48	5	16	2379	1	0.01	0.07	587.66
FL	GENERAL JOIN 9	53.19	51	2	16	111	1	0.01	0.74	51.73
FL	MASTE 1 JOIN 1	103.40	99	90	58	381	89	0.00	-0.24	37.84
FL	MASTER JOIN 2	214.33	193	209	30	823	161	-0.03	-0.35	115.99

FL	MASTER 3 QUERY 3	531.00	468	#N/A	3	1115	10	-0.03	-0.10	555.19
FL	MASTER ALL JOIN 1	80.75	80	80	64	148	67	0.00	-0.19	9.63
FL	MASTEr ALL JOIN 2	225.75	218	210	256	860	194	0.00	-0.21	58.38
FL	MASTER ALL JOIN 3	297.11	266	265	160	3195	230	-0.02	-0.21	244.24
NF	MASTER 4 QUERY 3	602167.28	25672	#N/A	50	2549082	7283	6.83	0.63	806907.62
NF	MASTER 4 QUERY 1	38007.50	32423	#N/A	6	66330	12314	0.11	0.98	20769.90
NF	MASTER 4 QUERY 2	84956.01	79419	#N/A	192	256432	34356	0.06	0.55	34228.19

***Figure 5.1* – Experimental statistical results**

The columns of this spreadsheet are described below:

Type → "FLUSHED" or "NOT FLUSHED". This will be defined in more detail below.

QUERY → The query that was run during this particular iteration, with run order preserved.

Avg CS → The average number of centiseconds associated with the query.

Mdn CS → The median number of centiseconds associated with the query.

Mode → The mode of the centiseconds for the query.

Count → The number of individual index-toggle combinations that were run for the query.

Max CS → The maximum number of centiseconds for the overall query results.

Min CS → The minimum number of centiseconds for the overall query results.

Slope → The slope of the line of best fit for correlating time to cost.

Correl → The correlational coefficient of the cost and execution speed for the query results.

Stdev → The standard deviation of the execution speeds for the query results.

Mention was made to "centiseconds." The output from the Oracle "Set Timing On" SQLPlus command is given in hours, minutes, seconds, and then hundredths of seconds. To use the highest level of granularity and accuracy, all captured speeds were converted into hundredths of seconds, or centiseconds. Also, the "Count" number, which came from the index-toggle combinations of each query, represents the number of samples used in each statistical calculation.

Non-Partitioned Queries

Several interesting things presented themselves with regards to the non-partitioned samples. First of all, when several of the queries were run the first time, a noticeable increase in response time appeared when compared to the reminder of the responses during subsequent runs. For example, GENERAL_JOIN_2 took 2.42 seconds when run the first time, and all subsequent runs were less than a second in duration. Similar happenings occurred with MASTER_1_JOIN_3, MASTER_ALL_JOIN_4, and others. This is not unusual because generally, the first time a query is run, data is often pulled from disk as opposed to memory, and data retrieval speeds from disk are usually much slower than memory retrievals. This result was not consistent between each query, however, and prompted an adjustment during the experimental portion of the thesis: a "FLUSHED" and "NOT FLUSHED" apparatus was employed.

Chapter 4 showed that each SQL file contained four general sections: a list of indexes, an initial explain plan cost capture, an actual execution speed capture, and a final run-time cost capture. After data was gathered from the original 23 queries in the non-partitioned area of this database, an adjustment was made to each of the 23 queries: all were re-written, and the line "alter system flush buffer cache" was added to the second and third portions respectively of each. At this point, the data capture was again run with these 23 queries. However, this time, the adjusted SQL commands cleared data out of memory between each SQL statement. This forced the RDBMS engine to repeatedly retrieve data from disk during each run (Burleson, 2008). So, in the case where data was left in memory between each SQL statement, it has been denoted "NOT FLUSHED", and when removed, "FLUSHED". Surprisingly, it was expected that this change would cause each query to show a noticeable slowing in response time during each combinatorial iteration, but this was not the case.

Another interesting finding was the general speed of each of these queries. The longest running queries in the NOT FLUSHED area were MASTER_3_QUERY at 9050 cs, and MASTER_ALL_JOIN_3 at 5052 cs. The average of both of these queries was respectively 4.9 and 2.9 seconds, which would still indicate to this researcher a relatively fast performance. In the FLUSHED area, the same queries were also the longest running at 8149 cs, and 3195 cs respectively. But it was surprising that FLUSHED queries would actually perform faster than those that contained some data in memory. The averages on these were 4.1 and 2.9 seconds respectively. Considering these queries pulled data from tables that contained 1,000,000 rows, and regardless of whether the data was in memory, the queries performed consistently under 10 seconds, and the overall average was 161 cs and 265 cs respectively for FLUSHED and NOT FLUSHED.

Correlational Results – NOT FLUSHED

In the NOT FLUSHED area, nine queries resulted in positive correlations, the most significant being 0.78 from 16 samples on GENERAL_JOIN_9, and the least significant being 0.04 from 127 samples on MASTER_ALL_JOIN_6. MASTER_3_QUERY_2 had the overall highest positive correlation, but was discarded because of its limited sample size of only 3. Of the original 23 queries, 13 samplings had negative correlation. Graph results from the NOT FLUSHED sampling are available for review in Appendix 5.

The mix of results from the NOT FLUSHED samples does not support high correlation. The original intent of this thesis was to ascertain the following: do lower cost queries result in faster response time; can individual queries be correlated with cost; and finally, can all queries together be correlated to cost. Results from this sampling suggest that these things are not consistently possible.

To illustrate the concerns with this first test objective, the following graphs are provided in Figures 5.2, 5.3, and 5.4:

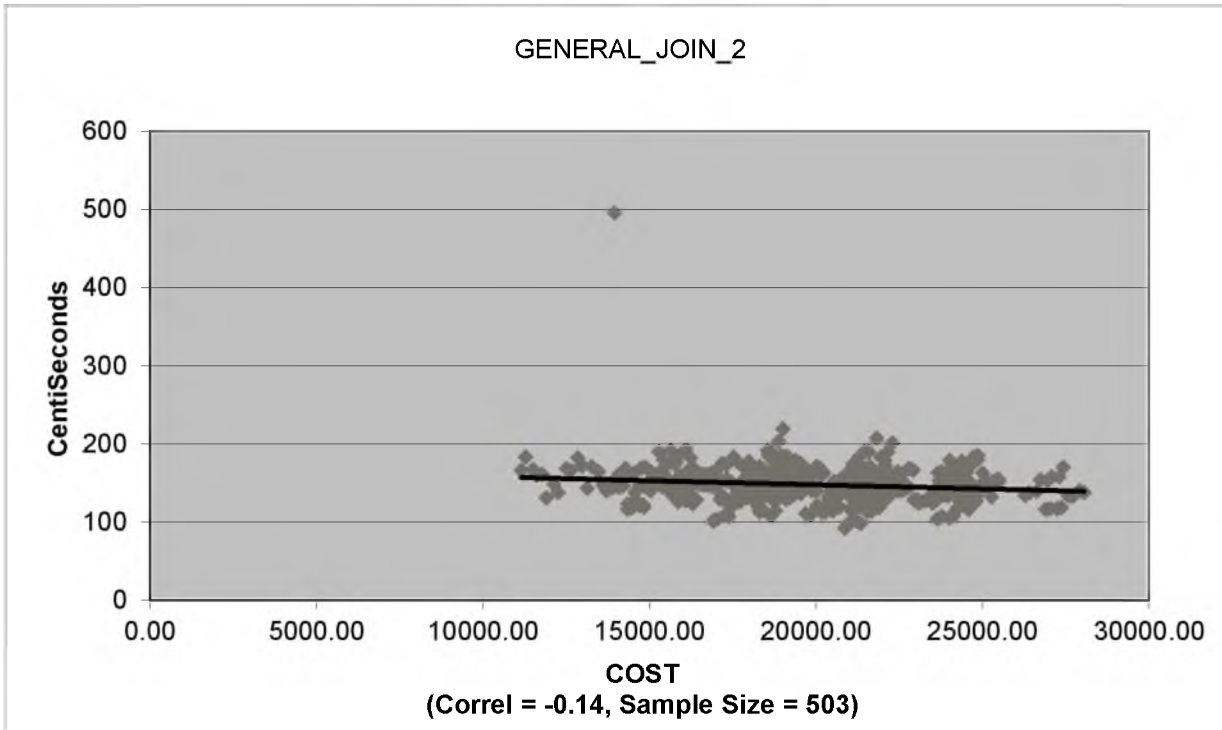


Figure 5.2 – **NOT FLUSHED General_Join_2**

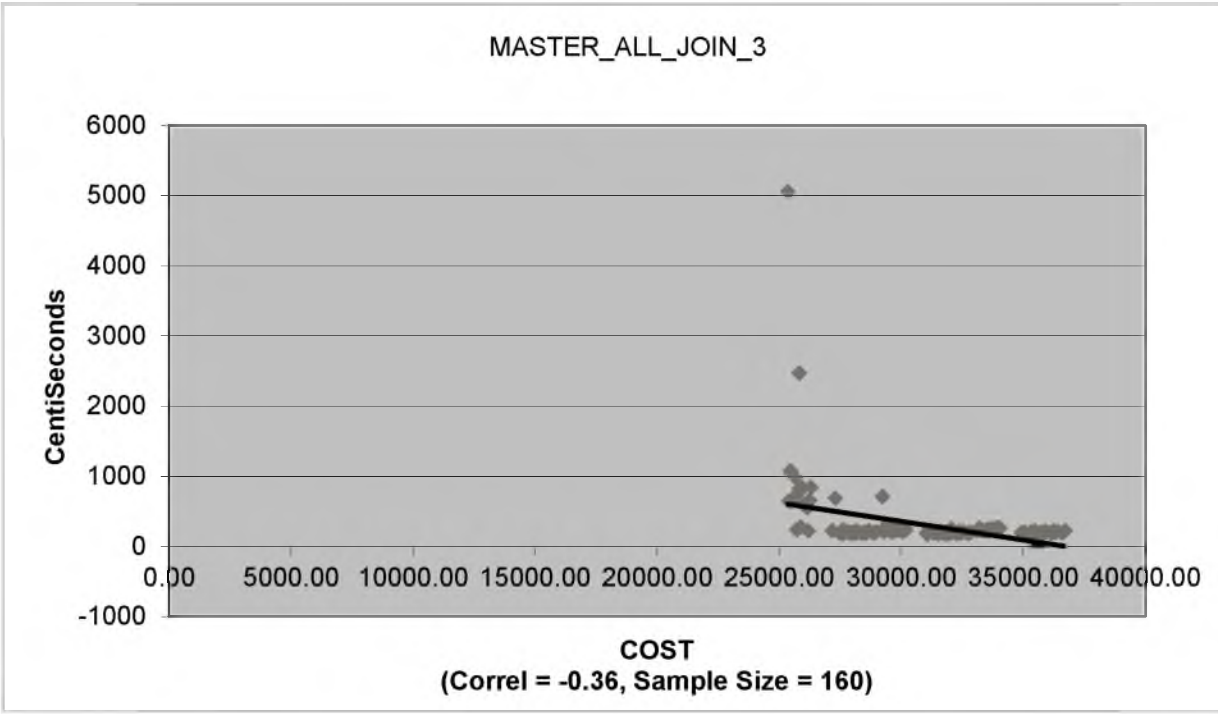


Figure 5.3 – **NOT FLUSHED Master_All_Join_3**

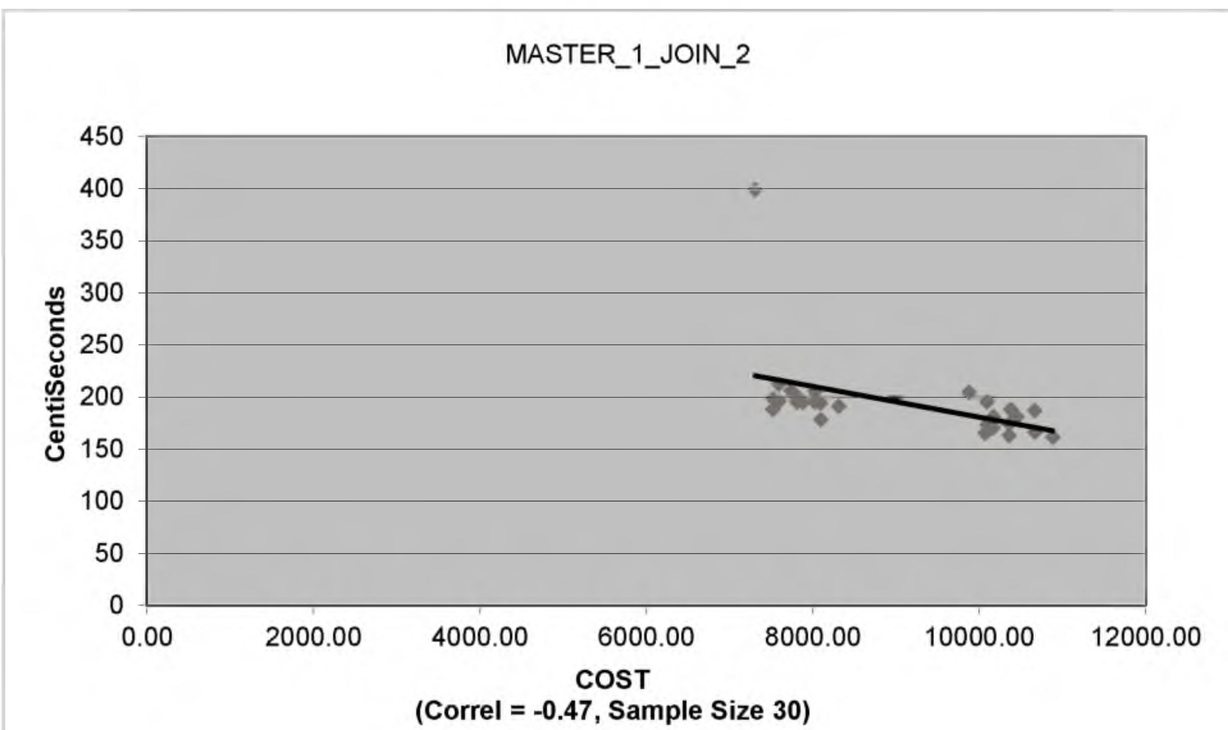


Figure 5.4 – **NOT FLUSHED Master_1_Join_2**

Each of these queries showed a negative correlation: as the COST has increased, the speed of the query decreased. This was counter-intuitive: the higher cost the query, the slower it is expected to operate. In real world terms, developers would be misled using the CBO estimates for these particular queries. As they attempted to lower the overall cost, they would in fact be slowing the queries down.

Time based correlations were meaningless on many of the queries when the overall time scale was taken into account. For example, GENERAL_JOIN_7 in Figure 5.5 has a range between 2 and 37 cs. Regardless of the disparity of data, all the groupings happen in less than a fraction of a second, and so cost, in this situation, is not important. In real world application, any query that returns results in less than a second is considered fast.

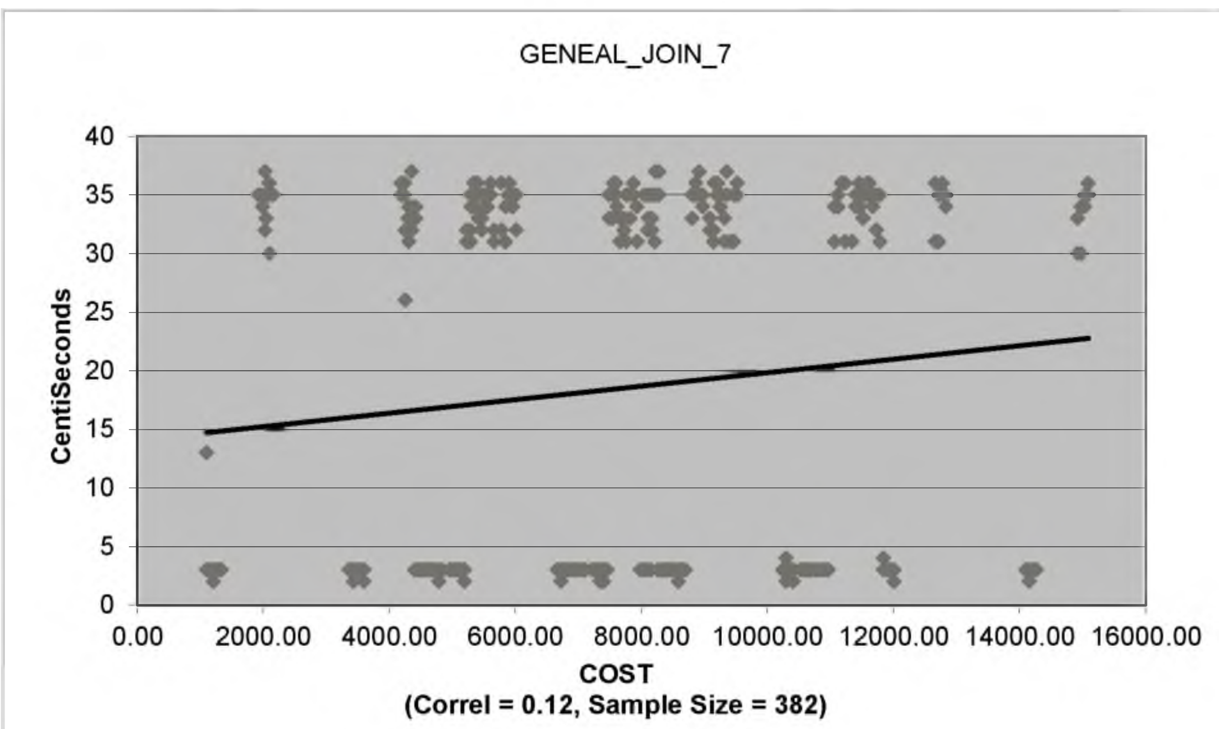


***Figure 5.5* – NOT FLUSHED General_Join_7**

GENERAL_JOIN_9, shown in Figure 5.6, also had a very small time range, from 1 to 96 cs. This particular query contained groupings that were confusing. For example, at the approximate cost of 7000, the resultant speed landed above the trend line. But, when the cost rose to approximately 8000, the resultant speed landed below the trend. This is completely opposite of what would be expected. Furthermore, at 8000, the result speeds was very fast. Regardless of this confusing data spread, all of the queries ran very fast, making predictability less important.

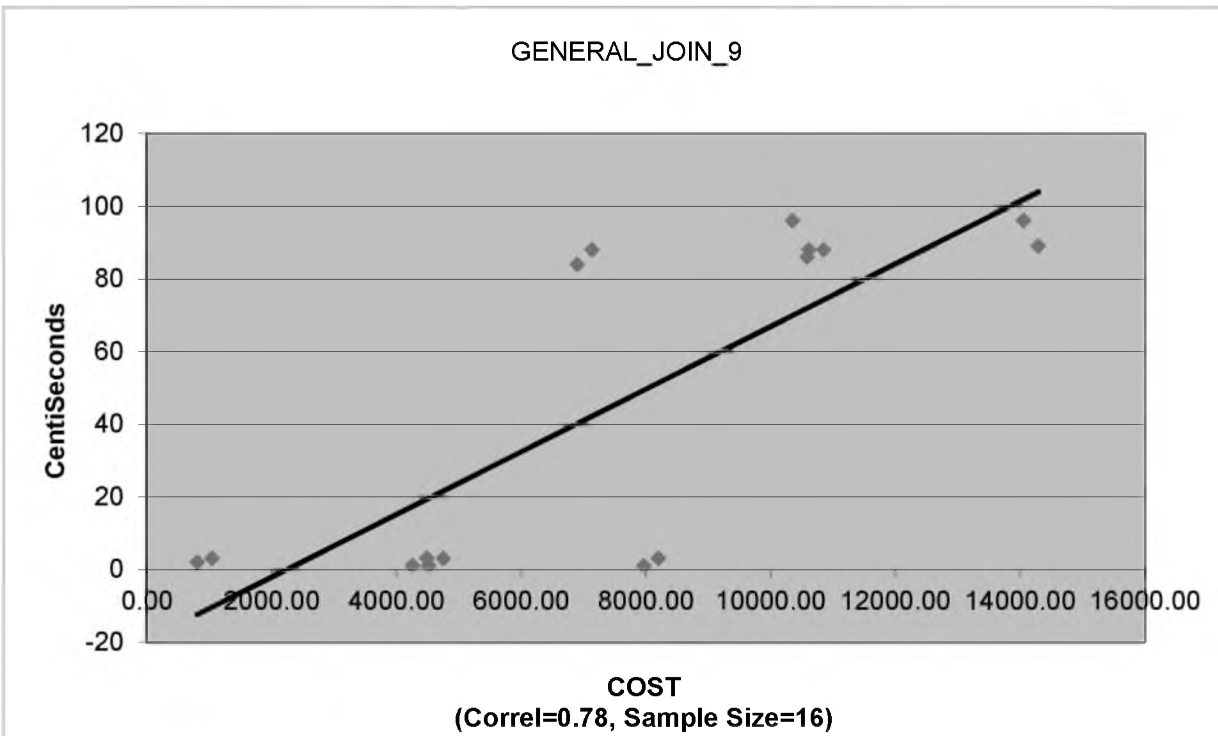


Figure 5.6 – **NOT FLUSHED General_Join_9**

Correlational Results – FLUSHED

In the FLUSHED area, eight queries resulted in positive correlations, the most significant being GENERAL_JOIN_9 and MASTER_ALL_JOIN_6 at 0.74 each, and 16 and 127 data samples respectively. Of all queries in this area, 15 had negative correlations. Again, MASTER_3_QUERY_2 was discarded do to its small sample size, but surprisingly, in the FLUSHED sampling, this query actually had a negative correlation. Graph results from the FLUSHED sampling are available for review in Appendix 6.

FLUSHED samples were similar to the NOT FLUSHED, and didn't support correlation either: the problems were virtually identical. The graphs in Figures 5.7, 5.8, and 5.9 illustrate negative correlations, which again violate the assumption that higher cost queries perform more slowly than those with lower costs:

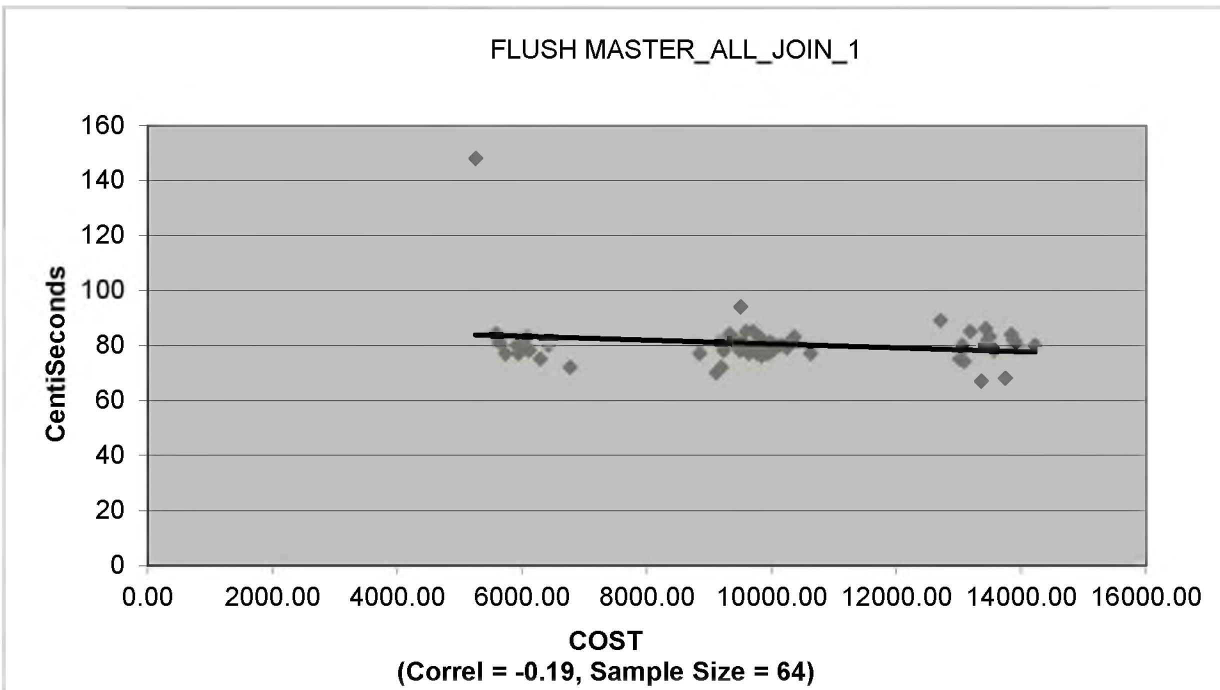


Figure 5.7 – **FLUSHED Master_All_Join_1**

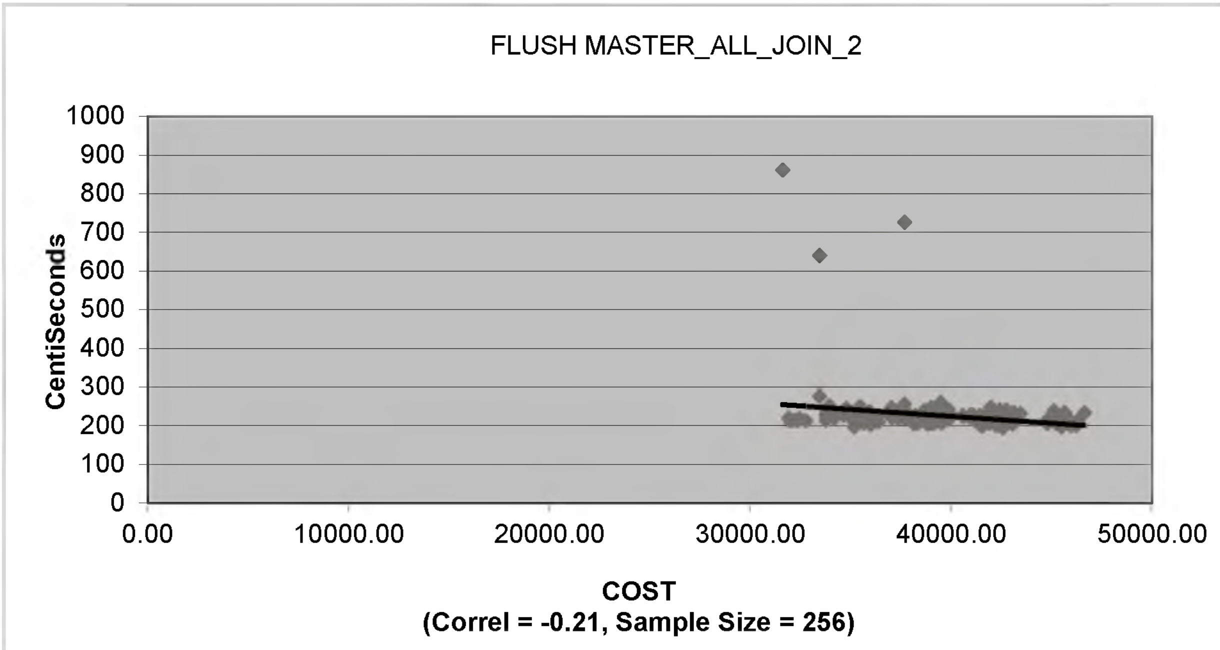


Figure 5.8 – **FLUSHED Master_All_Join_2**

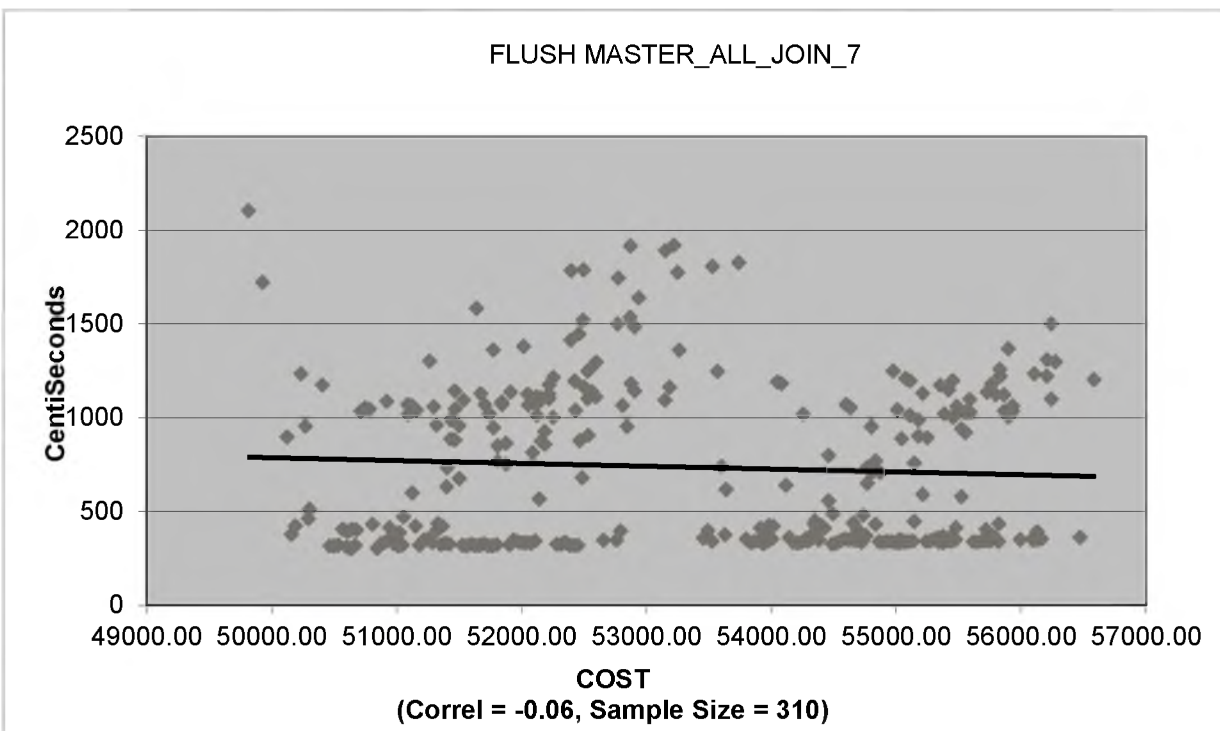


Figure 5.9 – **FLUSHED Master_All_Join_7**

As with the NOT FLUSHED results, those in the FLUSED sampling also suffered from significant variability. The scale of execution was also very shallow in many of them. Again, Figure 5.10 above is a good example of this: the standard deviation of this sampling is 430.27.

Partitioned Sampling

Three queries were run against the partitioned table MASTER_4. Significant performance differences were noted as the minimum value for overall speed came from MASTER_4_QUERY_3 at 7283 cs, but the maximum value from the same query was 2549082 cs, which is a little over 7 hours. Correlations for all of these queries were positive. MASTER_4_QUERY_1 had the highest correlation, and is illustrated in Figure 5.10:

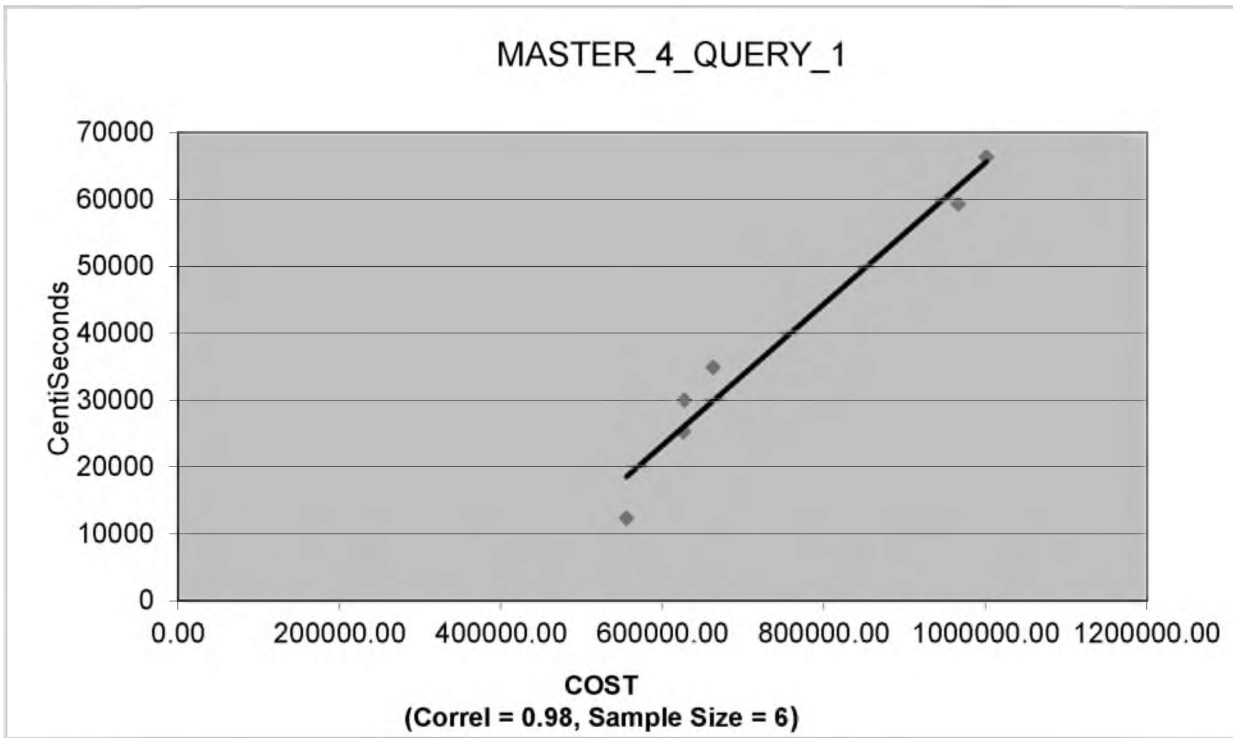


Figure 5.10 **– Master_4_Query_1**

MASTER_4_QUERY_2, shown in Figure 5.12 below, presented challenges related to potential confidence as cost values grew. Notice that the data variability spread away from the trend line as the cost increased. Fortunately, the end of the trend also represents the most inefficient possibilities with regards to index usage for this particular query—all of the indexes had been toggled off, and full table scans were the result. This mean there would be no way to further degrade this query without a complete rewrite or the introduction of hints to change the CBO execution pathway.

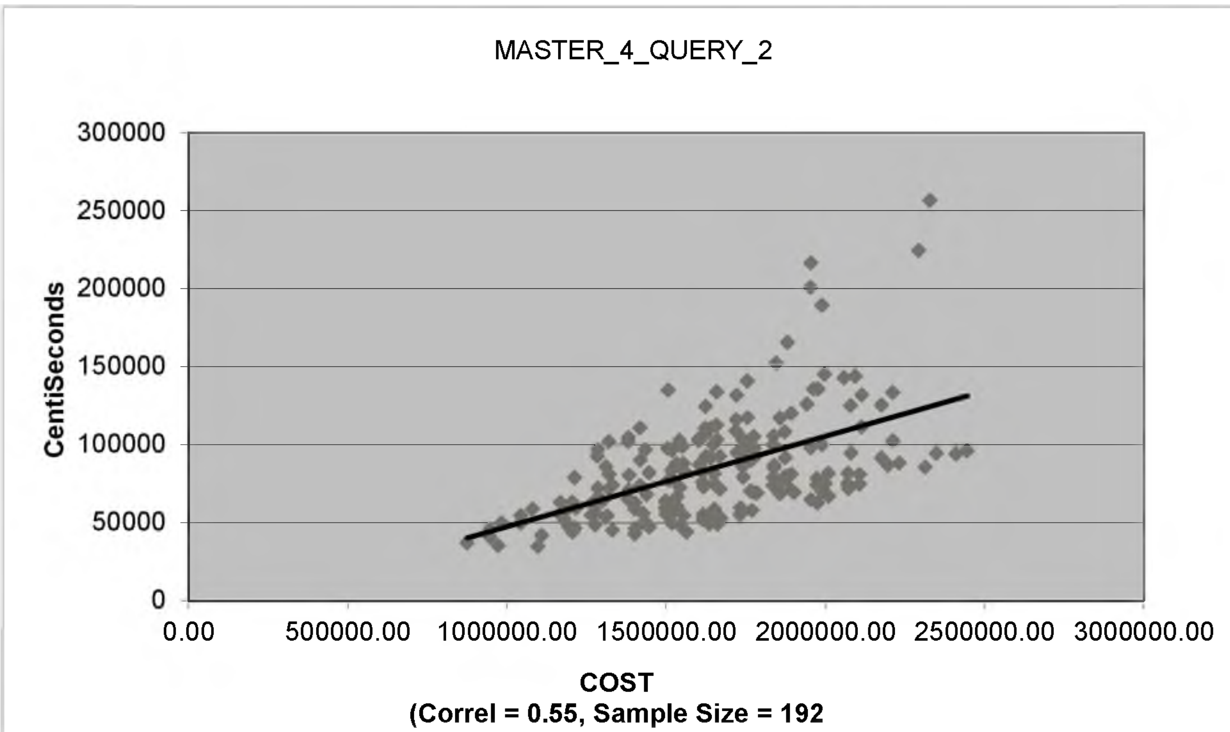


Figure 5.11 – **Master_4_Query_2**

MASTER_4_QUERY_3 was the most puzzling of all the partitioned queries. Data showed that the response time scale in centiseconds for all index-toggle combinations from 0 through 15 were in a range that made sense. However, from combinations 16 through 29, highlighted in yellow for illustration purposes, the scale shifted from minutes for each query to several hours. The illustration of this information is presented in Figure 5.12:

Query	EP Cost	Actual Cost	Time
MASTER_4_QUERY_3.sql	142271.00	142271.00	00:05:02.93
MASTER_4_QUERY_3.sql	165010.00	165010.00	00:01:15.73
MASTER_4_QUERY_3.sql	149282.00	149282.00	00:01:17.13
MASTER_4_QUERY_3.sql	166219.00	166219.00	00:02:33.19
MASTER_4_QUERY_3.sql	255481.00	255481.00	00:02:09.42
MASTER_4_QUERY_3.sql	278220.00	278220.00	00:02:23.41
MASTER_4_QUERY_3.sql	262492.00	262492.00	00:02:17.09
MASTER_4_QUERY_3.sql	279429.00	279429.00	00:03:16.13
MASTER_4_QUERY_3.sql	145474.00	145474.00	00:02:42.93
MASTER_4_QUERY_3.sql	168213.00	168213.00	00:02:48.64
MASTER_4_QUERY_3.sql	152486.00	152486.00	00:02:43.89
MASTER_4_QUERY_3.sql	169422.00	169422.00	00:04:20.42
MASTER_4_QUERY_3.sql	258684.00	258684.00	00:03:44.32
MASTER_4_QUERY_3.sql	281423.00	281423.00	00:03:55.03
MASTER_4_QUERY_3.sql	265696.00	265696.00	00:03:45.48
MASTER_4_QUERY_3.sql	282633.00	282633.00	00:05:21.40
MASTER_4_QUERY_3.sql	260428.00	260428.00	03:59:18.90
MASTER_4_QUERY_3.sql	283167.00	283167.00	06:24:39.72
MASTER_4_QUERY_3.sql	267440.00	267440.00	04:21:53.33
MASTER_4_QUERY_3.sql	284377.00	284377.00	04:20:47.09
MASTER_4_QUERY_3.sql	373638.00	373638.00	03:59:04.63
MASTER_4_QUERY_3.sql	396377.00	396377.00	04:01:18.96
MASTER_4_QUERY_3.sql	380650.00	380650.00	04:05:14.04
MASTER_4_QUERY_3.sql	397587.00	397587.00	04:46:34.88
MASTER_4_QUERY_3.sql	263631.00	263631.00	04:16:55.70
MASTER_4_QUERY_3.sql	286371.00	286371.00	04:07:26.00
MASTER_4_QUERY_3.sql	270643.00	270643.00	03:56:47.55
MASTER_4_QUERY_3.sql	287580.00	287580.00	03:57:28.01
MASTER_4_QUERY_3.sql	376841.00	376841.00	05:21:17.06
MASTER_4_QUERY_3.sql	399581.00	399581.00	04:49:26.98

Figure 5.12 – **Master_4_Query_3 Data**

The results of this query were checked multiple times and duplicated consistently. Many additional combinations could have been tested, but were not because the cost of each additional combination became labor intensive on the hosting machine. This query was manually terminated after the above information was captured.

These results produced confusion on at least two levels. The first was discussed with the sudden jump in performance degradation. The second involved the starkly contrasting disparity between cost estimates and execution speeds. For example, combination 15 had an actual cost of 282633, and an execution time of 00:05:21.40 which is a little over 5 minutes. However, the very next combination of 16 had an actual cost of 260428 which is less than its predecessor, but the execution time is 3:59:18.90, almost 4 hours. This is a significant difference.

The first 17 combinations in this query execution toggled off indexes associated with CHAR_1, CHAR_2, CHAR_3, and DATE_1 of this table. On all of the final combinations, the DATE_2 index was toggled off. The range partition model for this table was on DATE_1 and sub-partitioned on CHAR_1; DATE_2 was not involved in the structural creation of this table except as a common field. It is unknown why turning off the index of DATE_2 would cause issues. The graphical representation of MASTER_4_QUERY_3 is show in Figure 5.13:

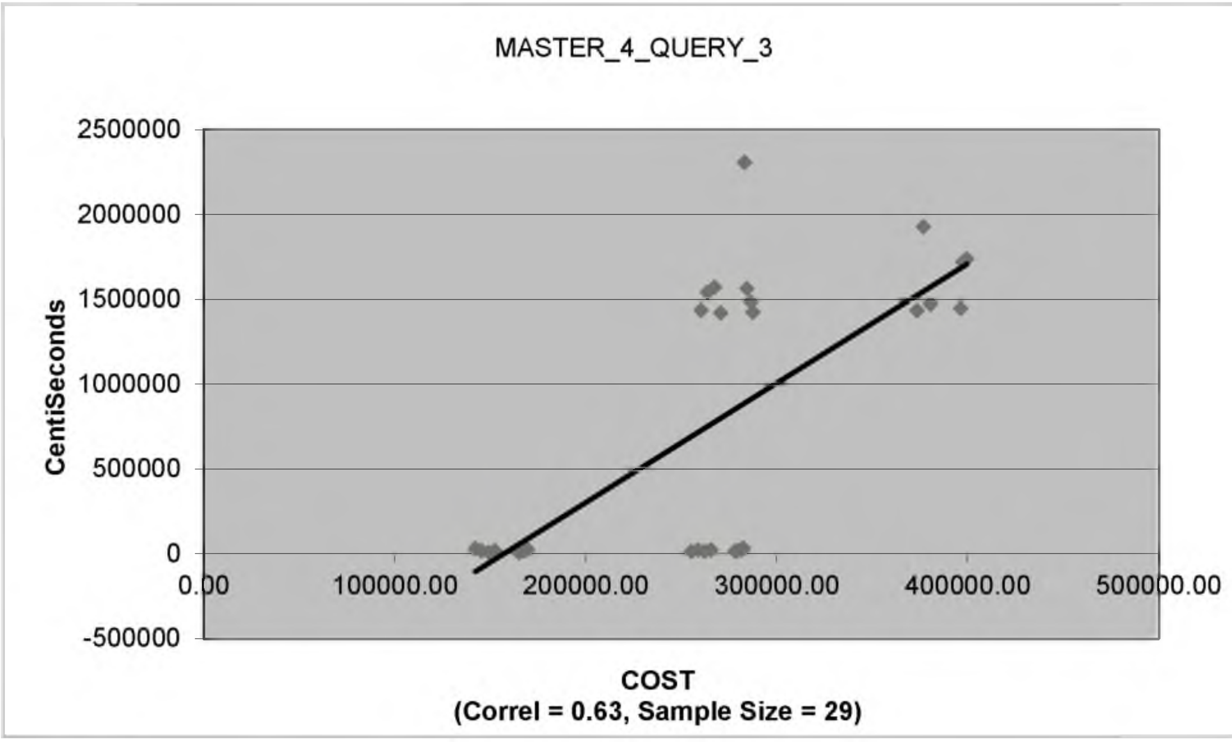


Figure 5.13 – **Master_4_Query_3**

Summary

This Oracle experiment did support some of the claims made by Kyte, but none of the claims of Lewis. Kyte's arguments were more restrictive than Lewis. Kyte argued CBO estimates were arbitrary numbers that could only be applied to a single query, and were only used by the CBO as a way to judge query efficiency. Also, he noted that cost could not be correlated to execution speeds, and especially not used to correlate between separate queries. Correlational findings in this experiment did not support cross query correlations with regards to speed predictability, which is in agreement with Kyte. However, the disparity with simply having a lower cost estimate for a query execution plan perform consistently faster than a higher cost for the same query is in opposition to Kyte. If the CBO estimate helps the database choose a more optimal plan, than it is reasonable to assume the more optimal plan would perform faster.

The above chapters established this. As shown, however, many of the queries actually had negative correlations, and the query results from MASTER_4_QUERY_3 in particular showed that even in positively correlated results, lower cost queries may perform significantly worse. Additional research is needed to understand how and why CBO costs provide inconsistent results, deviating away from expected performance objectives. Suggestions for additional research have been provided in the next chapter.

Chapter 6 – Conclusion

This researcher believes this project contributes to the sparse literature on CBO correlations. No attempts to correlate CBO estimates to execution speed were found in either the trade or academic literature, and therefore this study should be considered as an early attempt at understanding the feasibility of such correlations, as well as a possible starting point for further research efforts. Researchers have made reference to the desirability of predictive CBO correlations. Lewis was one of the main proponents of this. Despite this need, however, it was surprising to this researcher that findings discovered from experimentation showed all of the claims of Lewis as unsupportable, and only one of the claims of Kyte could be supported.

Different Approach Possibilities

In designing this research, various validating procedures and controls were utilized. The overall experimental run was performed three times, ensuring consistency of results. Furthermore, steps were taken to ensure that the underlying machine used during the experiment was completely isolated and idle from all outside users or non-Oracle processes. The machine itself was completely dedicated to this research.

Several challenges presented themselves during the experiment that could be addressed differently in subsequent attempts:

1. Machine memory was limited. This affected buffer-cache hit ratios. For example, when running the non-partitioned sections of the experiment, buffer-cache hit ratios only reached a 75% level, meaning significant I/O was still required for each subsequent query iteration. If memory was augmented and buffer-cache hit ratios significantly increased, this might influence outcomes.

2. Disk space was limited. This affected the way the partitioned data was stored, requiring compression. No attempts were made to look at CBO correlations in non-compressed partitioned data.

3. Limited disk space also created limits to the amount of data which could be stored in the non-partitioned tables. Had more disk space been available, increased data could have been stored, which might have affected the overall results of the experiment.

4. Query creation was difficult. Each query was created in an attempt to utilize individual indexes and avoid full-table scans when the indexes were visible. However, this objective required significant trial and error. Additional options for queries that would have returned more data, and perhaps taken longer to run, particularly in the non-partitioned area, would have been useful.

5. The data structure of the tables in the non-partitioned area was quite heterogeneous. For example, the date fields were all the same, character fields were the same, and the varchar2 fields had the same length. When an index was turned off for one char field, for example, it often had the exact same effect on the CBO estimate as if that original index were left on, and the next index on a similar char field were turned off. This added complexity to the query design as simply turning off an index didn't always change the CBO estimate.

6. The number of index toggles was limited to ensure timely completion of the experiment. At most, only 12 indexes were used on any one query, resulting in at most 4096 iterative runs. Had additional time been available, more data could have been gathered.

These challenges were part of the experimental portion of this thesis. However, the largest challenge this researcher encountered was the extremely fast results on virtually all of the queries in the non-partitioned area. When dealing with sub second result times, for example, it was difficult to ascertain if correlations had any meaning at all since query predictability becomes less important in such circumstances. Had the data been augmented, the heterogeneous structure of the data structures randomized differently, and the queries changed to increase the overall result set, it might have been possible to produce slower results. This would have made correlation more meaningful.

Associated with this speed problem was the lack of disparity with the individual results themselves. With the exception of the first query run which sometimes produced a slower result time, queries tended to clump together into pockets of completion time. This made correlation more difficult as the expectation was that when all indexes were utilized, queries would be fast. When all the indexes were toggled off, however, results didn't suffer as expected. They remained clumped with the others. It was in reaction to this finding that this researcher decided to augment the testing to include the "FLUSHED" approach outlined in Chapter 4. Again, it was surprising that these new results, which completely circumvented the buffer-cache area, could actually perform better.

Suggestions for Further Study

Several additional questions arose through this study, and present areas of further research. Some of these include:

1. What role does the number of rows in data structures play in the correlational potential of non-partitioned data? In the case of this research, the non-partitioned results were fast, in the centisecond range on most queries. Three general questions arise:

   a. Why were query results so fast when it appears these tables had significant numbers of rows, and when buffer-cache hit ratios were not taken into consideration as shown by data being flushed from memory and the queries rerun?
   b. If the amount of data were significantly increased in these tables, would this cause the values of each query to jump to a higher time factor, and would that allow for data spreading in such a way to better correlate?
   c. What role would buffer-cache hit ratios play in these queries if all of the data could be held in memory, and would there be a significant performance difference when cached results were flushed? In the case of this research, for example, flushed results and non-flushed results were generally very consistent, as though memory cache had little effect at all.
2. What is the role of non-key columns in partition tables, and why can turning off single indexes have such a dramatic effect on performance when other non-key indexes don't have such an effect? Four general questions arise here:
   a. If non-key indexes have dramatic effect on partitioned tables, would different index structures such as composite indexes provide superior results, and how would you know when to use them?
   b. Why would turning off a standard index have such a dramatic effect, but turning off a bit-map index not produce similar effects? The index in question was on a date field. Would similar problems arise if an index was turned off on a character field, and if so, why?

c. What affect does caching or non-caching of data have on partitioned tables? In this research, no attempt was made to flush memory as performance degradations were already significantly hampered, simply when specific indexes were turned off.

d. If correlation is not possible on cached data, could it be affected on non-cached data?

e. If non-cached correlation were possible, would it apply to non-partitioned tables of a certain size, partitioned tables of a certain size, or both?

3. If correlation is not possible, even when using multiple cost estimates for the same query, then why does the EP still exist, and why is it a tool suggested by Oracle as a way to investigate and tune queries?

4. Were there other factors, such as a bug, a missing patch, or a specific scenario type anomaly that affect this thesis' research in such a way as to produce false results? Can this research be duplicated?

Other topics might present themselves as this list is not exhaustive. However, these suggestions do open up additional questions that have not been answered with regards to CBO cost correlations.

Final Comments

This thesis proceeded on the assumption that CBO correlations might be possible. Results suggest this is not currently the case, at least not with any degree of predictability. However, this is a single study, and additional work to corroborate or refute the findings of this study is needed. From a development perspective, it is the hope of this researcher that CBO correlations will eventually become an integral and accessible part of the Oracle architecture.

References

Ahmed, R. (2010). Query processing in Oracle DBMS. *Paper presented at Processing of the ACM 13th International Workshop on Data Warehousing and OLAP. Toronto.* doi: 10.1145/1871940.1871942

Ahmed, R., Lee, A., Witkowski, A., Das, D., Su, H., & Zait, M. et al. (2006). Cost-based query transformation in Oracle. *Paper presented at Proceedings of the 32nd international conference on very large data bases. Seoul.*

Antoshenkov, G., & Ziauddin, M. (1996). Query processing and optimization in Oracle RDB. *The VLDB Journal - The International Journal on Very Large Data Bases,* 5(4), 229-237. doi: 10.1007/s007780050026

Burleson, D. K. (n.d.). Oracle bitmap indexes. Retrieved from http://www.dba-oracle.com/oracle_tips_bitmapped_indexes.htm

Burleson, D. K. (2006). e-Bay's massive Oracle database. Retrieved from http://www.dba-oracle.com/oracle_news/news_ebay_massive_oracle.htm

Burleson, D. K. (2007). Oracle MREADTIM and SREADTIM workload statistics. Retrieved from http://www.dba-oracle.com/t_mreadtim_sreadtim_workload_statistics.htm

Burleson, D. K. (2008). Oracle: Flush the data buffer cache. Retrieved from http://www.dba-oracle.com/t_flush_buffer_cache.htm

Burleson, D. K. (2009). Oracle SQL plan regression. Retrieved http://www.dba-oracle.com/t_sql_plan_regression.htm

Burleson, D. K. (2010). Turbocharge SQL with advanced Oracle indexing. Retrieved from http://www.dba-oracle.com/art_9i_indexing.htm

Burleson, D. K. (2012). Oracle partitioning. Retrieved http://www.dba-oracle.com/tips_oracle_partitioning_hash_range.htm

Burleson, D. K., & Danchenkov, A. (2005). *Oracle tuning, the definitive reference*. Kittrell: Rampant.

Carter, S. (2007). *The new language of business: SOA & web 2.0.* Upper Saddle River, NJ: IBM Press.

Chamberlin, D. & Boyce, R. (1974). SEQUEL: A structured English query language. *Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) workshop on Data description, access and control FIDET '74. 249 – 264. Ann Arbor.* doi: 10.1145/800296.811515.

Colgan, M. (2005). Query optimization in Oracle database 10g release 2 - an Oracle white paper June 2005. Redwood Shores, CA: Oracle.

Colgan, M. (2008). Upgrading from Oracle database 9i to 10g: What to expect from the optimizer - an Oracle white paper July 2008. Redwood Shores, CA : Oracle.

Colgan, M. (2010). Upgrading from Oracle database 10g to 11g: What to expect from the optimizer - an Oracle white paper November 2010. Redwood Shores, CA: Oracle.

Colgan, M. (2011a). The Oracle optimizer explain the explain plan - an Oracle white paper May 2011. Redwood Shores, CA: Oracle.

Colgan, M. (2011b). Understanding optimizer statistics - an Oracle white paper November 2011. Redwood Shores, CA: Oracle.

Colgan, M. (Nov, 2011c). Lies, damned lies, and statistics. Retrieved from https://blogs.oracle.com/optimizer/entry/lies_damned_lies_and_statistics

Date, C. (2003). Edgar F. Codd. A tribute and personal memoir. *ACM SIGMOD Record.* 32[4], 4-13. doi: 10.1145/959060.959061

Davies, J., Shaffer, D., & L'her, D. (2009). Oracle SOA Suite 11g -- an Oracle White Paper. Redwood Shores, CA: Oracle.

Fan, S., Jiang, L., Lumpkin, G., & Sancheti, S. (2003). DSS performance in Oracle database 10g - an Oracle White Paper September 2003. Redwood Shores, CA: Oracle.

Fechner, D. (2006, Jun 15). Distribution statistics uses with the DB2 optimizer. Retrieved from http://www.ibm.com/developerworks/data/library/techarticle/dm-0606fechner/

Galindo-Legaria, C., Joshi, M. M., Waas, F., & Wu, M. (2003). Statistics on views. *Paper presented at Proceedings of the 29th international conference on very large data bases* - 29. Berlin.

Hall, T. (2012). Automatic SQL tuning in Oracle database 10g. Retrieved from http://www.oracle-base.com/articles/10g/automatic-sql-tuning-10g.php

Herodotou, H., Borisov, N., & Babu, S. (2011). Query optimization techniques for partitioned tables. *Paper presented at Proceedings of the 2011 international conference on management of data. Athens.* doi: 10.1145/1989323.1989330

Hu, Y., Sundara, S., & Srinivasan, J. (2007). Supporting time constraint SQL queries in Oracle. *Paper presented at Proceedings of the 33rd international conference on very large data bases.* 1207 – 1218. Vienna.

Hu, Y., Sundara, S., & Srinivasan, J. (2009). Estimating aggregates in time-constrained approximate queries in Oracle. *Paper presented at Proceedings of the 12th international Conference on Extending Database Technology: Advances in Database Technology.* 1104-1107. Saint-Petersburg. doi: 10.1145/1516360.1516487

Krompass, S., Kuno, H., Wiener, J., Wilkinson, K., Dayal, U., & Kemper, A. (2009). Managing long-running queries. *Paper presented at the 12th International Conference on Extending*

Database Technology: Advances in Database Technology. 132-143. Saint-Petersberg. doi: 10.1145/1516360.1516377

Kuntulu, K. (2010). Invisible indexes in Oracle 11g. Retrieved from http://viralpatel.net/blogs/invisible-indexes-in-oracle-11g/

Kyte, T. (2000, Jun 15). Consider cost or time -- trying to compare the cost of two queries. Retrieved from http://asktom.oracle.com/pls/asktom/f?p=100:11:0::::P11_QUESTION_ID:313416745628

Kyte, T. (2005). Ask Tom. Retrieved from http://asktom.oracle.com/pls/asktom/f?p=100:11:0::::P11_QUESTION_ID:40230704959128

Kyte, T. (2009, Feb). On dynamic sampling. Retrieved from http://www.oracle.com/technetwork/issue-archive/2009/09-jan/o19asktom-086775.html

Kyte, T. (2013, Jul). Hints when to use. Retrieved from http://asktom.oracle.com/pls/asktom/f?p=100:11:0::::P11_QUESTION_ID:7038986332061

Lee, A. W., & Zait, M. (2008). Closing the query processing loop in Oracle 11g. *Proceedings of the VLDB Endowment, 1(2), 1368-1378. Auckland.* doi: 10.1145/1454159.1454178

Leishman, R. (2007, Feb). Understanding indexes. Retrieved from http://www.orafaq.com/node/1403

Lejeune, H., Buch, V., & Palmer, C. (2003). Database performance with Oracle database 10g release 2 - an Oracle white paper May 2005. Redwood Shores, CA: Oracle.

Lewis, J. (2006a). *Cost-based oracle fundamentals*. Berkeley: Apress.

Lewis, J. (2006b). Recent enhancement in optimization and some not so recent. Retrieved from http://www.oracle.com/technetwork/database/focus-areas/manageability/optimizer-enhancements-jlewis-131438.pdf

Lewis, J. (2006c). Cost is time. Retrieved from http://jonathanlewis.wordpress.com/2006/12/11/cost-is-time/

Lewis, J. (2011, Jan 10). Cost -- again. Retrieved from http://jonathanlewis.wordpress.com/2011/01/10/cost-again

Luo, G., Naughton, J. F., Ellmann, C. J., & Watzke, M. W. (2004). Toward a progress indicator for database queries. *In Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data. Paris*: ACM. 791-802. doi: 10.1145/1007568.1007658

Microsoft (2009, Jan). Database engine tuning advisor overview. Retrieved from http:// msdn.microsoft.com/en-us/library/ms173494.aspx

Niemiec, R. (1999). *Oracle performance tuning tips & techniques*. Berkeley: Osborne/McGraw-Hill.

Oracle (2005, Jun). Oracle database performance tuning guide 10g release 2 (10.2) B14211-01. Retrieved from http://web.njit.edu/info/limpid/DOC/server.102/b14211.pdf

Oracle (2007, Jul). VLDB and partitioning guide 11g Release 1. Retrieved from http://docs.oracle.com/cd/B28359_01/server.111/b32024.pdf

Oracle (2011, Sep). Oracle database concepts 11g release 2 (11.2) E25789-01. Retrieved from http://docs.oracle.com/cd/E11882_01/server.112/e25789.pdf

OracleFaqs (2008, Sep). Predicate. Retrieved from http://www.orafaq.com/wiki/Predicate

OracleFaqs (2012, Jan). Toad product review. Retrieved from http://www.orafaq.com/wiki/TOAD_Product_Review

Portugal, P. F. (2009). The history of Oracle. Retrieved from http://www.dba-oracle.com/t_history_oracle.htm

Ratzesberger, O. (2010, Oct 06). Interview by Monash. eBay followup -- Greenplum out, Teradata > 10 petabytes, Hadoop has some value, and more. Retrieved from http://www.dbms2.com/2010/10/06/ebay-followup-greenplum-out-teradata-10-petabytes-hadoop-has-some-value-and-more/

Rob, P., & Coronel, C. (2007). *Database systems, design, implementation, and management.* Boston: Thomson Learning.

Tendulkar, D. M., & Phalak, C. (2011). Proactive performance testing using SQL performance assurance services (sql-pass). *Paper presented at Proceedings of the International Conference & Workshop on Emerging Trends in Technology.* 541-547. Mumbai. doi: 10.1145/1980022.1980138

Thomas, J. (2011, Jul 25). Database revenues on the rise, so sayeth the Tracker. Retrieved from http://www.itjungle.com/tfh/tfh072511-story09.html

Trenholm, R. (2009, Nov 02). NSA to store yottabytes in Utah data center. Retrieved from http://crave.cnet.co.uk/gadgets/nsa-to-store-yottabytes-in-utah-data-centre-49304118/

Waas, F. M., Giakoumakis, L., & Zhang, S. (2011). Plan space analysis: An early warning system to detect plan regressions in cost-based optimizers. *Paper presented at Proceedings of the Fourth International Workshop on Testing. Article 2. Athens*. doi: 10.1145/1988842.1988844

Yagoub, K., & Gongloor, P. (2007). SQL performance analyzer – an Oracle white paper November 2007. Redwood Shores, CA: Oracle.

Yang, H., & Faris, N. (2009, Sep 14). Ackamai reveals 2 seconds as the new threshold of acceptability for ecommerce web page response times. Retrieved from http://www.akamai.com/html/about/press/releases/2009/press_091409.html

Ziauddin, M., Das, D., Su, H., Zhu, Y., & Yagoub, K. (2008). Optimizer plan change management: Improved stability and performance in Oracle 11g. *Proceedings of the VLDB Endowment, 1(2), 1346-1355. Auckland.* doi: 10.1145/1454159.1454175

Appendix 1

```
CREATE TABLE TEST_USER.MASTER_1
(
  PK_VALUE       NUMBER,
  MASTER_2_LINK  NUMBER,
  MASTER_3_LINK  NUMBER,
  DATE_1         DATE,
  DATE_2         DATE,
  DATE_3         DATE,
  DATE_4         DATE,
  DATE_5         DATE,
  CHAR_1         CHAR(1 BYTE),
  CHAR_2         CHAR(1 BYTE),
  CHAR_3         CHAR(1 BYTE),
  VALUE_1        VARCHAR2(10 BYTE),
  VALUE_2        VARCHAR2(10 BYTE)
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;

CREATE TABLE TEST_USER.MASTER_1_L2_TAB1
(
  PK_VALUE     NUMBER,
  PARENT_LINK  NUMBER,
  DATE_1       DATE,
  DATE_2       DATE,
  DATE_3       DATE,
  CHAR_1       CHAR(1 BYTE),
  CHAR_2       CHAR(1 BYTE),
  CHAR_3       CHAR(1 BYTE),
```

```
  VALUE_1       VARCHAR2(10 BYTE),
  VALUE_2       VARCHAR2(10 BYTE),
  VALUE_3       VARCHAR2(10 BYTE)
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_1_L2_TAB2
(
  PK_VALUE      NUMBER,
  PARENT_LINK   NUMBER,
  DATE_1        DATE,
  DATE_2        DATE,
  DATE_3        DATE,
  CHAR_1        CHAR(1 BYTE),
  CHAR_2        CHAR(1 BYTE),
  CHAR_3        CHAR(1 BYTE),
  VALUE_1       VARCHAR2(10 BYTE),
  VALUE_2       VARCHAR2(10 BYTE),
  VALUE_3       VARCHAR2(10 BYTE)
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
```

```
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_1_L2_TAB3
(
  PK_VALUE     NUMBER,
  PARENT_LINK  NUMBER,
  DATE_1       DATE,
  DATE_2       DATE,
  DATE_3       DATE,
  CHAR_1       CHAR(1 BYTE),
  CHAR_2       CHAR(1 BYTE),
  CHAR_3       CHAR(1 BYTE),
  VALUE_1      VARCHAR2(10 BYTE),
  VALUE_2      VARCHAR2(10 BYTE),
  VALUE_3      VARCHAR2(10 BYTE)
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_1_L3_TAB1
(
  PK_VALUE     NUMBER,
```

```
  PARENT_LINK  NUMBER,
  DATE_1       DATE,
  DATE_2       DATE,
  DATE_3       DATE,
  DATE_4       DATE,
  DATE_5       DATE,
  VALUE_1      VARCHAR2(10 BYTE),
  VALUE_2      VARCHAR2(10 BYTE),
  VALUE_3      VARCHAR2(10 BYTE),
  CHAR_1       CHAR(1 BYTE),
  CHAR_2       CHAR(1 BYTE)
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_1_L3_TAB2
(
  PK_VALUE     NUMBER,
  PARENT_LINK  NUMBER,
  DATE_1       DATE,
  DATE_2       DATE,
  DATE_3       DATE,
  DATE_4       DATE,
  DATE_5       DATE,
  VALUE_1      VARCHAR2(10 BYTE),
  VALUE_2      VARCHAR2(10 BYTE),
  VALUE_3      VARCHAR2(10 BYTE),
  CHAR_1       CHAR(1 BYTE),
  CHAR_2       CHAR(1 BYTE)
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
```

```
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_1_L3_TAB3
(
  PK_VALUE     NUMBER,
  PARENT_LINK  NUMBER,
  DATE_1       DATE,
  DATE_2       DATE,
  DATE_3       DATE,
  DATE_4       DATE,
  DATE_5       DATE,
  VALUE_1      VARCHAR2(10 BYTE),
  VALUE_2      VARCHAR2(10 BYTE),
  VALUE_3      VARCHAR2(10 BYTE),
  CHAR_1       CHAR(1 BYTE),
  CHAR_2       CHAR(1 BYTE)
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
```

```
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_2
(
  PK_VALUE       NUMBER,
  MASTER_1_LINK  NUMBER,
  MASTER_3_LINK  NUMBER,
  VALUE_1        VARCHAR2(10 BYTE),
  VALUE_2        VARCHAR2(10 BYTE),
  VALUE_3        VARCHAR2(10 BYTE),
  VALUE_4        VARCHAR2(10 BYTE),
  VALUE_5        VARCHAR2(10 BYTE),
  CHAR_1         CHAR(1 BYTE),
  CHAR_2         CHAR(1 BYTE),
  CHAR_3         CHAR(1 BYTE),
  DATE_1         DATE,
  DATE_2         DATE
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_2_L2_TAB1
(
  PK_VALUE     NUMBER,
  DATE_1       DATE,
  DATE_2       DATE,
  DATE_3       DATE,
```

```
  DATE_4       DATE,
  DATE_5       DATE,
  DATE_6       DATE,
  DATE_7       DATE,
  DATE_8       DATE,
  DATE_9       DATE,
  DATE10       DATE,
  PARENT_LINK  NUMBER
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_2_L2_TAB2
(
  PK_VALUE     NUMBER,
  NUM_1        NUMBER,
  NUM_2        NUMBER,
  NUM_3        NUMBER,
  NUM_4        NUMBER,
  NUM_5        NUMBER,
  NUM_6        NUMBER,
  NUM_7        NUMBER,
  NUM_8        NUMBER,
  NUM_9        NUMBER,
  NUM_10       NUMBER,
  PARENT_LINK  NUMBER
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
```

```
MAXTRANS    255
STORAGE     (
             INITIAL          64K
             NEXT             1M
             MINEXTENTS       1
             MAXEXTENTS       UNLIMITED
             PCTINCREASE      0
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_2_L2_TAB3
(
  PK_VALUE     NUMBER,
  VALUE_1      VARCHAR2(10 BYTE),
  VALUE_2      VARCHAR2(10 BYTE),
  VALUE_3      VARCHAR2(10 BYTE),
  VALUE_4      VARCHAR2(10 BYTE),
  VALUE_5      VARCHAR2(10 BYTE),
  VALUE_6      VARCHAR2(10 BYTE),
  VALUE_7      VARCHAR2(10 BYTE),
  VALUE_8      VARCHAR2(10 BYTE),
  VALUE_9      VARCHAR2(10 BYTE),
  VALUE_10     VARCHAR2(10 BYTE),
  PARENT_LINK  NUMBER
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED     0
PCTFREE     10
INITRANS    1
MAXTRANS    255
STORAGE     (
             INITIAL          64K
             NEXT             1M
             MINEXTENTS       1
             MAXEXTENTS       UNLIMITED
             PCTINCREASE      0
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
LOGGING
NOCOMPRESS
NOCACHE
```

```
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_2_L3_TAB1
(
  PK_VALUE     NUMBER,
  PARENT_LINK  NUMBER,
  VALUE_1      VARCHAR2(10 BYTE),
  VALUE_2      VARCHAR2(10 BYTE),
  VALUE_3      VARCHAR2(10 BYTE),
  VALUE_4      VARCHAR2(10 BYTE),
  DATE_1       DATE,
  DATE_2       DATE,
  DATE_3       DATE,
  CHAR_1       CHAR(1 BYTE),
  CHAR_2       CHAR(1 BYTE),
  CHAR_3       CHAR(1 BYTE)
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_2_L3_TAB2
(
  PK_VALUE     NUMBER,
  PARENT_LINK  NUMBER,
  VALUE_1      VARCHAR2(10 BYTE),
  VALUE_2      VARCHAR2(10 BYTE),
  VALUE_3      VARCHAR2(10 BYTE),
  VALUE_4      VARCHAR2(10 BYTE),
  DATE_1       DATE,
  DATE_2       DATE,
```

```
  DATE_3        DATE,
  CHAR_1        CHAR(1 BYTE),
  CHAR_2        CHAR(1 BYTE),
  CHAR_3        CHAR(1 BYTE)
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_2_L3_TAB3
(
  PK_VALUE      NUMBER,
  PARENT_LINK   NUMBER,
  VALUE_1       VARCHAR2(10 BYTE),
  VALUE_2       VARCHAR2(10 BYTE),
  VALUE_3       VARCHAR2(10 BYTE),
  VALUE_4       VARCHAR2(10 BYTE),
  DATE_1        DATE,
  DATE_2        DATE,
  DATE_3        DATE,
  CHAR_1        CHAR(1 BYTE),
  CHAR_2        CHAR(1 BYTE),
  CHAR_3        CHAR(1 BYTE)
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
```

```
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_3
(
  PK_VALUE       NUMBER,
  NUM_1          NUMBER,
  NUM_2          NUMBER,
  NUM_3          NUMBER,
  NUM_4          NUMBER,
  NUM_5          NUMBER,
  CHAR_1         CHAR(1 BYTE),
  CHAR_2         CHAR(1 BYTE),
  CHAR_3         CHAR(1 BYTE),
  VALUE_1        VARCHAR2(10 BYTE),
  VALUE_2        VARCHAR2(10 BYTE),
  MASTER_1_LINK  NUMBER,
  MASTER_2_LINK  NUMBER
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;
```

```
CREATE TABLE TEST_USER.MASTER_3_L2_TAB1
(
  PK_VALUE      NUMBER,
  COMP_1        CHAR(1 BYTE),
  COMP_2        CHAR(1 BYTE),
  COMP_3        CHAR(1 BYTE),
  DATE_COMP_1   DATE,
  DATE_COMP_2   DATE,
  NUM_COMP_1    NUMBER,
  NUM_COMP_2    NUMBER,
  NUM_COMP_3    NUMBER,
  PARENT_LINK   NUMBER
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_3_L2_TAB2
(
  PK_VALUE      NUMBER,
  COMP_1        CHAR(1 BYTE),
  COMP_2        CHAR(1 BYTE),
  COMP_3        CHAR(1 BYTE),
  DATE_COMP_1   DATE,
  DATE_COMP_2   DATE,
  NUM_COMP_1    NUMBER,
  NUM_COMP_2    NUMBER,
  NUM_COMP_3    NUMBER,
  PARENT_LINK   NUMBER
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
```

```
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_3_L2_TAB3
(
  PK_VALUE      NUMBER,
  COMP_1        CHAR(1 BYTE),
  COMP_2        CHAR(1 BYTE),
  COMP_3        CHAR(1 BYTE),
  DATE_COMP_1   DATE,
  DATE_COMP_2   DATE,
  NUM_COMP_1    NUMBER,
  NUM_COMP_2    NUMBER,
  NUM_COMP_3    NUMBER,
  PARENT_LINK   NUMBER
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
```

```
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_3_L3_TAB1
(
  PK_VALUE     NUMBER,
  PARENT_LINK  NUMBER,
  VALUE_1      VARCHAR2(10 BYTE),
  VALUE_2      VARCHAR2(10 BYTE),
  VALUE_3      VARCHAR2(10 BYTE),
  CHAR_1       CHAR(1 BYTE),
  CHAR_2       CHAR(1 BYTE),
  CHAR_3       CHAR(1 BYTE),
  DATE_1       DATE,
  DATE_2       DATE,
  DATE_3       DATE,
  DATE_4       DATE
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_3_L3_TAB2
(
  PK_VALUE     NUMBER,
  PARENT_LINK  NUMBER,
  VALUE_1      VARCHAR2(10 BYTE),
  VALUE_2      VARCHAR2(10 BYTE),
  VALUE_3      VARCHAR2(10 BYTE),
  CHAR_1       CHAR(1 BYTE),
  CHAR_2       CHAR(1 BYTE),
```

```
  CHAR_3        CHAR(1 BYTE),
  DATE_1        DATE,
  DATE_2        DATE,
  DATE_3        DATE,
  DATE_4        DATE
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE TABLE TEST_USER.MASTER_3_L3_TAB3
(
  PK_VALUE      NUMBER,
  PARENT_LINK   NUMBER,
  VALUE_1       VARCHAR2(10 BYTE),
  VALUE_2       VARCHAR2(10 BYTE),
  VALUE_3       VARCHAR2(10 BYTE),
  CHAR_1        CHAR(1 BYTE),
  CHAR_2        CHAR(1 BYTE),
  CHAR_3        CHAR(1 BYTE),
  DATE_1        DATE,
  DATE_2        DATE,
  DATE_3        DATE,
  DATE_4        DATE
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    10
INITRANS   1
MAXTRANS   255
STORAGE    (
            INITIAL          64K
```

```
            NEXT              1M
            MINEXTENTS        1
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
LOGGING
NOCOMPRESS
NOCACHE
NOPARALLEL
MONITORING;


CREATE BITMAP INDEX TEST_USER.M1_L2_T1_CHAR_1_IDX ON
TEST_USER.MASTER_1_L2_TAB1
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL           64K
            NEXT              1M
            MINEXTENTS        1
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L2_T1_CHAR_2_IDX ON
TEST_USER.MASTER_1_L2_TAB1
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL           64K
            NEXT              1M
            MINEXTENTS        1
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
```

```
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L2_T1_CHAR_3_IDX ON
TEST_USER.MASTER_1_L2_TAB1
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T1_DATE_1_IDX ON
TEST_USER.MASTER_1_L2_TAB1
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.M1_L2_T1_DATE_2_IDX ON
TEST_USER.MASTER_1_L2_TAB1
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T1_DATE_3_IDX ON
TEST_USER.MASTER_1_L2_TAB1
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M1_L2_T1_PK_IDX ON
TEST_USER.MASTER_1_L2_TAB1
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
```

```
STORAGE     (
             INITIAL          64K
             NEXT             1M
             MINEXTENTS       1
             MAXEXTENTS       UNLIMITED
             PCTINCREASE      0
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T1_VALUE_1_IDX ON
TEST_USER.MASTER_1_L2_TAB1
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE     (
             INITIAL          64K
             NEXT             1M
             MINEXTENTS       1
             MAXEXTENTS       UNLIMITED
             PCTINCREASE      0
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T1_VALUE_2_IDX ON
TEST_USER.MASTER_1_L2_TAB1
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE     (
             INITIAL          64K
             NEXT             1M
             MINEXTENTS       1
             MAXEXTENTS       UNLIMITED
             PCTINCREASE      0
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
```

```
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T1_VALUE_3_IDX ON
TEST_USER.MASTER_1_L2_TAB1
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L2_T2_CHAR_1_IDX ON
TEST_USER.MASTER_1_L2_TAB2
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L2_T2_CHAR_2_IDX ON
TEST_USER.MASTER_1_L2_TAB2
```

```
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L2_T2_CHAR_3_IDX ON
TEST_USER.MASTER_1_L2_TAB2
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_DATE_1_IDX ON
TEST_USER.MASTER_1_L2_TAB2
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
```

```
            NEXT              1M
            MINEXTENTS        1
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_DATE_2_IDX ON
TEST_USER.MASTER_1_L2_TAB2
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL           64K
            NEXT              1M
            MINEXTENTS        1
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_DATE_3_IDX ON
TEST_USER.MASTER_1_L2_TAB2
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL           64K
            NEXT              1M
            MINEXTENTS        1
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
```

```
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_NUM_10_IDX ON
TEST_USER.MASTER_2_L2_TAB2
(NUM_10)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_NUM_1_IDX ON
TEST_USER.MASTER_2_L2_TAB2
(NUM_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_NUM_2_IDX ON
TEST_USER.MASTER_2_L2_TAB2
(NUM_2)
LOGGING
```

```
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_NUM_3_IDX ON
TEST_USER.MASTER_2_L2_TAB2
(NUM_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_NUM_4_IDX ON
TEST_USER.MASTER_2_L2_TAB2
(NUM_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
```

```
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_NUM_5_IDX ON
TEST_USER.MASTER_2_L2_TAB2
(NUM_5)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_NUM_6_IDX ON
TEST_USER.MASTER_2_L2_TAB2
(NUM_6)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.M1_L2_T2_NUM_7_IDX ON
TEST_USER.MASTER_2_L2_TAB2
(NUM_7)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_NUM_8_IDX ON
TEST_USER.MASTER_2_L2_TAB2
(NUM_8)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_NUM_9_IDX ON
TEST_USER.MASTER_2_L2_TAB2
(NUM_9)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
```

```
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M1_L2_T2_PK_IDX ON
TEST_USER.MASTER_1_L2_TAB2
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_VALUE_1_IDX ON
TEST_USER.MASTER_1_L2_TAB2
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
```

```
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_VALUE_2_IDX ON
TEST_USER.MASTER_1_L2_TAB2
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T2_VALUE_3_IDX ON
TEST_USER.MASTER_1_L2_TAB2
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE BITMAP INDEX TEST_USER.M1_L2_T3_CHAR_1_IDX ON
TEST_USER.MASTER_1_L2_TAB3
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L2_T3_CHAR_2_IDX ON
TEST_USER.MASTER_1_L2_TAB3
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L2_T3_CHAR_3_IDX ON
TEST_USER.MASTER_1_L2_TAB3
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
```

```
STORAGE    (
             INITIAL          64K
             NEXT             1M
             MINEXTENTS       1
             MAXEXTENTS       UNLIMITED
             PCTINCREASE      0
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T3_DATE_1_IDX ON
TEST_USER.MASTER_1_L2_TAB3
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
             INITIAL          64K
             NEXT             1M
             MINEXTENTS       1
             MAXEXTENTS       UNLIMITED
             PCTINCREASE      0
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T3_DATE_2_IDX ON
TEST_USER.MASTER_1_L2_TAB3
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
             INITIAL          64K
             NEXT             1M
             MINEXTENTS       1
             MAXEXTENTS       UNLIMITED
             PCTINCREASE      0
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
```

```
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T3_DATE_3_IDX ON
TEST_USER.MASTER_1_L2_TAB3
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M1_L2_T3_PK_IDX ON
TEST_USER.MASTER_1_L2_TAB3
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T3_VALUE_1_IDX ON
TEST_USER.MASTER_1_L2_TAB3
```

```
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T3_VALUE_2_IDX ON
TEST_USER.MASTER_1_L2_TAB3
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L2_T3_VALUE_3_IDX ON
TEST_USER.MASTER_1_L2_TAB3
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
```

```
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L3_T1_CHAR_1_IDX ON
TEST_USER.MASTER_1_L3_TAB1
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L3_T1_CHAR_2_IDX ON
TEST_USER.MASTER_1_L3_TAB1
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
```

```
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T1_DATE_1_IDX ON
TEST_USER.MASTER_1_L3_TAB1
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T1_DATE_2_IDX ON
TEST_USER.MASTER_1_L3_TAB1
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T1_DATE_3_IDX ON
TEST_USER.MASTER_1_L3_TAB1
(DATE_3)
LOGGING
```

```
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T1_DATE_4_IDX ON
TEST_USER.MASTER_1_L3_TAB1
(DATE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T1_DATE_5_IDX ON
TEST_USER.MASTER_1_L3_TAB1
(DATE_5)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
```

```
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M1_L3_T1_PK_IDX ON
TEST_USER.MASTER_1_L3_TAB1
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T1_VALUE_1_IDX ON
TEST_USER.MASTER_1_L3_TAB1
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.M1_L3_T1_VALUE_2_IDX ON
TEST_USER.MASTER_1_L3_TAB1
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T1_VALUE_3_IDX ON
TEST_USER.MASTER_1_L3_TAB1
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L3_T2_CHAR_1_IDX ON
TEST_USER.MASTER_1_L3_TAB2
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
```

```
INITRANS    2
MAXTRANS    255
STORAGE     (
             INITIAL          64K
             NEXT             1M
             MINEXTENTS       1
             MAXEXTENTS       UNLIMITED
             PCTINCREASE      0
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L3_T2_CHAR_2_IDX ON
TEST_USER.MASTER_1_L3_TAB2
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE     10
INITRANS    2
MAXTRANS    255
STORAGE     (
             INITIAL          64K
             NEXT             1M
             MINEXTENTS       1
             MAXEXTENTS       UNLIMITED
             PCTINCREASE      0
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T2_DATE_1_IDX ON
TEST_USER.MASTER_1_L3_TAB2
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE     10
INITRANS    2
MAXTRANS    255
STORAGE     (
             INITIAL          64K
             NEXT             1M
             MINEXTENTS       1
             MAXEXTENTS       UNLIMITED
             PCTINCREASE      0
```

```
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T2_DATE_2_IDX ON
TEST_USER.MASTER_1_L3_TAB2
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T2_DATE_3_IDX ON
TEST_USER.MASTER_1_L3_TAB2
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.M1_L3_T2_DATE_4_IDX ON
TEST_USER.MASTER_1_L3_TAB2
(DATE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T2_DATE_5_IDX ON
TEST_USER.MASTER_1_L3_TAB2
(DATE_5)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M1_L3_T2_PK_IDX ON
TEST_USER.MASTER_1_L3_TAB2
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
```

```
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T2_VALUE_1_IDX ON
TEST_USER.MASTER_1_L3_TAB2
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T2_VALUE_2_IDX ON
TEST_USER.MASTER_1_L3_TAB2
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
```

```
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T2_VALUE_3_IDX ON
TEST_USER.MASTER_1_L3_TAB2
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L3_T3_CHAR_1_IDX ON
TEST_USER.MASTER_1_L3_TAB3
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M1_L3_T3_CHAR_2_IDX ON
TEST_USER.MASTER_1_L3_TAB3
```

```
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T3_DATE_1_IDX ON
TEST_USER.MASTER_1_L3_TAB3
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T3_DATE_2_IDX ON
TEST_USER.MASTER_1_L3_TAB3
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
```

```
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T3_DATE_3_IDX ON
TEST_USER.MASTER_1_L3_TAB3
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T3_DATE_4_IDX ON
TEST_USER.MASTER_1_L3_TAB3
(DATE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
```

```
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T3_DATE_5_IDX ON
TEST_USER.MASTER_1_L3_TAB3
(DATE_5)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M1_L3_T3_PK_IDX ON
TEST_USER.MASTER_1_L3_TAB3
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T3_VALUE_1_IDX ON
TEST_USER.MASTER_1_L3_TAB3
(VALUE_1)
LOGGING
```

```
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T3_VALUE_2_IDX ON
TEST_USER.MASTER_1_L3_TAB3
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M1_L3_T3_VALUE_3_IDX ON
TEST_USER.MASTER_1_L3_TAB3
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
```

```
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T1_DATE_10_IDX ON
TEST_USER.MASTER_2_L2_TAB1
(DATE10)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T1_DATE_1_IDX ON
TEST_USER.MASTER_2_L2_TAB1
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.M2_L2_T1_DATE_2_IDX ON
TEST_USER.MASTER_2_L2_TAB1
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T1_DATE_3_IDX ON
TEST_USER.MASTER_2_L2_TAB1
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T1_DATE_4_IDX ON
TEST_USER.MASTER_2_L2_TAB1
(DATE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
```

```
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T1_DATE_5_IDX ON
TEST_USER.MASTER_2_L2_TAB1
(DATE_5)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T1_DATE_6_IDX ON
TEST_USER.MASTER_2_L2_TAB1
(DATE_6)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
```

```
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T1_DATE_7_IDX ON
TEST_USER.MASTER_2_L2_TAB1
(DATE_7)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T1_DATE_8_IDX ON
TEST_USER.MASTER_2_L2_TAB1
(DATE_8)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.M2_L2_T1_DATE_9_IDX ON
TEST_USER.MASTER_2_L2_TAB1
(DATE_9)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M2_L2_T1_PK_IDX ON
TEST_USER.MASTER_2_L2_TAB1
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M2_L2_T3_PK_IDX ON
TEST_USER.MASTER_2_L2_TAB3
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
```

```
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T3_VALUE_10_IDX ON
TEST_USER.MASTER_2_L2_TAB3
(VALUE_10)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T3_VALUE_1_IDX ON
TEST_USER.MASTER_2_L2_TAB3
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
```

```
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T3_VALUE_2_IDX ON
TEST_USER.MASTER_2_L2_TAB3
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T3_VALUE_3_IDX ON
TEST_USER.MASTER_2_L2_TAB3
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T3_VALUE_4_IDX ON
TEST_USER.MASTER_2_L2_TAB3
```

```
(VALUE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T3_VALUE_5_IDX ON
TEST_USER.MASTER_2_L2_TAB3
(VALUE_5)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T3_VALUE_6_IDX ON
TEST_USER.MASTER_2_L2_TAB3
(VALUE_6)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
```

```
            NEXT              1M
            MINEXTENTS        1
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE  DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T3_VALUE_7_IDX ON
TEST_USER.MASTER_2_L2_TAB3
(VALUE_7)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL           64K
            NEXT              1M
            MINEXTENTS        1
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE  DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T3_VALUE_8_IDX ON
TEST_USER.MASTER_2_L2_TAB3
(VALUE_8)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL           64K
            NEXT              1M
            MINEXTENTS        1
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE  DEFAULT
           )
```

```
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L2_T3_VALUE_9_IDX ON
TEST_USER.MASTER_2_L2_TAB3
(VALUE_9)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M2_L2_TAB2_PK_IDX ON
TEST_USER.MASTER_2_L2_TAB2
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M2_L3_T1_CHAR_1_IDX ON
TEST_USER.MASTER_2_L3_TAB1
(CHAR_1)
LOGGING
```

```
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M2_L3_T1_CHAR_2_IDX ON
TEST_USER.MASTER_2_L3_TAB1
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M2_L3_T1_CHAR_3_IDX ON
TEST_USER.MASTER_2_L3_TAB1
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
```

```
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T1_DATE_1_IDX ON
TEST_USER.MASTER_2_L3_TAB1
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T1_DATE_2_IDX ON
TEST_USER.MASTER_2_L3_TAB1
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.M2_L3_T1_DATE_3_IDX ON
TEST_USER.MASTER_2_L3_TAB1
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M2_L3_T1_PK_IDX ON
TEST_USER.MASTER_2_L3_TAB1
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T1_VALUE_1_IDX ON
TEST_USER.MASTER_2_L3_TAB1
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
```

```
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T1_VALUE_2_IDX ON
TEST_USER.MASTER_2_L3_TAB1
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T1_VALUE_3_IDX ON
TEST_USER.MASTER_2_L3_TAB1
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
```

```
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T1_VALUE_4_IDX ON
TEST_USER.MASTER_2_L3_TAB1
(VALUE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
              INITIAL          64K
              NEXT             1M
              MINEXTENTS       1
              MAXEXTENTS       UNLIMITED
              PCTINCREASE      0
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M2_L3_T2_CHAR_1_IDX ON
TEST_USER.MASTER_2_L3_TAB2
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
              INITIAL          64K
              NEXT             1M
              MINEXTENTS       1
              MAXEXTENTS       UNLIMITED
              PCTINCREASE      0
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
NOPARALLEL
INVISIBLE;
```

```
CREATE BITMAP INDEX TEST_USER.M2_L3_T2_CHAR_2_IDX ON
TEST_USER.MASTER_2_L3_TAB2
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M2_L3_T2_CHAR_3_IDX ON
TEST_USER.MASTER_2_L3_TAB2
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T2_DATE_1_IDX ON
TEST_USER.MASTER_2_L3_TAB2
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
```

```
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T2_DATE_2_IDX ON
TEST_USER.MASTER_2_L3_TAB2
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T2_DATE_3_IDX ON
TEST_USER.MASTER_2_L3_TAB2
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
```

```
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M2_L3_T2_PK_IDX ON
TEST_USER.MASTER_2_L3_TAB2
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T2_VALUE_1_IDX ON
TEST_USER.MASTER_2_L3_TAB2
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T2_VALUE_2_IDX ON
TEST_USER.MASTER_2_L3_TAB2
```

```
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T2_VALUE_3_IDX ON
TEST_USER.MASTER_2_L3_TAB2
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T2_VALUE_4_IDX ON
TEST_USER.MASTER_2_L3_TAB2
(VALUE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
```

```
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M2_L3_T3_CHAR_1_IDX ON
TEST_USER.MASTER_2_L3_TAB3
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M2_L3_T3_CHAR_2_IDX ON
TEST_USER.MASTER_2_L3_TAB3
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
```

```
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M2_L3_T3_CHAR_3_IDX ON
TEST_USER.MASTER_2_L3_TAB3
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T3_DATE_1_IDX ON
TEST_USER.MASTER_2_L3_TAB3
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T3_DATE_2_IDX ON
TEST_USER.MASTER_2_L3_TAB3
(DATE_2)
LOGGING
```

```
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T3_DATE_3_IDX ON
TEST_USER.MASTER_2_L3_TAB3
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M2_L3_T3_PK_IDX ON
TEST_USER.MASTER_2_L3_TAB3
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
```

```
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T3_VALUE_1_IDX ON
TEST_USER.MASTER_2_L3_TAB3
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL           64K
            NEXT              1M
            MINEXTENTS        1
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T3_VALUE_2_IDX ON
TEST_USER.MASTER_2_L3_TAB3
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL           64K
            NEXT              1M
            MINEXTENTS        1
            MAXEXTENTS        UNLIMITED
            PCTINCREASE       0
            BUFFER_POOL       DEFAULT
            FLASH_CACHE       DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.M2_L3_T3_VALUE_3_IDX ON
TEST_USER.MASTER_2_L3_TAB3
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M2_L3_T3_VALUE_4_IDX ON
TEST_USER.MASTER_2_L3_TAB3
(VALUE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L2_T1_COMP_1_IDX ON
TEST_USER.MASTER_3_L2_TAB1
(COMP_1, COMP_2, COMP_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
```

```
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L2_T1_DATE_COMP_1_IDX ON
TEST_USER.MASTER_3_L2_TAB1
(DATE_COMP_1, DATE_COMP_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L2_T1_NUM_COMP_1_IDX ON
TEST_USER.MASTER_3_L2_TAB1
(NUM_COMP_1, NUM_COMP_2, NUM_COMP_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
```

```
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L2_T2_COMP_1_IDX ON
TEST_USER.MASTER_3_L2_TAB2
(COMP_1, COMP_2, COMP_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L2_T2_DATE_COMP_1_IDX ON
TEST_USER.MASTER_3_L2_TAB2
(DATE_COMP_1, DATE_COMP_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.M3_L2_T2_NUM_COMP_1_IDX ON
TEST_USER.MASTER_3_L2_TAB2
(NUM_COMP_1, NUM_COMP_2, NUM_COMP_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M3_L2_T2_PK_IDX ON
TEST_USER.MASTER_3_L2_TAB2
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L2_T3_COMP_1_IDX ON
TEST_USER.MASTER_3_L2_TAB3
(COMP_1, COMP_2, COMP_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
```

```
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L2_T3_DATE_COMP_1_IDX ON
TEST_USER.MASTER_3_L2_TAB3
(DATE_COMP_1, DATE_COMP_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L2_T3_NUM_COMP_1_IDX ON
TEST_USER.MASTER_3_L2_TAB3
(NUM_COMP_1, NUM_COMP_2, NUM_COMP_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
```

```
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M3_L2_T3_PK_IDX ON
TEST_USER.MASTER_3_L2_TAB3
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M3_L2_TB_PK_IDX ON
TEST_USER.MASTER_3_L2_TAB1
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M3_L3_T1_CHAR_1_IDX ON
TEST_USER.MASTER_3_L3_TAB1
```

```
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M3_L3_T1_CHAR_2_IDX ON
TEST_USER.MASTER_3_L3_TAB1
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M3_L3_T1_CHAR_3_IDX ON
TEST_USER.MASTER_3_L3_TAB1
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
```

```
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T1_DATE_1_IDX ON
TEST_USER.MASTER_3_L3_TAB1
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T1_DATE_2_IDX ON
TEST_USER.MASTER_3_L3_TAB1
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
```

```
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T1_DATE_3_IDX ON
TEST_USER.MASTER_3_L3_TAB1
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T1_DATE_4_IDX ON
TEST_USER.MASTER_3_L3_TAB1
(DATE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M3_L3_T1_PK_IDX ON
TEST_USER.MASTER_3_L3_TAB1
(PK_VALUE)
LOGGING
```

```
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T1_VALUE_1_IDX ON
TEST_USER.MASTER_3_L3_TAB1
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T1_VALUE_2_IDX ON
TEST_USER.MASTER_3_L3_TAB1
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
```

```
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T1_VALUE_3_IDX ON
TEST_USER.MASTER_3_L3_TAB1
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M3_L3_T2_CHAR_1_IDX ON
TEST_USER.MASTER_3_L3_TAB2
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE BITMAP INDEX TEST_USER.M3_L3_T2_CHAR_2_IDX ON
TEST_USER.MASTER_3_L3_TAB2
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M3_L3_T2_CHAR_3_IDX ON
TEST_USER.MASTER_3_L3_TAB2
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T2_DATE_1_IDX ON
TEST_USER.MASTER_3_L3_TAB2
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
```

```
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T2_DATE_2_IDX ON
TEST_USER.MASTER_3_L3_TAB2
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T2_DATE_3_IDX ON
TEST_USER.MASTER_3_L3_TAB2
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
```

```
             BUFFER_POOL       DEFAULT
             FLASH_CACHE       DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T2_DATE_4_IDX ON
TEST_USER.MASTER_3_L3_TAB2
(DATE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
             INITIAL           64K
             NEXT              1M
             MINEXTENTS        1
             MAXEXTENTS        UNLIMITED
             PCTINCREASE       0
             BUFFER_POOL       DEFAULT
             FLASH_CACHE       DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M3_L3_T2_PK_IDX ON
TEST_USER.MASTER_3_L3_TAB2
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
             INITIAL           64K
             NEXT              1M
             MINEXTENTS        1
             MAXEXTENTS        UNLIMITED
             PCTINCREASE       0
             BUFFER_POOL       DEFAULT
             FLASH_CACHE       DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.M3_L3_T2_VALUE_1_IDX ON
TEST_USER.MASTER_3_L3_TAB2
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T2_VALUE_2_IDX ON
TEST_USER.MASTER_3_L3_TAB2
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T2_VALUE_3_IDX ON
TEST_USER.MASTER_3_L3_TAB2
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
```

```
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M3_L3_T3_CHAR_1_IDX ON
TEST_USER.MASTER_3_L3_TAB3
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M3_L3_T3_CHAR_2_IDX ON
TEST_USER.MASTER_3_L3_TAB3
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
```

```
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.M3_L3_T3_CHAR_3_IDX ON
TEST_USER.MASTER_3_L3_TAB3
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T3_DATE_1_IDX ON
TEST_USER.MASTER_3_L3_TAB3
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T3_DATE_2_IDX ON
TEST_USER.MASTER_3_L3_TAB3
```

```
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T3_DATE_3_IDX ON
TEST_USER.MASTER_3_L3_TAB3
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T3_DATE_4_IDX ON
TEST_USER.MASTER_3_L3_TAB3
(DATE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
```

```
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.M3_L3_T3_PK_IDX ON
TEST_USER.MASTER_3_L3_TAB3
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T3_VALUE_1_IDX ON
TEST_USER.MASTER_3_L3_TAB3
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
```

```
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T3_VALUE_2_IDX ON
TEST_USER.MASTER_3_L3_TAB3
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.M3_L3_T3_VALUE_3_IDX ON
TEST_USER.MASTER_3_L3_TAB3
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_1_CHAR_1_IDX ON
TEST_USER.MASTER_1
(CHAR_1)
LOGGING
```

```
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_1_CHAR_2_IDX ON
TEST_USER.MASTER_1
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_1_CHAR_3_IDX ON
TEST_USER.MASTER_1
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
```

```
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_1_DATE_1_IDX ON TEST_USER.MASTER_1
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_1_DATE_2_IDX ON TEST_USER.MASTER_1
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.MASTER_1_DATE_3_IDX ON TEST_USER.MASTER_1
(DATE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_1_DATE_4_IDX ON TEST_USER.MASTER_1
(DATE_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_1_DATE_5_IDX ON TEST_USER.MASTER_1
(DATE_5)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
```

```
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_1_LINKER ON TEST_USER.MASTER_1
(MASTER_2_LINK, MASTER_3_LINK)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.MASTER_1_PK_IDX ON TEST_USER.MASTER_1
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.MASTER_1_VALUE_1_IDX ON TEST_USER.MASTER_1
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_1_VALUE_2_IDX ON TEST_USER.MASTER_1
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_2_CHAR_1_IDX ON
TEST_USER.MASTER_2
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
```

```
            INITIAL         64K
            NEXT            1M
            MINEXTENTS      1
            MAXEXTENTS      UNLIMITED
            PCTINCREASE     0
            BUFFER_POOL     DEFAULT
            FLASH_CACHE     DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_2_CHAR_2_IDX ON
TEST_USER.MASTER_2
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL         64K
            NEXT            1M
            MINEXTENTS      1
            MAXEXTENTS      UNLIMITED
            PCTINCREASE     0
            BUFFER_POOL     DEFAULT
            FLASH_CACHE     DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_2_CHAR_3_IDX ON
TEST_USER.MASTER_2
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL         64K
            NEXT            1M
            MINEXTENTS      1
            MAXEXTENTS      UNLIMITED
            PCTINCREASE     0
            BUFFER_POOL     DEFAULT
            FLASH_CACHE     DEFAULT
            CELL_FLASH_CACHE DEFAULT
```

```
             )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_2_DATE_1_IDX ON TEST_USER.MASTER_2
(DATE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_2_DATE_2_IDX ON TEST_USER.MASTER_2
(DATE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_2_LINKER ON TEST_USER.MASTER_2
(MASTER_1_LINK, MASTER_3_LINK)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
```

```
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.MASTER_2_PK_IDX ON TEST_USER.MASTER_2
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_2_VALUE_1_IDX ON TEST_USER.MASTER_2
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
```

```
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_2_VALUE_2_IDX ON TEST_USER.MASTER_2
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_2_VALUE_3_IDX ON TEST_USER.MASTER_2
(VALUE_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_2_VALUE_4_IDX ON TEST_USER.MASTER_2
(VALUE_4)
LOGGING
TABLESPACE TEST_DATA
```

```
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_2_VALUE_5_IDX ON TEST_USER.MASTER_2
(VALUE_5)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_3_CHAR_1_IDX ON
TEST_USER.MASTER_3
(CHAR_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
```

```
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_3_CHAR_2_IDX ON
TEST_USER.MASTER_3
(CHAR_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_3_CHAR_3_IDX ON
TEST_USER.MASTER_3
(CHAR_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE INDEX TEST_USER.MASTER_3_LINKER ON TEST_USER.MASTER_3
(MASTER_1_LINK, MASTER_2_LINK)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_3_NUM_1_IDX ON TEST_USER.MASTER_3
(NUM_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_3_NUM_2_IDX ON TEST_USER.MASTER_3
(NUM_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
```

```
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_3_NUM_3_IDX ON TEST_USER.MASTER_3
(NUM_3)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_3_NUM_4_IDX ON TEST_USER.MASTER_3
(NUM_4)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;
```

```
CREATE BITMAP INDEX TEST_USER.MASTER_3_NUM_5_IDX ON TEST_USER.MASTER_3
(NUM_5)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE UNIQUE INDEX TEST_USER.MASTER_3_PK_IDX ON TEST_USER.MASTER_3
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_3_VALUE_1_IDX ON TEST_USER.MASTER_3
(VALUE_1)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
```

```
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_3_VALUE_2_IDX ON TEST_USER.MASTER_3
(VALUE_2)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL
INVISIBLE;


ALTER TABLE TEST_USER.MASTER_1 ADD (
  CONSTRAINT MASTER_1_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.MASTER_1_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L2_TAB1 ADD (
  CONSTRAINT M1_L2_T1_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M1_L2_T1_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L2_TAB2 ADD (
  CONSTRAINT M1_L2_T2_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M1_L2_T2_PK_IDX
```

```
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L2_TAB3 ADD (
  CONSTRAINT M1_L2_T3_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M1_L2_T3_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L3_TAB1 ADD (
  CONSTRAINT M1_L3_T1_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M1_L3_T1_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L3_TAB2 ADD (
  CONSTRAINT M1_L3_T2_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M1_L3_T2_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L3_TAB3 ADD (
  CONSTRAINT M1_L3_T3_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M1_L3_T3_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2 ADD (
  CONSTRAINT MASTER_2_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.MASTER_2_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L2_TAB1 ADD (
  CONSTRAINT M2_L2_T1_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M2_L2_T1_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L2_TAB2 ADD (
  CONSTRAINT M2_L2_TAB2_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M2_L2_TAB2_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L2_TAB3 ADD (
```

```
  CONSTRAINT M2_L2_T3_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M2_L2_T3_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L3_TAB1 ADD (
  CONSTRAINT M2_L3_T1_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M2_L3_T1_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L3_TAB2 ADD (
  CONSTRAINT M2_L3_T2_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M2_L3_T2_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L3_TAB3 ADD (
  CONSTRAINT M2_L3_T3_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M2_L3_T3_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3 ADD (
  CONSTRAINT MASTER_3_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.MASTER_3_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L2_TAB1 ADD (
  CONSTRAINT M3_L2_TB_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M3_L2_TB_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L2_TAB2 ADD (
  CONSTRAINT M3_L2_T2_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M3_L2_T2_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L2_TAB3 ADD (
  CONSTRAINT M3_L2_T3_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
```

```
  USING INDEX TEST_USER.M3_L2_T3_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L3_TAB1 ADD (
  CONSTRAINT M3_L3_T1_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M3_L3_T1_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L3_TAB2 ADD (
  CONSTRAINT M3_L3_T2_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M3_L3_T2_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L3_TAB3 ADD (
  CONSTRAINT M3_L3_T3_PK_IDX
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.M3_L3_T3_PK_IDX
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L2_TAB1 ADD (
  CONSTRAINT M1_L2_T1_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_1 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L2_TAB2 ADD (
  CONSTRAINT M1_L2_T2_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_1 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L2_TAB3 ADD (
  CONSTRAINT M1_L2_T3_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_1 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L3_TAB1 ADD (
  CONSTRAINT M1_L3_T1_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_1_L2_TAB1 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L3_TAB2 ADD (
  CONSTRAINT M1_L3_T2_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_1_L2_TAB2 (PK_VALUE)
```

```
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_1_L3_TAB3 ADD (
  CONSTRAINT M1_L3_T3_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_1_L2_TAB3 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L2_TAB1 ADD (
  CONSTRAINT M2_L2_T1_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_2 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L2_TAB2 ADD (
  CONSTRAINT M2_L2_T2_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_2 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L2_TAB3 ADD (
  CONSTRAINT M2_L2_T3_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_2 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L3_TAB1 ADD (
  CONSTRAINT M2_L3_T1_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_2_L2_TAB1 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L3_TAB2 ADD (
  CONSTRAINT M2_L3_T2_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_2_L2_TAB2 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_2_L3_TAB3 ADD (
  CONSTRAINT M2_L3_T3_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_2_L2_TAB3 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L2_TAB1 ADD (
  CONSTRAINT M3_L2_T1_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_3 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L2_TAB2 ADD (
  CONSTRAINT M3_L2_T2_LINK
```

```
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_3 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L2_TAB3 ADD (
  CONSTRAINT M3_L2_T3_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_3 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L3_TAB1 ADD (
  CONSTRAINT M3_L3_T1_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_3_L2_TAB1 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L3_TAB2 ADD (
  CONSTRAINT M3_L3_T2_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_3_L2_TAB2 (PK_VALUE)
  ENABLE VALIDATE);

ALTER TABLE TEST_USER.MASTER_3_L3_TAB3 ADD (
  CONSTRAINT M3_L3_T3_LINK
  FOREIGN KEY (PARENT_LINK)
  REFERENCES TEST_USER.MASTER_3_L2_TAB3 (PK_VALUE)
  ENABLE VALIDATE);

CREATE TABLE TEST_USER.MASTER_4
(
  PK_VALUE  NUMBER                             NOT NULL,
  DATE_1    DATE,
  DATE_2    DATE,
  DATE_3    DATE,
  CHAR_1    CHAR(1 BYTE),
  CHAR_2    CHAR(1 BYTE),
  CHAR_3    CHAR(1 BYTE),
  VALUE_1   VARCHAR2(10 BYTE),
  VALUE_2   VARCHAR2(10 BYTE),
  VALUE_3   VARCHAR2(10 BYTE)
)
TABLESPACE TEST_DATA
RESULT_CACHE (MODE DEFAULT)
PCTUSED    0
PCTFREE    0
INITRANS   1
MAXTRANS   255
STORAGE    (
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
```

```
LOGGING
PARTITION BY RANGE (DATE_1)
SUBPARTITION BY LIST (CHAR_1)
SUBPARTITION TEMPLATE
  (SUBPARTITION SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E', 'F', 'G',
'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U',
'V', 'W', 'Y', 'X', 'Z'),
   SUBPARTITION SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e', 'f', 'g',
'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u',
'v', 'w', 'x', 'y', 'z'),
   SUBPARTITION SP_1_0_3 VALUES ('1', '2', '3', '4', '5', '6', '7',
'8', '9', '0')
  )
(
  PARTITION P_DATE_0_0 VALUES LESS THAN (TO_DATE(' 1999-12-31
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_0_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_0_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_0_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_0 VALUES LESS THAN (TO_DATE(' 2000-01-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_1_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_0 VALUES LESS THAN (TO_DATE(' 2000-02-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_0 VALUES LESS THAN (TO_DATE(' 2000-03-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_0 VALUES LESS THAN (TO_DATE(' 2000-04-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
```

```
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_0 VALUES LESS THAN (TO_DATE(' 2000-05-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_0 VALUES LESS THAN (TO_DATE(' 2000-06-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
```

```
  ( SUBPARTITION P_DATE_6_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_0 VALUES LESS THAN (TO_DATE(' 2000-07-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_0 VALUES LESS THAN (TO_DATE(' 2000-08-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_0 VALUES LESS THAN (TO_DATE(' 2000-09-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
```

```
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_0 VALUES LESS THAN (TO_DATE(' 2000-10-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_0 VALUES LESS THAN (TO_DATE(' 2000-11-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
```

```
               )
  ( SUBPARTITION P_DATE_11_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_11_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_11_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_0 VALUES LESS THAN (TO_DATE(' 2000-12-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_0_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_12_0_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_12_0_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_1 VALUES LESS THAN (TO_DATE(' 2001-01-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_1_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_1_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
```

```
  PARTITION P_DATE_2_1 VALUES LESS THAN (TO_DATE(' 2001-02-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_1 VALUES LESS THAN (TO_DATE(' 2001-03-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_1 VALUES LESS THAN (TO_DATE(' 2001-04-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
```

```
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_1 VALUES LESS THAN (TO_DATE(' 2001-05-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_1 VALUES LESS THAN (TO_DATE(' 2001-06-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_6_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_1 VALUES LESS THAN (TO_DATE(' 2001-07-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_1 VALUES LESS THAN (TO_DATE(' 2001-08-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_1 VALUES LESS THAN (TO_DATE(' 2001-09-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
```

```
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_1 VALUES LESS THAN (TO_DATE(' 2001-10-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_1 VALUES LESS THAN (TO_DATE(' 2001-11-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_11_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e', 
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5', 
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_1 VALUES LESS THAN (TO_DATE(' 2001-12-01 
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_1_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E', 
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e', 
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_3 VALUES ('1', '2', '3', '4', '5', 
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_2 VALUES LESS THAN (TO_DATE(' 2002-01-01 
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E', 
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e', 
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5', 
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_2 VALUES LESS THAN (TO_DATE(' 2002-02-01 
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
```

```
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_2 VALUES LESS THAN (TO_DATE(' 2002-03-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_2 VALUES LESS THAN (TO_DATE(' 2002-04-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
```

```
  ( SUBPARTITION P_DATE_4_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_2 VALUES LESS THAN (TO_DATE(' 2002-05-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_2 VALUES LESS THAN (TO_DATE(' 2002-06-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_2 VALUES LESS THAN (TO_DATE(' 2002-07-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
```

```
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_2 VALUES LESS THAN (TO_DATE(' 2002-08-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_2 VALUES LESS THAN (TO_DATE(' 2002-09-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
```

```
               )
  ( SUBPARTITION P_DATE_9_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_2 VALUES LESS THAN (TO_DATE(' 2002-10-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_2 VALUES LESS THAN (TO_DATE(' 2002-11-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
```

```
  PARTITION P_DATE_12_2 VALUES LESS THAN (TO_DATE(' 2002-12-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_2_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_9 VALUES LESS THAN (MAXVALUE)
    NOLOGGING
    NOCOMPRESS
    TABLESPACE TEST_DATA
    PCTFREE    10
    INITRANS   1
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_9_SP_1_0_1 VALUES ('A', 'B', 'C', 'D', 'E',
'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'Y', 'X', 'Z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_2 VALUES ('a', 'b', 'c', 'd', 'e',
'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's',
't', 'u', 'v', 'w', 'x', 'y', 'z')      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_3 VALUES ('1', '2', '3', '4', '5',
'6', '7', '8', '9', '0')      TABLESPACE TEST_DATA )
)
COMPRESS BASIC
NOCACHE
NOPARALLEL
MONITORING;


CREATE BITMAP INDEX TEST_USER.MASTER_4_CHAR_1_IDX ON
TEST_USER.MASTER_4
(CHAR_1)
  PCTFREE    10
```

```
  INITRANS    2
  MAXTRANS    255
  STORAGE     (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
LOGGING
LOCAL (
  PARTITION P_DATE_0_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_0_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_0_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_0_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_1_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_2_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_0
    LOGGING
    PCTFREE    10
    INITRANS   2
```

```
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_0
    LOGGING
    PCTFREE    10
    INITRANS   2
```

```
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_7_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_8_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_8_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_9_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_9_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_10_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_10_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_11_0
  LOGGING
  PCTFREE    10
  INITRANS   2
```

```
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_1
    LOGGING
    PCTFREE    10
    INITRANS   2
```

```
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_1
    LOGGING
    PCTFREE    10
    INITRANS   2
```

```
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_1
    LOGGING
    PCTFREE    10
    INITRANS   2
```

```
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_11_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_1
    LOGGING
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_12_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_2
    LOGGING
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_1_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_2
    LOGGING
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_2_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_2
    LOGGING
    PCTFREE     10
    INITRANS    2
```

```
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL       DEFAULT
                FLASH_CACHE       DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL       DEFAULT
                FLASH_CACHE       DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL       DEFAULT
                FLASH_CACHE       DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL       DEFAULT
                FLASH_CACHE       DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_2
    LOGGING
    PCTFREE    10
    INITRANS   2
```

```
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_2
    LOGGING
    PCTFREE    10
    INITRANS   2
```

```
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_9
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_9_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_3      TABLESPACE TEST_DATA )
)
NOPARALLEL;


CREATE BITMAP INDEX TEST_USER.MASTER_4_CHAR_2_IDX ON
TEST_USER.MASTER_4
(CHAR_2)
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
LOGGING
LOCAL (
```

```
PARTITION P_DATE_0_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_0_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_0_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_0_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_1_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_2_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_3_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_3      TABLESPACE TEST_DATA ),
```

```
PARTITION P_DATE_4_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_4_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_5_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_5_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_6_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_6_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_7_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_7_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_0_SP_1_0_3      TABLESPACE TEST_DATA ),
```

```
PARTITION P_DATE_8_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_8_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_9_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_9_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_10_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_10_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_11_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_11_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_0_SP_1_0_3      TABLESPACE TEST_DATA ),
```

```
PARTITION P_DATE_12_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_12_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_1_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_2_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_3_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_1_SP_1_0_3      TABLESPACE TEST_DATA ),
```

```
PARTITION P_DATE_4_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_4_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_5_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_5_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_6_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_6_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_7_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_7_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_1_SP_1_0_3      TABLESPACE TEST_DATA ),
```

```
PARTITION P_DATE_8_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_8_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_9_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_9_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_10_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_10_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_11_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_11_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_1_SP_1_0_3      TABLESPACE TEST_DATA ),
```

```
PARTITION P_DATE_12_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_12_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_2
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_1_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_2
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_2_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_2
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_3_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_2_SP_1_0_3      TABLESPACE TEST_DATA ),
```

```
PARTITION P_DATE_4_2
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_4_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_5_2
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_5_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_6_2
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_6_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_7_2
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_7_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_2_SP_1_0_3      TABLESPACE TEST_DATA ),
```

```
PARTITION P_DATE_8_2
  LOGGING
  PCTFREE     10
  INITRANS    2
  MAXTRANS    255
  STORAGE     (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
( SUBPARTITION P_DATE_8_2_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_2_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_2_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_9_2
  LOGGING
  PCTFREE     10
  INITRANS    2
  MAXTRANS    255
  STORAGE     (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
( SUBPARTITION P_DATE_9_2_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_2_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_2_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_10_2
  LOGGING
  PCTFREE     10
  INITRANS    2
  MAXTRANS    255
  STORAGE     (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
( SUBPARTITION P_DATE_10_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_11_2
  LOGGING
  PCTFREE     10
  INITRANS    2
  MAXTRANS    255
  STORAGE     (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
( SUBPARTITION P_DATE_11_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_2_SP_1_0_3      TABLESPACE TEST_DATA ),
```

```
  PARTITION P_DATE_12_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_9
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_9_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_3      TABLESPACE TEST_DATA )
)
NOPARALLEL
INVISIBLE;


CREATE BITMAP INDEX TEST_USER.MASTER_4_CHAR_3_IDX ON
TEST_USER.MASTER_4
(CHAR_3)
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
LOGGING
LOCAL (
  PARTITION P_DATE_0_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
```

```
             CELL_FLASH_CACHE DEFAULT
            )
( SUBPARTITION P_DATE_0_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_0_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_0_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
( SUBPARTITION P_DATE_1_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
( SUBPARTITION P_DATE_2_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
( SUBPARTITION P_DATE_3_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_4_0
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
```

```
              CELL_FLASH_CACHE DEFAULT
             )
  ( SUBPARTITION P_DATE_4_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
  ( SUBPARTITION P_DATE_5_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
  ( SUBPARTITION P_DATE_6_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
  ( SUBPARTITION P_DATE_7_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
```

```
               CELL_FLASH_CACHE DEFAULT
              )
  ( SUBPARTITION P_DATE_8_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
  ( SUBPARTITION P_DATE_9_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
  ( SUBPARTITION P_DATE_10_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
  ( SUBPARTITION P_DATE_11_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_0
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
```

```
              CELL_FLASH_CACHE DEFAULT
             )
  ( SUBPARTITION P_DATE_12_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_1
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
```

```
            CELL_FLASH_CACHE DEFAULT
           )
( SUBPARTITION P_DATE_4_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_5_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
( SUBPARTITION P_DATE_5_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_6_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
( SUBPARTITION P_DATE_6_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_7_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
( SUBPARTITION P_DATE_7_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_8_1
  LOGGING
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
```

```
            CELL_FLASH_CACHE DEFAULT
           )
 ( SUBPARTITION P_DATE_8_1_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_8_1_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_8_1_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_9_1
   LOGGING
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
 ( SUBPARTITION P_DATE_9_1_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_9_1_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_9_1_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_10_1
   LOGGING
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
 ( SUBPARTITION P_DATE_10_1_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_10_1_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_10_1_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_11_1
   LOGGING
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
 ( SUBPARTITION P_DATE_11_1_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_11_1_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_11_1_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_12_1
   LOGGING
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
```

```
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_1_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_2_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_2_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_2_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_2_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_2_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
```

```
             CELL_FLASH_CACHE DEFAULT
            )
 ( SUBPARTITION P_DATE_4_2_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_4_2_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_4_2_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_5_2
   LOGGING
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
 ( SUBPARTITION P_DATE_5_2_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_5_2_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_5_2_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_6_2
   LOGGING
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
 ( SUBPARTITION P_DATE_6_2_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_6_2_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_6_2_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_7_2
   LOGGING
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
 ( SUBPARTITION P_DATE_7_2_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_7_2_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_7_2_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_8_2
   LOGGING
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
```

```
              CELL_FLASH_CACHE DEFAULT
             )
  ( SUBPARTITION P_DATE_8_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
  ( SUBPARTITION P_DATE_9_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
  ( SUBPARTITION P_DATE_10_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
  ( SUBPARTITION P_DATE_11_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_2
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
```

```
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_9
    LOGGING
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_9_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_3      TABLESPACE TEST_DATA )
)
NOPARALLEL
INVISIBLE;


CREATE INDEX TEST_USER.MASTER_4_DATE_1_IDX ON TEST_USER.MASTER_4
(DATE_1)
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
LOGGING
LOCAL (
  PARTITION P_DATE_0_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_0_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_0_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_0_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_0
    LOGGING
    NOCOMPRESS
```

```
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_0_SP_1_0_2      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_4_0_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_0_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_0_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_0_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
```

```
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_8_0_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_0_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_0_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_9_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_9_0_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_0_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_0_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_10_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_10_0_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_0_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_0_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_11_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_11_0_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_0_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_0_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_12_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
```

```
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_12_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_1_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_2_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_3_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_4_1
```

```
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
```

```
( SUBPARTITION P_DATE_7_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_8_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_8_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_9_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_9_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_10_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_10_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_11_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
```

```
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_2
    LOGGING
    NOCOMPRESS
```

```
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_2      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_6_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
```

```
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_2_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_2_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_2_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_9
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_9_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_3      TABLESPACE TEST_DATA )
)
NOPARALLEL
INVISIBLE
COMPRESS 1;
```

```
CREATE INDEX TEST_USER.MASTER_4_DATE_2_IDX ON TEST_USER.MASTER_4
(DATE_2)
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
LOGGING
LOCAL (
  PARTITION P_DATE_0_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_0_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_0_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_0_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
```

```
( SUBPARTITION P_DATE_2_0_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_3_0_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_4_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_4_0_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_0_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_0_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_5_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_5_0_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_0_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_0_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_6_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
```

```
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_0
    LOGGING
    NOCOMPRESS
```

```
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL       DEFAULT
                 FLASH_CACHE       DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_10_0_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_0_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_0_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_0
    LOGGING
    NOCOMPRESS
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL       DEFAULT
                 FLASH_CACHE       DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_11_0_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_0_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_0_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_0
    LOGGING
    NOCOMPRESS
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL       DEFAULT
                 FLASH_CACHE       DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_12_0_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_0_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_0_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_1
    LOGGING
    NOCOMPRESS
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL       DEFAULT
                 FLASH_CACHE       DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_1_1_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_1_SP_1_0_2       TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_1_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
```

```
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
```

```
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_2
```

```
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
```

```
( SUBPARTITION P_DATE_4_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_5_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_5_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_6_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_6_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_7_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_7_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_8_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
```

```
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_2
    LOGGING
    NOCOMPRESS
```

```
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_9
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_9_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_3      TABLESPACE TEST_DATA )
)
NOPARALLEL
INVISIBLE
COMPRESS 1;


CREATE INDEX TEST_USER.MASTER_4_DATE_3_IDX ON TEST_USER.MASTER_4
(DATE_3)
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
LOGGING
LOCAL (
  PARTITION P_DATE_0_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
```

```
               CELL_FLASH_CACHE DEFAULT
              )
( SUBPARTITION P_DATE_0_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_0_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_0_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
( SUBPARTITION P_DATE_1_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
( SUBPARTITION P_DATE_2_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
( SUBPARTITION P_DATE_3_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_4_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
```

```
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_0
```

```
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_8_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_9_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_9_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_10_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_10_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_11_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
```

```
( SUBPARTITION P_DATE_11_0_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_0_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_0_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_12_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_12_0_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_0_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_0_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_1_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_2_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
```

```
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_1
    LOGGING
    NOCOMPRESS
```

```
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_1_SP_1_0_2      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_10_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
```

```
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_2_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_3_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_4_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_4_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_5_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_5_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_6_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
```

```
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_2
```

```
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_9
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
```

```
  ( SUBPARTITION P_DATE_9_9_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_3      TABLESPACE TEST_DATA )
)
NOPARALLEL
COMPRESS 1;


CREATE UNIQUE INDEX TEST_USER.MASTER_4_PK ON TEST_USER.MASTER_4
(PK_VALUE)
LOGGING
TABLESPACE TEST_DATA
PCTFREE    10
INITRANS   2
MAXTRANS   255
STORAGE    (
            INITIAL          64K
            NEXT             1M
            MINEXTENTS       1
            MAXEXTENTS       UNLIMITED
            PCTINCREASE      0
            BUFFER_POOL      DEFAULT
            FLASH_CACHE      DEFAULT
            CELL_FLASH_CACHE DEFAULT
           )
NOPARALLEL;


CREATE INDEX TEST_USER.MASTER_4_VALUE_1_IDX ON TEST_USER.MASTER_4
(VALUE_1)
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
LOGGING
LOCAL (
  PARTITION P_DATE_0_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_0_0_SP_1_0_1      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_0_0_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_0_0_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_0_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_0_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_0_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_0_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_0_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_0_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_0_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_0_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_0_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
```

```
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
( SUBPARTITION P_DATE_4_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_5_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
( SUBPARTITION P_DATE_5_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_6_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
( SUBPARTITION P_DATE_6_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_7_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
( SUBPARTITION P_DATE_7_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_8_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
```

```
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_8_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_9_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_9_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_10_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_10_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_11_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_11_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_0_SP_1_0_3      TABLESPACE TEST_DATA ),
```

```
PARTITION P_DATE_12_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_12_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_1_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_2_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
```

```
              )
( SUBPARTITION P_DATE_3_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_4_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_4_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_5_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_5_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_6_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_6_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_6_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_7_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
```

```
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_7_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_7_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_8_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_8_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_8_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_9_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_9_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_9_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_10_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_10_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_11_1
  LOGGING
```

```
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_2_2_SP_1_0_1      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_2_2_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_2_2_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_3_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_3_2_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_2_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_2_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_2_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_2_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_2_SP_1_0_1       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_2_SP_1_0_2       TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_2_SP_1_0_3       TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
```

```
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
```

```
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_10_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_2
    LOGGING
    NOCOMPRESS
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_11_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_2
    LOGGING
    NOCOMPRESS
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_12_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_9
    LOGGING
    NOCOMPRESS
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_9_9_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_3      TABLESPACE TEST_DATA )
```

```
)
NOPARALLEL
COMPRESS 1;


CREATE INDEX TEST_USER.MASTER_4_VALUE_2_IDX ON TEST_USER.MASTER_4
(VALUE_2)
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
LOGGING
LOCAL (
  PARTITION P_DATE_0_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_0_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_0_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_0_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_1_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_1_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_2_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
```

```
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
( SUBPARTITION P_DATE_2_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
( SUBPARTITION P_DATE_3_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_4_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
( SUBPARTITION P_DATE_4_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_5_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
( SUBPARTITION P_DATE_5_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_6_0
  LOGGING
  NOCOMPRESS
```

```
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_0_SP_1_0_2      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_9_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_1_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
```

```
              CELL_FLASH_CACHE DEFAULT
             )
 ( SUBPARTITION P_DATE_1_1_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_1_1_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_1_1_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_2_1
   LOGGING
   NOCOMPRESS
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
 ( SUBPARTITION P_DATE_2_1_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_2_1_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_2_1_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_3_1
   LOGGING
   NOCOMPRESS
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
 ( SUBPARTITION P_DATE_3_1_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_3_1_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_3_1_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_4_1
   LOGGING
   NOCOMPRESS
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
 ( SUBPARTITION P_DATE_4_1_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_4_1_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_4_1_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_5_1
   LOGGING
   NOCOMPRESS
   PCTFREE    10
   INITRANS   2
```

```
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_1
```

```
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_1
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
```

```
( SUBPARTITION P_DATE_12_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_1_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_2_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_3_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_4_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
```

```
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_2
    LOGGING
    NOCOMPRESS
```

```
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_2      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_11_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_9
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_9_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_3      TABLESPACE TEST_DATA )
)
NOPARALLEL
COMPRESS 1;


CREATE INDEX TEST_USER.MASTER_4_VALUE_3_IDX ON TEST_USER.MASTER_4
(VALUE_3)
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
LOGGING
LOCAL (
  PARTITION P_DATE_0_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
```

```
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_0_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_0_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_0_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_1_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_2_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_3_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_4_0
  LOGGING
```

```
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_4_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_5_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_0_SP_1_0_1      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_7_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_0_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_0_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_0_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_0
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
```

```
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
( SUBPARTITION P_DATE_11_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_12_0
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
( SUBPARTITION P_DATE_12_0_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_0_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_0_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
( SUBPARTITION P_DATE_1_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
             BUFFER_POOL      DEFAULT
             FLASH_CACHE      DEFAULT
             CELL_FLASH_CACHE DEFAULT
            )
( SUBPARTITION P_DATE_2_1_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_1_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_1_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
```

```
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_3_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_3_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_4_1
    LOGGING
    NOCOMPRESS
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_4_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_4_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_5_1
    LOGGING
    NOCOMPRESS
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_5_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_5_1_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_6_1
    LOGGING
    NOCOMPRESS
    PCTFREE     10
    INITRANS    2
    MAXTRANS    255
    STORAGE     (
                 BUFFER_POOL      DEFAULT
                 FLASH_CACHE      DEFAULT
                 CELL_FLASH_CACHE DEFAULT
                )
  ( SUBPARTITION P_DATE_6_1_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_1_SP_1_0_3      TABLESPACE TEST_DATA ),
```

```
 PARTITION P_DATE_7_1
   LOGGING
   NOCOMPRESS
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
 ( SUBPARTITION P_DATE_7_1_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_7_1_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_7_1_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_8_1
   LOGGING
   NOCOMPRESS
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
 ( SUBPARTITION P_DATE_8_1_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_8_1_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_8_1_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_9_1
   LOGGING
   NOCOMPRESS
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
              )
 ( SUBPARTITION P_DATE_9_1_SP_1_0_1      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_9_1_SP_1_0_2      TABLESPACE TEST_DATA,
   SUBPARTITION P_DATE_9_1_SP_1_0_3      TABLESPACE TEST_DATA ),
 PARTITION P_DATE_10_1
   LOGGING
   NOCOMPRESS
   PCTFREE    10
   INITRANS   2
   MAXTRANS   255
   STORAGE    (
               BUFFER_POOL      DEFAULT
               FLASH_CACHE      DEFAULT
               CELL_FLASH_CACHE DEFAULT
```

```
                )
( SUBPARTITION P_DATE_10_1_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_1_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_10_1_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_11_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_11_1_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_1_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_11_1_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_12_1
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_12_1_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_1_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_12_1_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_1_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_1_2_SP_1_0_1       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_2_SP_1_0_2       TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_1_2_SP_1_0_3       TABLESPACE TEST_DATA ),
PARTITION P_DATE_2_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
```

```
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_2_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_2_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_3_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_3_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_3_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_4_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_4_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_4_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_5_2
  LOGGING
  NOCOMPRESS
  PCTFREE    10
  INITRANS   2
  MAXTRANS   255
  STORAGE    (
              BUFFER_POOL      DEFAULT
              FLASH_CACHE      DEFAULT
              CELL_FLASH_CACHE DEFAULT
             )
( SUBPARTITION P_DATE_5_2_SP_1_0_1      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_2_SP_1_0_2      TABLESPACE TEST_DATA,
  SUBPARTITION P_DATE_5_2_SP_1_0_3      TABLESPACE TEST_DATA ),
PARTITION P_DATE_6_2
  LOGGING
```

```
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_6_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_6_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_7_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_7_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_7_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_8_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_8_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_8_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_2_SP_1_0_1      TABLESPACE TEST_DATA,
```

```
    SUBPARTITION P_DATE_9_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_10_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_10_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_10_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_11_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_11_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_11_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_12_2
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_12_2_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_12_2_SP_1_0_3      TABLESPACE TEST_DATA ),
  PARTITION P_DATE_9_9
    LOGGING
    NOCOMPRESS
    PCTFREE    10
    INITRANS   2
    MAXTRANS   255
    STORAGE    (
                BUFFER_POOL      DEFAULT
```

```
                FLASH_CACHE      DEFAULT
                CELL_FLASH_CACHE DEFAULT
               )
  ( SUBPARTITION P_DATE_9_9_SP_1_0_1      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_2      TABLESPACE TEST_DATA,
    SUBPARTITION P_DATE_9_9_SP_1_0_3      TABLESPACE TEST_DATA )
)
NOPARALLEL;


ALTER TABLE TEST_USER.MASTER_4 ADD (
  CONSTRAINT MASTER_4_PK
  PRIMARY KEY
  (PK_VALUE)
  USING INDEX TEST_USER.MASTER_4_PK
  ENABLE VALIDATE);
```

Appendix 2

```
#!/usr/bin/perl

$master_1_row_number         = 1000000;
$master_1_level_2_row_number = 100000;
$master_1_level_3_row_number = 100000;

$master_2_row_number         = 1000000;
$master_2_level_2_row_number = 100000;
$master_2_level_3_row_number = 100000;

$master_3_row_number         = 1000000;
$master_3_level_2_row_number = 100000;
$master_3_level_3_row_number = 100000;

$master_4_row_number         = 30000000;

@names = ();

#-----SUB ROUTINES
#-----------------


sub process_names{

   open (NAMES, "</export/home/oracle/perl_scripts/names.txt");
   @temp_names = <NAMES>;
   foreach $name (@temp_names){
      chomp($name);
      push (@names, $name);
   }
   close NAMES;
}

sub row_number {

   my $table_name = shift;
   my $return_value;

   chomp($table_name);
   if ($table_name eq 'MASTER_1'){
      $return_value = $master_1_row_number;
   }
   elsif ($table_name eq 'MASTER_2'){
      $return_value = $master_2_row_number;
   }
   elsif ($table_name eq 'MASTER_3'){
      $return_value = $master_3_row_number;
   }
   elsif ($table_name eq 'MASTER_1_L2_TAB1' ||
```

```
          $table_name eq 'MASTER_1_L2_TAB2' ||
          $table_name eq 'MASTER_1_L2_TAB3'){
      $return_value = $master_1_level_2_row_number;
   }
   elsif ($table_name eq 'MASTER_1_L3_TAB1' ||
          $table_name eq 'MASTER_1_L3_TAB2' ||
  $table_name eq 'MASTER_1_L3_TAB3'){
      $return_value = $master_1_level_3_row_number;
   }
   elsif ($table_name eq 'MASTER_2_L2_TAB1' ||
          $table_name eq 'MASTER_2_L2_TAB2' ||
          $table_name eq 'MASTER_2_L2_TAB3'){
      $return_value = $master_2_level_2_row_number;
   }
   elsif ($table_name eq 'MASTER_2_L3_TAB1' ||
          $table_name eq 'MASTER_2_L3_TAB2' ||
          $table_name eq 'MASTER_2_L3_TAB3'){
      $return_value = $master_2_level_3_row_number;
   }
   elsif ($table_name eq 'MASTER_3_L2_TAB1' ||
          $table_name eq 'MASTER_3_L2_TAB2' ||
          $table_name eq 'MASTER_3_L2_TAB3'){
      $return_value = $master_3_level_2_row_number;
   }
   elsif ($table_name eq 'MASTER_3_L3_TAB1' ||
          $table_name eq 'MASTER_3_L3_TAB2' ||
          $table_name eq 'MASTER_3_L3_TAB3'){
      $return_value = $master_3_level_3_row_number;
   }
   elsif ($table_name eq 'MASTER_4'){
      $return_value = $master_4_row_number;
   }


   return $return_value;


}

sub sql {

   my $execute_file = shift;

   my $output = `sqlplus -S test_user\/test\@SANDBOX \@$execute_file`;
   return $output;
}

sub randomizer {

   my $type = shift;
   my $return_value;
   my $range;
```

```
   my $temp_choice;


   if ($type eq 'NUMBER'){

      #--will return numbers 0 .. 999999998
      $range = 999999999;
      $return_value = int(rand($range));
  }
   elsif ($type eq 'CHAR'){

      #--will choose between numbers and letters here
      $range = 3;
      $temp_choice = int(rand($range));

      if ($temp_choice == 0){
         #--Numbers;
         $range = 10;
         $return_value = chr(int(rand($range)) + 48);

      }
      elsif ($temp_choice == 1){
         #--Upper Case Letters
               $range = 25;
         $return_value = chr(int(rand($range)) + 65);
      }
      else{
         #--Lower Case Letters
               $range = 25;
         $return_value = chr(int(rand($range)) + 97);
      }
   }
   elsif ($type eq 'DATE'){

      #--first the year  1970--2013 = 43 year span
      $range = 43;
      my $year = int(rand($range)) + 1970;

      #--month
      $range = 12;
      my $month = int(rand($range)) + 1;

      #--day -- just avoid leap day....
      $range = 28;
      my $day = int(rand($range)) + 1;

      $return_value = 'to_date(\'' . "$year-$month-$day" . '\',\'YYYY-
MM-DD\')';
   }
   elsif ($type eq 'MASTER_4_DATE'){
```

```
      #--first the year  2000-2002 = 3 year span total: 2000, 2001,
2002
      $range = 3;
      my $year = int(rand($range)) + 2000;

      #--month
      $range = 12;
      my $month = int(rand($range)) + 1;

      #--day -- just avoid leap day....
      $range = 28;
      my $day = int(rand($range)) + 1;

      $return_value = 'to_date(\'' . "$year-$month-$day" . '\',\'YYYY-
MM-DD\')';
}

   return $return_value;
}

sub build_string {

   #--Note.  data_type will be in ('NUMBER','DATE','VARCHAR2', 'CHAR',
'CLOB')


   my $data_type   = shift;
   my $data_length = shift;
   my $column_id   = shift;
   my $table_name  = shift;
   my $return_value;

   if ($data_type =~ /NUMBER/){

      $return_value = randomizer('NUMBER');

   }

   elsif ($data_type =~ /DATE/){

      if ($table_name eq 'MASTER_4'){

         $return_value = randomizer('MASTER_4_DATE');
      }
      else{

         $return_value = randomizer('DATE');
      }

   }

   elsif ($data_type =~  /VARCHAR2/){
```

```
      if ($table_name eq 'MASTER_4'){

         #--will find a number between 0 and 499 inclusive, and pull
from the names array the first 10 characters
         $return_value = substr($names[int(rand(500))], 0, 10);
      }
      else{
         for (my $i = 0; $i < $data_length; $i++){

            $return_value .= randomizer('CHAR');

         }
      }
      $return_value = "'" . $return_value . "'";

   }

   elsif ($data_type =~ /CHAR/){

      $return_value = "'" . randomizer ('CHAR') . "'";

   }

   elsif ($data_type =~ /CLOB/){

      for (my $i = 0; $i < int(rand(1000)) + 100; $i++){ #--
Randomizing the length

         $return_value .= randomizer('CHAR');
      }
      $return_value = "'" . $return_value . "'";

   }

   return $return_value;

}

sub find_parent_row_number{

   my $table_name = shift;
   my $return_value = "";

   if ($table_name eq 'MASTER_1_L2_TAB1' ||
       $table_name eq 'MASTER_1_L2_TAB2' ||
       $table_name eq 'MASTER_1_L2_TAB3'){
       $return_value = $master_1_row_number - 1;
   }
   if ($table_name eq 'MASTER_2_L2_TAB1' ||
       $table_name eq 'MASTER_2_L2_TAB2' ||
       $table_name eq 'MASTER_2_L2_TAB3'){
```

```
        $return_value = $master_2_row_number - 1;
    }
    if ($table_name eq 'MASTER_3_L2_TAB1' ||
        $table_name eq 'MASTER_3_L2_TAB2' ||
        $table_name eq 'MASTER_3_L2_TAB3'){
        $return_value = $master_3_row_number - 1;
    }
    if ($table_name eq 'MASTER_1_L3_TAB1' ||
        $table_name eq 'MASTER_1_L3_TAB2' ||
        $table_name eq 'MASTER_1_L3_TAB3'){
        $return_value = $master_1_level_2_row_number - 1;
    }
    if ($table_name eq 'MASTER_2_L3_TAB1' ||
        $table_name eq 'MASTER_2_L3_TAB2' ||
        $table_name eq 'MASTER_2_L3_TAB3'){
        $return_value = $master_2_level_2_row_number - 1;
    }
    if ($table_name eq 'MASTER_3_L3_TAB1' ||
        $table_name eq 'MASTER_3_L3_TAB2' ||
        $table_name eq 'MASTER_3_L3_TAB3'){
        $return_value = $master_3_level_2_row_number - 1;
    }

    return $return_value;
}
sub load_table {

    my $table_name = shift;
    my $table_load_number = 0;
    my $total_load_number = 0;
    my $row_number = 0;

    my $sql_def_file =
"\/export\/home\/oracle\/perl_scripts\/sql_def_file.sql";
    my $output_file  =
"\/export\/home\/oracle\/perl_scripts\/$table_name.sql";

    open (EXE_FILE, ">$sql_def_file");
    print EXE_FILE qq[set heading off
                      set pagesize 0
                      set feedback off
                      select table_name     ||'~'||
                             column_name    ||'~'||
                             data_type      ||'~'||
                             data_Length    ||'~'||
                             column_Id
                        from user_tab_columns
                       where table_name = '$table_name'
                       order by column_Id asc
/
exit
/];
```

```
    close EXE_FILE;

    my @lines = split (/\n/, sql($sql_def_file));

    $row_number = row_number($table_name);
    print "$table_name   $row_number\n";

    open (OP, ">$output_file");
    print OP "";
    close OP;

    #--Tank up some rows.....
    for (my $i = 1; $i <= $row_number; $i++){

       $table_load_number++;
       $total_load_number++;

       my %line_value = ();
       my $insert_string = "";

       foreach $line (@lines){ #--this holdes the table_definition, so
we tank up each column one at time.

          my ($def_table_name, $column_name, $data_type, $data_length,
$column_id) = split(/~/, $line);


          if ($column_name eq 'PK_VALUE' && $data_type eq 'NUMBER'){

             $line_value{$column_name} = $i;

          }
         }
          elsif($column_name eq 'PK_VALUE2' && $data_type eq
'NUMBER'){

             $line_value{$column_name} = $i;

          }
          elsif ($column_name eq 'PARENT_LINK' && $data_type eq
'NUMBER'){

             $parent_row_number = find_parent_row_number($table_name);
             $line_value{$column_name} = int(rand($parent_row_number))
+ 1;

          }
          elsif ($column_name eq 'MASTER_1_LINK'){

             $row_number = row_number("MASTER_1");;
             $line_value{$column_name} = int(rand($row_number)) + 1;
```

```
            }
            elsif ($column_name eq 'MASTER_2_LINK'){

               $row_number = row_number("MASTER_2");;
               $line_value{$column_name} = int(rand($row_number)) + 1;

            }
            elsif ($column_name eq 'MASTER_3_LINK'){

               $row_number = row_number("MASTER_3");;
               $line_value{$column_name} = int(rand($row_number)) + 1;

            }
            else{

               $line_value{$column_name} = build_string($data_type,
$data_length, $column_id, $table_name);

            }
         }

         $insert_string = "insert into $table_name (\n";

         foreach $key (sort keys %line_value){

            $insert_string .= "$key,\n";

         }

         $insert_string .= ") values (\n";

         foreach $key (sort keys %line_value){

            $insert_string .= "$line_value{$key},\n";
         }

         $insert_string .= ");";
         $insert_string =~ s/,\n\)/\n\)/g;

         open (OP, ">>$output_file");
         print OP "$insert_string\n";
         close OP;

         #---Going to load_table into database in batches of 500
each,.....
         if ($table_load_number >= 500){

            open (OP, ">>$output_file");
            print OP "commit\n\/\n";
            print OP "exit\n\/\n";
            close OP;
```

```
            $temp_output = sql($output_file);

            #--reinitialize the output file
            open (OP, ">$output_file");
            print OP "";
            close OP;
            $table_load_number = 0;
            print "Loaded $total_load_number\n";

         }


      }
      #---Executute the final load
      open (OP, ">>$output_file");
      print OP "commit\n\/\n";
      print OP "exit\n\/\n";
      close OP;

      $temp_output = sql($output_file);
      print "Total load on $table_name was $total_load_number\n";

}


#-----MAIN SECTION to TANKUP the SYSTEM

process_names();

load_table("MASTER_4");

load_table("MASTER_1");

load_table ("MASTER_1_L2_TAB1");
load_table ("MASTER_1_L2_TAB2");
load_table ("MASTER_1_L2_TAB3");

load_table ("MASTER_1_L3_TAB1");
load_table ("MASTER_1_L3_TAB2");
load_table ("MASTER_1_L3_TAB3");

load_table("MASTER_2");
load_table ("MASTER_2_L2_TAB1");
load_table ("MASTER_2_L2_TAB2");
load_table ("MASTER_2_L2_TAB3");

load_table ("MASTER_2_L3_TAB1");
load_table ("MASTER_2_L3_TAB2");
load_table ("MASTER_2_L3_TAB3");

load_table("MASTER_3");
```

```
load_table ("MASTER_3_L2_TAB1");
load_table ("MASTER_3_L2_TAB2");
load_table ("MASTER_3_L2_TAB3");

load_table ("MASTER_3_L3_TAB1");
load_table ("MASTER_3_L3_TAB2");
load_table ("MASTER_3_L3_TAB3")
```

Appendix 3

```
--INDEXES
MASTER_1_CHAR_1_IDX
MASTER_1_CHAR_2_IDX
MASTER_1_CHAR_3_IDX
MASTER_1_DATE_1_IDX
MASTER_1_DATE_2_IDX
MASTER_1_DATE_3_IDX
MASTER_1_DATE_4_IDX
MASTER_1_DATE_5_IDX
MASTER_1_VALUE_1_IDX
MASTER_1_VALUE_2_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
select count(*)
from master_1
where date_1 in (select date_1 from master_1 where date_1 between
to_date('01-JAN-1970','DD-MON-YYYY') and sysdate)
  and date_2 in (select date_2 from master_1 where date_2 between
to_date('01-JAN-1971','DD-MON-YYYY') and sysdate)
  and date_3 in (select date_3 from master_1 where date_3 between
to_date('01-JAN-1972','DD-MON-YYYY') and sysdate)
  and date_4 in (select date_4 from master_1 where date_4 between
to_date('01-JAN-1973','DD-MON-YYYY') and sysdate)
  and date_5 in (select date_5 from master_1 where date_5 between
to_date('01-JAN-1974','DD-MON-YYYY') and sysdate)
  and char_1 in (select char_1 from master_1 where char_1 between 'A'
and 'Z')
  and char_2 in (select char_2 from master_1 where char_2 between 'A'
and 'Z')
  and char_3 in (select char_3 from master_1 where char_3 between 'A'
and 'Z')
  and (value_1 like 'a%' or value_1 like 'b%')
  and (value_2 like 'A%' or value_2 like 'B%')
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_1_QUERY*/
```

```
count(*)
from master_1
where date_1 in (select date_1 from master_1 where date_1 between
to_date('01-JAN-1970','DD-MON-YYYY') and sysdate)
  and date_2 in (select date_2 from master_1 where date_2 between
to_date('01-JAN-1971','DD-MON-YYYY') and sysdate)
  and date_3 in (select date_3 from master_1 where date_3 between
to_date('01-JAN-1972','DD-MON-YYYY') and sysdate)
  and date_4 in (select date_4 from master_1 where date_4 between
to_date('01-JAN-1973','DD-MON-YYYY') and sysdate)
  and date_5 in (select date_5 from master_1 where date_5 between
to_date('01-JAN-1974','DD-MON-YYYY') and sysdate)
  and char_1 in (select char_1 from master_1 where char_1 between 'A'
and 'Z')
  and char_2 in (select char_2 from master_1 where char_2 between 'A'
and 'Z')
  and char_3 in (select char_3 from master_1 where char_3 between 'A'
and 'Z')
  and (value_1 like 'a%' or value_1 like 'b%')
  and (value_2 like 'A%' or value_2 like 'B%')
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_1_QUERY%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
--------------------------------------------
--INDEXES
MASTER_2_CHAR_1_IDX
MASTER_2_CHAR_2_IDX
MASTER_2_CHAR_3_IDX
MASTER_2_DATE_1_IDX
MASTER_2_DATE_2_IDX
MASTER_2_VALUE_1_IDX
MASTER_2_VALUE_2_IDX
MASTER_2_VALUE_3_IDX
MASTER_2_VALUE_4_IDX
MASTER_2_VALUE_5_IDX

--GET_COST
delete from plan_table
```

```
/
explain plan
set statement_id = 'A' for
select count(*)
from master_2
where value_1 in (select value_1 from master_2 where value_1 like
'%A%')
  and value_2 in (select value_2 from master_2 where value_2 like
'%B%')
  and value_3 in (select value_3 from master_2 where value_3 like
'%C%')
  and (value_4 like 'A%' or
       value_4 like 'B%' or
       value_4 like 'C%' or
       value_4 like 'D%' or
       value_4 like 'E%')
  and (value_5 like '0%' or value_5 like 'b%' or value_5 like '0%' or
value_5 like '7%' or value_5 like 'Z%')
  and (char_1 between 'a' and 'z')
  and (char_2 between 'a' and 'z' or char_2 between 'A' and 'Z')
  and (char_3 between 'a' and 'z' or char_3 between 'A' and 'Z' or
char_3 between '0' and '9')
  and date_1 in (select date_1 from master_2 where date_1 between
to_date('01-JAN-1970','DD-MON-YYYY') and sysdate)
  and date_2 in (select date_2 from master_2 where date_2 between
to_date('01-JAN-1971','DD-MON-YYYY') and sysdate)
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_2_QUERY*/
count(*)
from master_2
where value_1 in (select value_1 from master_2 where value_1 like
'%A%')
  and value_2 in (select value_2 from master_2 where value_2 like
'%B%')
  and value_3 in (select value_3 from master_2 where value_3 like
'%C%')
  and (value_4 like 'A%' or
       value_4 like 'B%' or
       value_4 like 'C%' or
       value_4 like 'D%' or
       value_4 like 'E%')
```

```
  and (value_5 like '0%' or value_5 like 'b%' or value_5 like '0%' or
value_5 like '7%' or value_5 like 'Z%')
  and (char_1 between 'a' and 'z')
  and (char_2 between 'a' and 'z' or char_2 between 'A' and 'Z')
  and (char_3 between 'a' and 'z' or char_3 between 'A' and 'Z' or
char_3 between '0' and '9')
  and date_1 in (select date_1 from master_2 where date_1 between
to_date('01-JAN-1970','DD-MON-YYYY') and sysdate)
  and date_2 in (select date_2 from master_2 where date_2 between
to_date('01-JAN-1971','DD-MON-YYYY') and sysdate)
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_2_QUERY%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
--------------------------------------------
--INDEXES
MASTER_3_CHAR_1_IDX
MASTER_3_CHAR_2_IDX
MASTER_3_CHAR_3_IDX
MASTER_3_NUM_1_IDX
MASTER_3_NUM_2_IDX
MASTER_3_NUM_3_IDX
MASTER_3_NUM_4_IDX
MASTER_3_NUM_5_IDX
MASTER_3_VALUE_1_IDX
MASTER_3_VALUE_2_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
select count(*)
from master_3
where num_1 between 1 and 98768000
  and num_2 between 28 and 291381400
  and num_3 >= (select min(num_3) from master_3)
  and num_4 <= (select avg(num_4) from master_3)
  and num_5 <= (select max(num_5) from master_3)
```

```
  and (char_1 between 'a' and 'z')
  and (char_2 between 'a' and 'z' or char_2 between 'A' and 'Z')
  and (char_3 between 'a' and 'z' or char_3 between 'A' and 'Z' or
char_3 between '0' and '9')
  and value_1 in (select value_1 from master_3 where value_1 like
'%A%' or value_1 like '%B%')
  and value_2 in (select value_2 from master_3 where value_2 like
'%C%')
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_3_QUERY*/
count(*)
from master_3
where num_1 between 1 and 98768000
  and num_2 between 28 and 291381400
  and num_3 >= (select min(num_3) from master_3)
  and num_4 <= (select avg(num_4) from master_3)
  and num_5 <= (select max(num_5) from master_3)
  and (char_1 between 'a' and 'z')
  and (char_2 between 'a' and 'z' or char_2 between 'A' and 'Z')
  and (char_3 between 'a' and 'z' or char_3 between 'A' and 'Z' or
char_3 between '0' and '9')
  and value_1 in (select value_1 from master_3 where value_1 like
'%A%' or value_1 like '%B%')
  and value_2 in (select value_2 from master_3 where value_2 like
'%C%')
/
exit
/
----------------------------------------------------------------------
--------------------------------------------
--INDEXES
MASTER_1_DATE_1_IDX
MASTER_1_DATE_2_IDX
MASTER_1_DATE_3_IDX
MASTER_1_DATE_4_IDX
M1_L2_T1_DATE_1_IDX
M1_L2_T2_DATE_1_IDX
M1_L2_T3_DATE_1_IDX
M1_L2_T1_VALUE_1_IDX
M1_L2_T2_VALUE_1_IDX
M1_L2_T3_VALUE_1_IDX
```

```
--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
select count(*)
from master_1 a join master_1_l2_tab1 b on (b.parent_link =
a.pk_value)
                join master_1_l2_tab2 c on (c.parent_link =
a.pk_value)
                join master_1_l2_tab3 d on (d.parent_link =
a.pk_value)
where A.DATE_1 in (select date_1 from master_1 where date_1 between
to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and A.DATE_2 in (select date_2 from master_1 where date_2 between
to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and A.DATE_3 in (select date_3 from master_1 where date_3 between
to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and A.DATE_4 in (select date_4 from master_1 where date_4 between
to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and B.DATE_1 in (select date_1 from master_1_l2_tab1 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and C.DATE_1 in (select date_1 from master_1_l2_tab2 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and D.DATE_1 in (select date_1 from master_1_l2_tab3 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and (B.value_1 like '%A%' or B.value_1 like '%B%' or B.value_1 like
'%C%' or B.value_1 like '%D%')
  and (C.value_1 like '%1%' or C.value_1 like '%2%' or C.value_1 like
'%3%' or C.value_1 like '%4%')
  and (D.value_1 like '%w%' or D.value_1 like '%x%' or D.value_1 like
'%y%' or D.value_1 like '%z%')
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_1_JOIN_1*/
count(*)
from master_1 a join master_1_l2_tab1 b on (b.parent_link =
a.pk_value)
                join master_1_l2_tab2 c on (c.parent_link =
a.pk_value)
                join master_1_l2_tab3 d on (d.parent_link =
a.pk_value)
```

```
where A.DATE_1 in (select date_1 from master_1 where date_1 between
to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and A.DATE_2 in (select date_2 from master_1 where date_2 between
to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and A.DATE_3 in (select date_3 from master_1 where date_3 between
to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and A.DATE_4 in (select date_4 from master_1 where date_4 between
to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and B.DATE_1 in (select date_1 from master_1_l2_tab1 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and C.DATE_1 in (select date_1 from master_1_l2_tab2 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and D.DATE_1 in (select date_1 from master_1_l2_tab3 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and (B.value_1 like '%A%' or B.value_1 like '%B%' or B.value_1 like
'%C%' or B.value_1 like '%D%')
  and (C.value_1 like '%1%' or C.value_1 like '%2%' or C.value_1 like
'%3%' or C.value_1 like '%4%')
  and (D.value_1 like '%w%' or D.value_1 like '%x%' or D.value_1 like
'%y%' or D.value_1 like '%z%')
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_1_JOIN_1%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
MASTER_1_CHAR_1_IDX
MASTER_1_CHAR_2_IDX
MASTER_1_CHAR_3_IDX
MASTER_1_DATE_1_IDX
M1_L2_T1_DATE_1_IDX
M1_L3_T1_DATE_1_IDX
M1_L2_T2_DATE_1_IDX
M1_L3_T2_DATE_1_IDX
M1_L2_T3_CHAR_1_IDX
M1_L3_T3_CHAR_1_IDX

--GET_COST
delete from plan_table
```

```
/
explain plan
set statement_id = 'A' for
select count(*)
from master_1 a join master_1_l2_tab1 b on (b.parent_link =
a.pk_value)
                join master_1_l3_tab1 c on (c.parent_link =
b.pk_value)
                join master_1_l2_tab2 d on (d.parent_link =
a.pk_value)
                join master_1_l3_tab2 e on (e.parent_link =
d.pk_value)
                join master_1_l2_tab3 f on (f.parent_link =
a.pk_value)
                join master_1_l3_tab3 g on (g.parent_link =
f.pk_value)
where a.char_1 between 'A' and 'Z'
  and a.char_2 between '0' and '9'
  and (a.char_3 between 'a' and 'z' or a.char_3 between '1' and '9')
  and A.DATE_1 in (select date_1 from master_1 where date_1 between
to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and B.DATE_1 in (select date_1 from master_1_l2_tab1 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and C.DATE_1 in (select date_1 from master_1_l3_tab1 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and d.DATE_1 in (select date_1 from master_1_l2_tab2 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and e.DATE_1 in (select date_1 from master_1_l3_tab2 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and f.char_1 in ('6','D','O','T')
  and (g.char_1 between 'A' and 'Z' or g.char_1 in
('1','2','3','4','5','6'))
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_1_JOIN_2*/
count(*)
from master_1 a join master_1_l2_tab1 b on (b.parent_link =
a.pk_value)
                join master_1_l3_tab1 c on (c.parent_link =
b.pk_value)
                join master_1_l2_tab2 d on (d.parent_link =
a.pk_value)
```

```
                join master_1_l3_tab2 e on (e.parent_link =
d.pk_value)
                join master_1_l2_tab3 f on (f.parent_link =
a.pk_value)
                join master_1_l3_tab3 g on (g.parent_link =
f.pk_value)
where a.char_1 between 'A' and 'Z'
  and a.char_2 between '0' and '9'
  and (a.char_3 between 'a' and 'z' or a.char_3 between '1' and '9')
  and A.DATE_1 in (select date_1 from master_1 where date_1 between
to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and B.DATE_1 in (select date_1 from master_1_l2_tab1 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and C.DATE_1 in (select date_1 from master_1_l3_tab1 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and d.DATE_1 in (select date_1 from master_1_l2_tab2 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and e.DATE_1 in (select date_1 from master_1_l3_tab2 where date_1
between to_date('01-JAN-1984','DD-MON-YYYY') and sysdate)
  and f.char_1 in ('6','D','O','T')
  and (g.char_1 between 'A' and 'Z' or g.char_1 in
('1','2','3','4','5','6'))
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_1_JOIN_2%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
-INDEXES
MASTER_1_CHAR_1_IDX
MASTER_1_CHAR_2_IDX
MASTER_1_CHAR_3_IDX
MASTER_1_DATE_1_IDX
M1_L2_T1_DATE_1_IDX
M1_L3_T1_DATE_1_IDX
M1_L2_T2_DATE_1_IDX
M1_L3_T2_DATE_1_IDX
M1_L2_T3_CHAR_1_IDX
M1_L3_T3_CHAR_1_IDX
M1_L3_T3_VALUE_1_IDX
```

```
--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT SUM (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_1_l2_tab1 b
                  ON (b.parent_link = a.pk_value)
               JOIN master_1_l3_tab1 c
                  ON (c.parent_link = b.pk_value)
               JOIN master_1_l2_tab2 d
                  ON (d.parent_link = a.pk_value)
               JOIN master_1_l3_tab2 e
                  ON (e.parent_link = d.pk_value)
               JOIN master_1_l2_tab3 f
                  ON (f.parent_link = a.pk_value)
               JOIN master_1_l3_tab3 g
                  ON (g.parent_link = f.pk_value)
         WHERE     g.char_1 IN (SELECT char_1
                                  FROM master_1_l3_tab3
                                 WHERE char_1 BETWEEN 'A' AND 'Z')
               AND G.value_1 IN
                      (SELECT value_1
                         FROM master_1_l3_tab3
                        WHERE    value_1 LIKE '%A%'
                              OR value_1 LIKE '%B%'
                              OR value_1 LIKE '%C%')
        UNION
        SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_1_l2_tab1 b
                  ON (b.parent_link = a.pk_value)
               JOIN master_1_l3_tab1 c
                  ON (c.parent_link = b.pk_value)
               JOIN master_1_l2_tab2 d
                  ON (d.parent_link = a.pk_value)
               JOIN master_1_l3_tab2 e
                  ON (e.parent_link = d.pk_value)
               JOIN master_1_l2_tab3 f
                  ON (f.parent_link = a.pk_value)
               JOIN master_1_l3_tab3 g
                  ON (g.parent_link = f.pk_value)
         WHERE     a.char_1 BETWEEN 'A' AND 'Z'
               AND a.char_2 BETWEEN '0' AND '9'
               AND (   a.char_3 BETWEEN 'a' AND 'z'
                    OR a.char_3 BETWEEN '1' AND '9')
               AND A.DATE_1 IN
                      (SELECT date_1
                         FROM master_1
```

```
                        WHERE date_1 BETWEEN TO_DATE ('01-JAN-1984',
                                                      'DD-MON-YYYY')
                                          AND SYSDATE)
               AND B.DATE_1 IN
                      (SELECT date_1
                         FROM master_1_l2_tab1
                        WHERE date_1 BETWEEN TO_DATE ('01-JAN-1984',
                                                      'DD-MON-YYYY')
                                          AND SYSDATE)
               AND C.DATE_1 IN
                      (SELECT date_1
                         FROM master_1_l3_tab1
                        WHERE date_1 BETWEEN TO_DATE ('01-JAN-1984',
                                                      'DD-MON-YYYY')
                                          AND SYSDATE)
               AND d.DATE_1 IN
                      (SELECT date_1
                         FROM master_1_l2_tab2
                        WHERE date_1 BETWEEN TO_DATE ('01-JAN-1984',
                                                      'DD-MON-YYYY')
                                          AND SYSDATE)
               AND e.DATE_1 IN
                      (SELECT date_1
                         FROM master_1_l3_tab2
                        WHERE date_1 BETWEEN TO_DATE ('01-JAN-1984',
                                                      'DD-MON-YYYY')
                                          AND SYSDATE)
               AND f.char_1 IN ('6', 'D', 'O', 'T')
               AND (   g.char_1 BETWEEN 'A' AND 'Z'
                    OR g.char_1 IN ('1', '2', '3', '4', '5', '6')))
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_1_JOIN_3*/
SUM (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_1_l2_tab1 b
                  ON (b.parent_link = a.pk_value)
               JOIN master_1_l3_tab1 c
                  ON (c.parent_link = b.pk_value)
               JOIN master_1_l2_tab2 d
                  ON (d.parent_link = a.pk_value)
               JOIN master_1_l3_tab2 e
```

```
           ON (e.parent_link = d.pk_value)
        JOIN master_1_l2_tab3 f
           ON (f.parent_link = a.pk_value)
        JOIN master_1_l3_tab3 g
           ON (g.parent_link = f.pk_value)
  WHERE     g.char_1 IN (SELECT char_1
                           FROM master_1_l3_tab3
                          WHERE char_1 BETWEEN 'A' AND 'Z')
        AND G.value_1 IN
               (SELECT value_1
                  FROM master_1_l3_tab3
                 WHERE    value_1 LIKE '%A%'
                       OR value_1 LIKE '%B%'
                       OR value_1 LIKE '%C%')
 UNION
 SELECT COUNT (*) counter
   FROM master_1 a
        JOIN master_1_l2_tab1 b
           ON (b.parent_link = a.pk_value)
        JOIN master_1_l3_tab1 c
           ON (c.parent_link = b.pk_value)
        JOIN master_1_l2_tab2 d
           ON (d.parent_link = a.pk_value)
        JOIN master_1_l3_tab2 e
           ON (e.parent_link = d.pk_value)
        JOIN master_1_l2_tab3 f
           ON (f.parent_link = a.pk_value)
        JOIN master_1_l3_tab3 g
           ON (g.parent_link = f.pk_value)
  WHERE     a.char_1 BETWEEN 'A' AND 'Z'
        AND a.char_2 BETWEEN '0' AND '9'
        AND (   a.char_3 BETWEEN 'a' AND 'z'
             OR a.char_3 BETWEEN '1' AND '9')
        AND A.DATE_1 IN
               (SELECT date_1
                  FROM master_1
                 WHERE date_1 BETWEEN TO_DATE ('01-JAN-1984',
                                               'DD-MON-YYYY')
                                  AND SYSDATE)
        AND B.DATE_1 IN
               (SELECT date_1
                  FROM master_1_l2_tab1
                 WHERE date_1 BETWEEN TO_DATE ('01-JAN-1984',
                                               'DD-MON-YYYY')
                                  AND SYSDATE)
        AND C.DATE_1 IN
               (SELECT date_1
                  FROM master_1_l3_tab1
                 WHERE date_1 BETWEEN TO_DATE ('01-JAN-1984',
                                               'DD-MON-YYYY')
                                  AND SYSDATE)
        AND d.DATE_1 IN
```

```
                    (SELECT date_1
                       FROM master_1_l2_tab2
                      WHERE date_1 BETWEEN TO_DATE ('01-JAN-1984',
                                                    'DD-MON-YYYY')
                                       AND SYSDATE)
               AND e.DATE_1 IN
                    (SELECT date_1
                       FROM master_1_l3_tab2
                      WHERE date_1 BETWEEN TO_DATE ('01-JAN-1984',
                                                    'DD-MON-YYYY')
                                       AND SYSDATE)
               AND f.char_1 IN ('6', 'D', 'O', 'T')
               AND (    g.char_1 BETWEEN 'A' AND 'Z'
                    OR g.char_1 IN ('1', '2', '3', '4', '5', '6')))
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_1_JOIN_3%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
--------------------------------------------
--INDEXES
M2_L2_T3_PK_IDX
M3_L3_T3_CHAR_1_IDX
M3_L3_T3_CHAR_2_IDX
M1_L2_T1_PK_IDX
MASTER_1_DATE_1_IDX
M3_L3_T3_VALUE_1_IDX
MASTER_1_PK_IDX
MASTER_2_PK_IDX
MASTER_3_PK_IDX
M2_L2_T1_DATE_10_IDX
M2_L3_T1_DATE_1_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
select count(*)
from master_1 a join master_2 b on (A.PK_VALUE = B.PK_VALUE)
```

```
                join master_3 c on (B.PK_VALUE = C.PK_VALUE)
                join master_1_l2_tab1 d on (d.pk_value = a.pk_value)
                join master_1_l3_tab1 e on (e.pk_value = a.pk_value)
                join master_1_l2_tab2 f on (f.pk_value = a.pk_value)
                join master_1_l3_tab2 g on (g.pk_value = a.pk_value)
                join master_1_l2_tab3 h on (h.pk_value = a.pk_value)
                join master_1_l3_tab3 i on (i.pk_value = a.pk_value)
                join master_2_l2_tab1 j on (j.pk_value = b.pk_value)
                join master_2_l3_tab1 k on (k.pk_value = j.pk_value)
                join master_2_l2_tab2 l on (l.pk_value = b.pk_value)
                join master_2_l3_tab2 m on (m.pk_value = l.pk_value)
                join master_2_l2_tab3 n on (n.pk_value = b.pk_value)
                join master_2_l3_tab3 o on (o.pk_value = n.pk_value)
                join master_3_l2_tab1 p on (p.parent_link =
c.pk_value)
                join master_3_l3_tab1 q on (q.parent_link =
p.pk_value)
                join master_3_l2_tab2 r on (r.parent_link =
c.pk_value)
                join master_3_l3_tab2 s on (s.parent_link =
r.pk_value)
                join master_3_l2_tab3 t on (t.parent_link =
c.pk_value)
                join master_3_l3_tab3 u on (u.parent_link =
t.pk_value)
where u.char_1 between 'A' and 'Z'
  and (u.char_2 between 'A' and 'Z' or u.char_2 between '1' and '9')
  AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-YYYY')
                        AND SYSDATE
       AND s.value_3 IN
              ('92E0b1CBj0',
               'L51lHeuaJo',
               'fQaLo5msQY',
               '1tffVC1iy9',
               'LMc2250887',
               '80TiO4F81M',
               'rg68IGlfEp',
               'A7B900WG85')
 and J.DATE10 between (select min(date10) from master_2_l2_tab1) and
sysdate
 and K.DATE_1 between (select max(date_1) from master_2_l3_tab1) and
sysdate - 999999999
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
```

```
set timing on
select /*+ MASTER_ALL_JOIN_1*/
count(*)
from master_1 a join master_2 b on (A.PK_VALUE = B.PK_VALUE)
                join master_3 c on (B.PK_VALUE = C.PK_VALUE)
                join master_1_l2_tab1 d on (d.pk_value = a.pk_value)
                join master_1_l3_tab1 e on (e.pk_value = a.pk_value)
                join master_1_l2_tab2 f on (f.pk_value = a.pk_value)
                join master_1_l3_tab2 g on (g.pk_value = a.pk_value)
                join master_1_l2_tab3 h on (h.pk_value = a.pk_value)
                join master_1_l3_tab3 i on (i.pk_value = a.pk_value)
                join master_2_l2_tab1 j on (j.pk_value = b.pk_value)
                join master_2_l3_tab1 k on (k.pk_value = j.pk_value)
                join master_2_l2_tab2 l on (l.pk_value = b.pk_value)
                join master_2_l3_tab2 m on (m.pk_value = l.pk_value)
                join master_2_l2_tab3 n on (n.pk_value = b.pk_value)
                join master_2_l3_tab3 o on (o.pk_value = n.pk_value)
                join master_3_l2_tab1 p on (p.parent_link =
c.pk_value)
                join master_3_l3_tab1 q on (q.parent_link =
p.pk_value)
                join master_3_l2_tab2 r on (r.parent_link =
c.pk_value)
                join master_3_l3_tab2 s on (s.parent_link =
r.pk_value)
                join master_3_l2_tab3 t on (t.parent_link =
c.pk_value)
                join master_3_l3_tab3 u on (u.parent_link =
t.pk_value)
where u.char_1 between 'A' and 'Z'
  and (u.char_2 between 'A' and 'Z' or u.char_2 between '1' and '9')
AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-YYYY')
                        AND SYSDATE
       AND s.value_3 IN
              ('92E0b1CBj0',
               'L51lHeuaJo',
               'fQaLo5msQY',
               '1tffVC1iy9',
               'LMc2250887',
               '80TiO4F81M',
               'rg68IGlfEp',
               'A7B900WG85')
 and J.DATE10 between (select min(date10) from master_2_l2_tab1) and
sysdate
 and K.DATE_1 between (select max(date_1) from master_2_l3_tab1) and
sysdate - 999999999
/
exit
/

--CROSS_CHECK_COST
set heading off
```

```
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_ALL_JOIN_1%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
--------------------------------------------------------------------------------
-------------------------------------------
--INDEXES
M2_L2_T3_PK_IDX
M1_L2_T1_PK_IDX
M3_L3_T3_VALUE_1_IDX
MASTER_1_PK_IDX
MASTER_3_PK_IDX
M2_L2_T1_DATE_10_IDX
M2_L3_T1_DATE_1_IDX
MASTER_1_CHAR_1_IDX
MASTER_1_DATE_2_IDX
MASTER_1_VALUE_1_IDX
MASTER_1_VALUE_2_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT SUM(counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
                  ON (B.PK_VALUE = C.PK_VALUE)
               JOIN master_1_l2_tab1 d
                  ON (d.pk_value = a.pk_value)
               JOIN master_1_l3_tab1 e
                  ON (e.pk_value = a.pk_value)
               JOIN master_1_l2_tab2 f
                  ON (f.pk_value = a.pk_value)
               JOIN master_1_l3_tab2 g
                  ON (g.pk_value = a.pk_value)
               JOIN master_1_l2_tab3 h
                  ON (h.pk_value = a.pk_value)
               JOIN master_1_l3_tab3 i
                  ON (i.pk_value = a.pk_value)
               JOIN master_2_l2_tab1 j
                  ON (j.pk_value = b.pk_value)
               JOIN master_2_l3_tab1 k
```

```
               ON (k.pk_value = j.pk_value)
            JOIN master_2_l2_tab2 l
               ON (l.pk_value = b.pk_value)
            JOIN master_2_l3_tab2 m
               ON (m.pk_value = l.pk_value)
            JOIN master_2_l2_tab3 n
               ON (n.pk_value = b.pk_value)
            JOIN master_2_l3_tab3 o
               ON (o.pk_value = n.pk_value)
            JOIN master_3_l2_tab1 p
               ON (p.parent_link = c.pk_value)
            JOIN master_3_l3_tab1 q
               ON (q.parent_link = p.pk_value)
            JOIN master_3_l2_tab2 r
               ON (r.parent_link = c.pk_value)
            JOIN master_3_l3_tab2 s
               ON (s.parent_link = r.pk_value)
            JOIN master_3_l2_tab3 t
               ON (t.parent_link = c.pk_value)
            JOIN master_3_l3_tab3 u
               ON (u.parent_link = t.pk_value)
      WHERE     u.char_1 BETWEEN 'A' AND 'Z'
            AND (   u.char_2 BETWEEN 'A' AND 'Z'
                 OR u.char_2 BETWEEN '1' AND '9')
            AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
                             AND SYSDATE
            AND s.value_3 IN
                   ('92E0b1CBj0',
                    'L51lHeuaJo',
                    'fQaLo5msQY',
                    '1tffVC1iy9',
                    'LMc2250887',
                    '80TiO4F81M',
                    'rg68IGlfEp',
                    'A7B900WG85')
            AND J.DATE10 BETWEEN (SELECT MIN (date10)
                                    FROM master_2_l2_tab1)
                             AND SYSDATE
            AND K.DATE_2 BETWEEN (SELECT MAX (date_2)
                                    FROM master_2_l3_tab1)
                             AND SYSDATE - 99999
     UNION
     SELECT COUNT (*) counter
       FROM master_1
      WHERE     date_1 IN
                   (SELECT date_1
                      FROM master_1
                     WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                   'DD-MON-YYYY')
                                      AND SYSDATE)
            AND date_2 IN
```

```
                    (SELECT date_2
                       FROM master_1
                      WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                    'DD-MON-YYYY')
                                       AND SYSDATE)
              AND date_3 IN
                    (SELECT date_3
                       FROM master_1
                      WHERE date_3 BETWEEN TO_DATE ('01-JAN-1972',
                                                    'DD-MON-YYYY')
                                       AND SYSDATE)
              AND date_4 IN
                    (SELECT date_4
                       FROM master_1
                      WHERE date_4 BETWEEN TO_DATE ('01-JAN-1973',
                                                    'DD-MON-YYYY')
                                       AND SYSDATE)
              AND date_5 IN
                    (SELECT date_5
                       FROM master_1
                      WHERE date_5 BETWEEN TO_DATE ('01-JAN-1974',
                                                    'DD-MON-YYYY')
                                       AND SYSDATE)
              AND char_1 IN (SELECT char_1
                               FROM master_1
                              WHERE char_1 BETWEEN 'A' AND 'Z')
              AND char_2 IN (SELECT char_2
                               FROM master_1
                              WHERE char_2 BETWEEN 'A' AND 'Z')
              AND char_3 IN (SELECT char_3
                               FROM master_1
                              WHERE char_3 BETWEEN 'A' AND 'Z')
              AND (value_1 LIKE 'a%' OR value_1 LIKE 'b%')
              AND (value_2 LIKE 'A%' OR value_2 LIKE 'B%'))
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_ALL_JOIN_2*/
SUM(counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
```

```
                ON (B.PK_VALUE = C.PK_VALUE)
             JOIN master_1_l2_tab1 d
                ON (d.pk_value = a.pk_value)
             JOIN master_1_l3_tab1 e
                ON (e.pk_value = a.pk_value)
             JOIN master_1_l2_tab2 f
                ON (f.pk_value = a.pk_value)
             JOIN master_1_l3_tab2 g
                ON (g.pk_value = a.pk_value)
             JOIN master_1_l2_tab3 h
                ON (h.pk_value = a.pk_value)
             JOIN master_1_l3_tab3 i
                ON (i.pk_value = a.pk_value)
             JOIN master_2_l2_tab1 j
                ON (j.pk_value = b.pk_value)
             JOIN master_2_l3_tab1 k
                ON (k.pk_value = j.pk_value)
             JOIN master_2_l2_tab2 l
                ON (l.pk_value = b.pk_value)
             JOIN master_2_l3_tab2 m
                ON (m.pk_value = l.pk_value)
             JOIN master_2_l2_tab3 n
                ON (n.pk_value = b.pk_value)
             JOIN master_2_l3_tab3 o
                ON (o.pk_value = n.pk_value)
             JOIN master_3_l2_tab1 p
                ON (p.parent_link = c.pk_value)
             JOIN master_3_l3_tab1 q
                ON (q.parent_link = p.pk_value)
             JOIN master_3_l2_tab2 r
                ON (r.parent_link = c.pk_value)
             JOIN master_3_l3_tab2 s
                ON (s.parent_link = r.pk_value)
             JOIN master_3_l2_tab3 t
                ON (t.parent_link = c.pk_value)
             JOIN master_3_l3_tab3 u
                ON (u.parent_link = t.pk_value)
      WHERE     u.char_1 BETWEEN 'A' AND 'Z'
             AND (   u.char_2 BETWEEN 'A' AND 'Z'
                  OR u.char_2 BETWEEN '1' AND '9')
             AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
                                  AND SYSDATE
             AND s.value_3 IN
                     ('92E0b1CBj0',
                      'L51lHeuaJo',
                      'fQaLo5msQY',
                      '1tffVC1iy9',
                      'LMc2250887',
                      '80TiO4F81M',
                      'rg68IGlfEp',
                      'A7B900WG85')
```

```
          AND J.DATE10 BETWEEN (SELECT MIN (date10)
                                  FROM master_2_l2_tab1)
                           AND SYSDATE
          AND K.DATE_2 BETWEEN (SELECT MAX (date_2)
                                  FROM master_2_l3_tab1)
                           AND SYSDATE - 99999
   UNION
   SELECT COUNT (*) counter
     FROM master_1
    WHERE     date_1 IN
                 (SELECT date_1
                    FROM master_1
                   WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                 'DD-MON-YYYY')
                                    AND SYSDATE)
          AND date_2 IN
                 (SELECT date_2
                    FROM master_1
                   WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                 'DD-MON-YYYY')
                                    AND SYSDATE)
          AND date_3 IN
                 (SELECT date_3
                    FROM master_1
                   WHERE date_3 BETWEEN TO_DATE ('01-JAN-1972',
                                                 'DD-MON-YYYY')
                                    AND SYSDATE)
          AND date_4 IN
                 (SELECT date_4
                    FROM master_1
                   WHERE date_4 BETWEEN TO_DATE ('01-JAN-1973',
                                                 'DD-MON-YYYY')
                                    AND SYSDATE)
          AND date_5 IN
                 (SELECT date_5
                    FROM master_1
                   WHERE date_5 BETWEEN TO_DATE ('01-JAN-1974',
                                                 'DD-MON-YYYY')
                                    AND SYSDATE)
          AND char_1 IN (SELECT char_1
                           FROM master_1
                          WHERE char_1 BETWEEN 'A' AND 'Z')
          AND char_2 IN (SELECT char_2
                           FROM master_1
                          WHERE char_2 BETWEEN 'A' AND 'Z')
          AND char_3 IN (SELECT char_3
                           FROM master_1
                          WHERE char_3 BETWEEN 'A' AND 'Z')
          AND (value_1 LIKE 'a%' OR value_1 LIKE 'b%')
          AND (value_2 LIKE 'A%' OR value_2 LIKE 'B%'))
/
exit
```

```
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_ALL_JOIN_2%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
--------------------------------------------
--INDEXES
M2_L2_T3_PK_IDX
M3_L3_T3_CHAR_1_IDX
M1_L2_T1_PK_IDX
MASTER_1_PK_IDX
MASTER_3_PK_IDX
MASTER_2_CHAR_1_IDX
MASTER_2_CHAR_3_IDX
MASTER_2_DATE_2_IDX
MASTER_2_VALUE_2_IDX
MASTER_2_VALUE_4_IDX
MASTER_2_VALUE_5_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT SUM (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
                  ON (B.PK_VALUE = C.PK_VALUE)
               JOIN master_1_l2_tab1 d
                  ON (d.pk_value = a.pk_value)
               JOIN master_1_l3_tab1 e
                  ON (e.pk_value = a.pk_value)
               JOIN master_1_l2_tab2 f
                  ON (f.pk_value = a.pk_value)
               JOIN master_1_l3_tab2 g
                  ON (g.pk_value = a.pk_value)
               JOIN master_1_l2_tab3 h
                  ON (h.pk_value = a.pk_value)
               JOIN master_1_l3_tab3 i
```

```
              ON (i.pk_value = a.pk_value)
           JOIN master_2_l2_tab1 j
              ON (j.pk_value = b.pk_value)
           JOIN master_2_l3_tab1 k
              ON (k.pk_value = j.pk_value)
           JOIN master_2_l2_tab2 l
              ON (l.pk_value = b.pk_value)
           JOIN master_2_l3_tab2 m
              ON (m.pk_value = l.pk_value)
           JOIN master_2_l2_tab3 n
              ON (n.pk_value = b.pk_value)
           JOIN master_2_l3_tab3 o
              ON (o.pk_value = n.pk_value)
           JOIN master_3_l2_tab1 p
              ON (p.parent_link = c.pk_value)
           JOIN master_3_l3_tab1 q
              ON (q.parent_link = p.pk_value)
           JOIN master_3_l2_tab2 r
              ON (r.parent_link = c.pk_value)
           JOIN master_3_l3_tab2 s
              ON (s.parent_link = r.pk_value)
           JOIN master_3_l2_tab3 t
              ON (t.parent_link = c.pk_value)
           JOIN master_3_l3_tab3 u
              ON (u.parent_link = t.pk_value)
     WHERE     u.char_1 BETWEEN 'A' AND 'Z'
           AND (   u.char_2 BETWEEN 'A' AND 'Z'
                OR u.char_2 BETWEEN '1' AND '9')
           AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
                                 AND SYSDATE
    UNION
    SELECT COUNT (*) counter
      FROM master_2
     WHERE     value_1 IN (SELECT value_1
                             FROM master_2
                            WHERE value_1 LIKE '%A%')
           AND value_2 IN (SELECT value_2
                             FROM master_2
                            WHERE value_2 LIKE '%B%')
           AND value_3 IN (SELECT value_3
                             FROM master_2
                            WHERE value_3 LIKE '%C%')
           AND (   value_4 LIKE 'A%'
                OR value_4 LIKE 'B%'
                OR value_4 LIKE 'C%'
                OR value_4 LIKE 'D%'
                OR value_4 LIKE 'E%')
           AND (   value_5 LIKE '0%'
                OR value_5 LIKE 'b%'
                OR value_5 LIKE '0%'
                OR value_5 LIKE '7%'
```

```
                    OR value_5 LIKE 'Z%')
               AND (char_1 BETWEEN 'a' AND 'z')
               AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
               AND (    char_3 BETWEEN 'a' AND 'z'
                    OR char_3 BETWEEN 'A' AND 'Z'
                    OR char_3 BETWEEN '0' AND '9')
               AND date_1 IN
                      (SELECT date_1
                         FROM master_2
                        WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                      'DD-MON-YYYY')
                                         AND SYSDATE)
               AND date_2 IN
                      (SELECT date_2
                         FROM master_2
                        WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                      'DD-MON-YYYY')
                                         AND SYSDATE))
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_ALL_JOIN_3*/
SUM (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
                  ON (B.PK_VALUE = C.PK_VALUE)
               JOIN master_1_l2_tab1 d
                  ON (d.pk_value = a.pk_value)
               JOIN master_1_l3_tab1 e
                  ON (e.pk_value = a.pk_value)
               JOIN master_1_l2_tab2 f
                  ON (f.pk_value = a.pk_value)
               JOIN master_1_l3_tab2 g
                  ON (g.pk_value = a.pk_value)
               JOIN master_1_l2_tab3 h
                  ON (h.pk_value = a.pk_value)
               JOIN master_1_l3_tab3 i
                  ON (i.pk_value = a.pk_value)
               JOIN master_2_l2_tab1 j
                  ON (j.pk_value = b.pk_value)
```

```
            JOIN master_2_l3_tab1 k
               ON (k.pk_value = j.pk_value)
            JOIN master_2_l2_tab2 l
               ON (l.pk_value = b.pk_value)
            JOIN master_2_l3_tab2 m
               ON (m.pk_value = l.pk_value)
            JOIN master_2_l2_tab3 n
               ON (n.pk_value = b.pk_value)
            JOIN master_2_l3_tab3 o
               ON (o.pk_value = n.pk_value)
            JOIN master_3_l2_tab1 p
               ON (p.parent_link = c.pk_value)
            JOIN master_3_l3_tab1 q
               ON (q.parent_link = p.pk_value)
            JOIN master_3_l2_tab2 r
               ON (r.parent_link = c.pk_value)
            JOIN master_3_l3_tab2 s
               ON (s.parent_link = r.pk_value)
            JOIN master_3_l2_tab3 t
               ON (t.parent_link = c.pk_value)
            JOIN master_3_l3_tab3 u
               ON (u.parent_link = t.pk_value)
      WHERE     u.char_1 BETWEEN 'A' AND 'Z'
            AND (   u.char_2 BETWEEN 'A' AND 'Z'
                 OR u.char_2 BETWEEN '1' AND '9')
            AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
                                     AND SYSDATE
        UNION
        SELECT COUNT (*) counter
          FROM master_2
         WHERE     value_1 IN (SELECT value_1
                                  FROM master_2
                                 WHERE value_1 LIKE '%A%')
               AND value_2 IN (SELECT value_2
                                  FROM master_2
                                 WHERE value_2 LIKE '%B%')
               AND value_3 IN (SELECT value_3
                                  FROM master_2
                                 WHERE value_3 LIKE '%C%')
               AND (   value_4 LIKE 'A%'
                    OR value_4 LIKE 'B%'
                    OR value_4 LIKE 'C%'
                    OR value_4 LIKE 'D%'
                    OR value_4 LIKE 'E%')
               AND (   value_5 LIKE '0%'
                    OR value_5 LIKE 'b%'
                    OR value_5 LIKE '0%'
                    OR value_5 LIKE '7%'
                    OR value_5 LIKE 'Z%')
               AND (char_1 BETWEEN 'a' AND 'z')
```

```
               AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
               AND (   char_3 BETWEEN 'a' AND 'z'
                    OR char_3 BETWEEN 'A' AND 'Z'
                    OR char_3 BETWEEN '0' AND '9')
               AND date_1 IN
                      (SELECT date_1
                         FROM master_2
                        WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                      'DD-MON-YYYY')
                                         AND SYSDATE)
               AND date_2 IN
                      (SELECT date_2
                         FROM master_2
                        WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                      'DD-MON-YYYY')
                                         AND SYSDATE))
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_ALL_JOIN_3%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
--------------------------------------------
--INDEXES
M3_L3_T3_CHAR_1_IDX
M3_L3_T3_CHAR_2_IDX
M1_L2_T1_PK_IDX
M3_L3_T3_VALUE_1_IDX
MASTER_1_PK_IDX
MASTER_2_PK_IDX
MASTER_3_PK_IDX
M2_L2_T1_DATE_10_IDX
M2_L3_T1_DATE_1_IDX
MASTER_1_CHAR_1_IDX
MASTER_1_DATE_1_IDX
MASTER_1_VALUE_1_IDX


--GET_COST
```

```
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT SUM (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
                  ON (B.PK_VALUE = C.PK_VALUE)
               JOIN master_1_l2_tab1 d
                  ON (d.pk_value = a.pk_value)
               JOIN master_1_l3_tab1 e
                  ON (e.pk_value = a.pk_value)
               JOIN master_1_l2_tab2 f
                  ON (f.pk_value = a.pk_value)
               JOIN master_1_l3_tab2 g
                  ON (g.pk_value = a.pk_value)
               JOIN master_1_l2_tab3 h
                  ON (h.pk_value = a.pk_value)
               JOIN master_1_l3_tab3 i
                  ON (i.pk_value = a.pk_value)
               JOIN master_2_l2_tab1 j
                  ON (j.pk_value = b.pk_value)
               JOIN master_2_l3_tab1 k
                  ON (k.pk_value = j.pk_value)
               JOIN master_2_l2_tab2 l
                  ON (l.pk_value = b.pk_value)
               JOIN master_2_l3_tab2 m
                  ON (m.pk_value = l.pk_value)
               JOIN master_2_l2_tab3 n
                  ON (n.pk_value = b.pk_value)
               JOIN master_2_l3_tab3 o
                  ON (o.pk_value = n.pk_value)
               JOIN master_3_l2_tab1 p
                  ON (p.parent_link = c.pk_value)
               JOIN master_3_l3_tab1 q
                  ON (q.parent_link = p.pk_value)
               JOIN master_3_l2_tab2 r
                  ON (r.parent_link = c.pk_value)
               JOIN master_3_l3_tab2 s
                  ON (s.parent_link = r.pk_value)
               JOIN master_3_l2_tab3 t
                  ON (t.parent_link = c.pk_value)
               JOIN master_3_l3_tab3 u
                  ON (u.parent_link = t.pk_value)
         WHERE     u.char_1 BETWEEN 'A' AND 'Z'
               AND (   u.char_2 BETWEEN 'A' AND 'Z'
                    OR u.char_2 BETWEEN '1' AND '9')
               AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
```

```
                                             AND SYSDATE
        UNION
        SELECT COUNT (*) counter
          FROM master_3
         WHERE     num_1 BETWEEN 1 AND 98768000
               AND num_2 BETWEEN 28 AND 291381400
               AND num_3 >= (SELECT MIN (num_3) FROM master_3)
               AND num_4 <= (SELECT AVG (num_4) FROM master_3)
               AND num_5 <= (SELECT MAX (num_5) FROM master_3)
               AND (char_1 BETWEEN 'a' AND 'z')
               AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
               AND (   char_3 BETWEEN 'a' AND 'z'
                    OR char_3 BETWEEN 'A' AND 'Z'
                    OR char_3 BETWEEN '0' AND '9')
               AND value_1 IN
                      (SELECT value_1
                         FROM master_3
                        WHERE value_1 LIKE '%A%' OR value_1 LIKE
'%B%')
               AND value_2 IN (SELECT value_2
                                 FROM master_3
                                WHERE value_2 LIKE '%C%'))
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_ALL_JOIN_4*/
SUM (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
                  ON (B.PK_VALUE = C.PK_VALUE)
               JOIN master_1_l2_tab1 d
                  ON (d.pk_value = a.pk_value)
               JOIN master_1_l3_tab1 e
                  ON (e.pk_value = a.pk_value)
               JOIN master_1_l2_tab2 f
                  ON (f.pk_value = a.pk_value)
               JOIN master_1_l3_tab2 g
                  ON (g.pk_value = a.pk_value)
               JOIN master_1_l2_tab3 h
                  ON (h.pk_value = a.pk_value)
```

```
              JOIN master_1_l3_tab3 i
                 ON (i.pk_value = a.pk_value)
              JOIN master_2_l2_tab1 j
                 ON (j.pk_value = b.pk_value)
              JOIN master_2_l3_tab1 k
                 ON (k.pk_value = j.pk_value)
              JOIN master_2_l2_tab2 l
                 ON (l.pk_value = b.pk_value)
              JOIN master_2_l3_tab2 m
                 ON (m.pk_value = l.pk_value)
              JOIN master_2_l2_tab3 n
                 ON (n.pk_value = b.pk_value)
              JOIN master_2_l3_tab3 o
                 ON (o.pk_value = n.pk_value)
              JOIN master_3_l2_tab1 p
                 ON (p.parent_link = c.pk_value)
              JOIN master_3_l3_tab1 q
                 ON (q.parent_link = p.pk_value)
              JOIN master_3_l2_tab2 r
                 ON (r.parent_link = c.pk_value)
              JOIN master_3_l3_tab2 s
                 ON (s.parent_link = r.pk_value)
              JOIN master_3_l2_tab3 t
                 ON (t.parent_link = c.pk_value)
              JOIN master_3_l3_tab3 u
                 ON (u.parent_link = t.pk_value)
        WHERE     u.char_1 BETWEEN 'A' AND 'Z'
              AND (   u.char_2 BETWEEN 'A' AND 'Z'
                   OR u.char_2 BETWEEN '1' AND '9')
              AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
                                   AND SYSDATE
       UNION
       SELECT COUNT (*) counter
         FROM master_3
        WHERE     num_1 BETWEEN 1 AND 98768000
              AND num_2 BETWEEN 28 AND 291381400
              AND num_3 >= (SELECT MIN (num_3) FROM master_3)
              AND num_4 <= (SELECT AVG (num_4) FROM master_3)
              AND num_5 <= (SELECT MAX (num_5) FROM master_3)
              AND (char_1 BETWEEN 'a' AND 'z')
              AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
              AND (   char_3 BETWEEN 'a' AND 'z'
                   OR char_3 BETWEEN 'A' AND 'Z'
                   OR char_3 BETWEEN '0' AND '9')
              AND value_1 IN
                     (SELECT value_1
                        FROM master_3
                       WHERE value_1 LIKE '%A%' OR value_1 LIKE
'%B%')
              AND value_2 IN (SELECT value_2
```

```
                                FROM master_3
                               WHERE value_2 LIKE '%C%'))
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_ALL_JOIN_4%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
M1_L3_T1_PK_IDX
M1_L3_T2_PK_IDX
M1_L3_T3_PK_IDX
M2_L2_T1_PK_IDX
M2_L2_T3_PK_IDX
M2_L2_TAB2_PK_IDX
M2_L3_T1_PK_IDX
M2_L3_T2_PK_IDX
M3_L2_T3_PK_IDX
MASTER_1_PK_IDX
MASTER_2_PK_IDX
MASTER_3_VALUE_1_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
select counter
from (SELECT COUNT (*) counter
  FROM master_3
 WHERE     num_1 BETWEEN 1 AND 98768000
       AND num_2 BETWEEN 28 AND 291381400
       AND num_3 >= (SELECT MIN (num_3) FROM master_3)
       AND num_4 <= (SELECT AVG (num_4) FROM master_3)
       AND num_5 <= (SELECT MAX (num_5) FROM master_3)
       AND (char_1 BETWEEN 'a' AND 'z')
       AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A' AND 'Z')
       AND (    char_3 BETWEEN 'a' AND 'z'
            OR char_3 BETWEEN 'A' AND 'Z'
            OR char_3 BETWEEN '0' AND '9')
```

```
       AND value_1 IN (SELECT value_1
                         FROM master_3
                        WHERE value_1 LIKE '%A%' OR value_1 LIKE
'%B%')
       AND value_2 IN (SELECT value_2
                         FROM master_3
                        WHERE value_2 LIKE '%C%')
UNION
SELECT SUM (counter) counter
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
                  ON (B.PK_VALUE = C.PK_VALUE)
               JOIN master_1_l2_tab1 d
                  ON (d.pk_value = a.pk_value)
               JOIN master_1_l3_tab1 e
                  ON (e.pk_value = a.pk_value)
               JOIN master_1_l2_tab2 f
                  ON (f.pk_value = a.pk_value)
               JOIN master_1_l3_tab2 g
                  ON (g.pk_value = a.pk_value)
               JOIN master_1_l2_tab3 h
                  ON (h.pk_value = a.pk_value)
               JOIN master_1_l3_tab3 i
                  ON (i.pk_value = a.pk_value)
               JOIN master_2_l2_tab1 j
                  ON (j.pk_value = b.pk_value)
               JOIN master_2_l3_tab1 k
                  ON (k.pk_value = j.pk_value)
               JOIN master_2_l2_tab2 l
                  ON (l.pk_value = b.pk_value)
               JOIN master_2_l3_tab2 m
                  ON (m.pk_value = l.pk_value)
               JOIN master_2_l2_tab3 n
                  ON (n.pk_value = b.pk_value)
               JOIN master_2_l3_tab3 o
                  ON (o.pk_value = n.pk_value)
               JOIN master_3_l2_tab1 p
                  ON (p.parent_link = c.pk_value)
               JOIN master_3_l3_tab1 q
                  ON (q.parent_link = p.pk_value)
               JOIN master_3_l2_tab2 r
                  ON (r.parent_link = c.pk_value)
               JOIN master_3_l3_tab2 s
                  ON (s.parent_link = r.pk_value)
               JOIN master_3_l2_tab3 t
                  ON (t.parent_link = c.pk_value)
               JOIN master_3_l3_tab3 u
                  ON (u.parent_link = t.pk_value)
         WHERE     u.char_1 BETWEEN 'A' AND 'Z'
```

```
              AND (   u.char_2 BETWEEN 'A' AND 'Z'
                   OR u.char_2 BETWEEN '1' AND '9')
              AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
                                 AND SYSDATE
        UNION
        SELECT COUNT (*) counter
          FROM master_2
         WHERE     value_1 IN (SELECT value_1
                                 FROM master_2
                                WHERE value_1 LIKE '%A%')
              AND value_2 IN (SELECT value_2
                                 FROM master_2
                                WHERE value_2 LIKE '%B%')
              AND value_3 IN (SELECT value_3
                                 FROM master_2
                                WHERE value_3 LIKE '%C%')
              AND (   value_4 LIKE 'A%'
                   OR value_4 LIKE 'B%'
                   OR value_4 LIKE 'C%'
                   OR value_4 LIKE 'D%'
                   OR value_4 LIKE 'E%')
              AND (   value_5 LIKE '0%'
                   OR value_5 LIKE 'b%'
                   OR value_5 LIKE '0%'
                   OR value_5 LIKE '7%'
                   OR value_5 LIKE 'Z%')
              AND (char_1 BETWEEN 'a' AND 'z')
              AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
              AND (   char_3 BETWEEN 'a' AND 'z'
                   OR char_3 BETWEEN 'A' AND 'Z'
                   OR char_3 BETWEEN '0' AND '9')
              AND date_1 IN
                     (SELECT date_1
                        FROM master_2
                       WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                     'DD-MON-YYYY')
                                         AND SYSDATE)
              AND date_2 IN
                     (SELECT date_2
                        FROM master_2
                       WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                     'DD-MON-YYYY')
                                         AND SYSDATE)))

/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
```

```
exit
/

--TIME_COST
set timing on
select /*+ MASTER_ALL_JOIN_5*/
counter
from (SELECT COUNT (*) counter
  FROM master_3
 WHERE     num_1 BETWEEN 1 AND 98768000
       AND num_2 BETWEEN 28 AND 291381400
       AND num_3 >= (SELECT MIN (num_3) FROM master_3)
       AND num_4 <= (SELECT AVG (num_4) FROM master_3)
       AND num_5 <= (SELECT MAX (num_5) FROM master_3)
       AND (char_1 BETWEEN 'a' AND 'z')
       AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A' AND 'Z')
       AND (   char_3 BETWEEN 'a' AND 'z'
            OR char_3 BETWEEN 'A' AND 'Z'
            OR char_3 BETWEEN '0' AND '9')
       AND value_1 IN (SELECT value_1
                         FROM master_3
                        WHERE value_1 LIKE '%A%' OR value_1 LIKE
'%B%')
       AND value_2 IN (SELECT value_2
                         FROM master_3
                        WHERE value_2 LIKE '%C%')
UNION
SELECT SUM (counter) counter
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
                  ON (B.PK_VALUE = C.PK_VALUE)
               JOIN master_1_l2_tab1 d
                  ON (d.pk_value = a.pk_value)
               JOIN master_1_l3_tab1 e
                  ON (e.pk_value = a.pk_value)
               JOIN master_1_l2_tab2 f
                  ON (f.pk_value = a.pk_value)
               JOIN master_1_l3_tab2 g
                  ON (g.pk_value = a.pk_value)
               JOIN master_1_l2_tab3 h
                  ON (h.pk_value = a.pk_value)
               JOIN master_1_l3_tab3 i
                  ON (i.pk_value = a.pk_value)
               JOIN master_2_l2_tab1 j
                  ON (j.pk_value = b.pk_value)
               JOIN master_2_l3_tab1 k
                  ON (k.pk_value = j.pk_value)
               JOIN master_2_l2_tab2 l
                  ON (l.pk_value = b.pk_value)
```

```
              JOIN master_2_l3_tab2 m
                 ON (m.pk_value = l.pk_value)
              JOIN master_2_l2_tab3 n
                 ON (n.pk_value = b.pk_value)
              JOIN master_2_l3_tab3 o
                 ON (o.pk_value = n.pk_value)
              JOIN master_3_l2_tab1 p
                 ON (p.parent_link = c.pk_value)
              JOIN master_3_l3_tab1 q
                 ON (q.parent_link = p.pk_value)
              JOIN master_3_l2_tab2 r
                 ON (r.parent_link = c.pk_value)
              JOIN master_3_l3_tab2 s
                 ON (s.parent_link = r.pk_value)
              JOIN master_3_l2_tab3 t
                 ON (t.parent_link = c.pk_value)
              JOIN master_3_l3_tab3 u
                 ON (u.parent_link = t.pk_value)
        WHERE     u.char_1 BETWEEN 'A' AND 'Z'
              AND (   u.char_2 BETWEEN 'A' AND 'Z'
                   OR u.char_2 BETWEEN '1' AND '9')
              AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
                               AND SYSDATE
        UNION
        SELECT COUNT (*) counter
          FROM master_2
         WHERE     value_1 IN (SELECT value_1
                                 FROM master_2
                                WHERE value_1 LIKE '%A%')
              AND value_2 IN (SELECT value_2
                                 FROM master_2
                                WHERE value_2 LIKE '%B%')
              AND value_3 IN (SELECT value_3
                                 FROM master_2
                                WHERE value_3 LIKE '%C%')
              AND (   value_4 LIKE 'A%'
                   OR value_4 LIKE 'B%'
                   OR value_4 LIKE 'C%'
                   OR value_4 LIKE 'D%'
                   OR value_4 LIKE 'E%')
              AND (   value_5 LIKE '0%'
                   OR value_5 LIKE 'b%'
                   OR value_5 LIKE '0%'
                   OR value_5 LIKE '7%'
                   OR value_5 LIKE 'Z%')
              AND (char_1 BETWEEN 'a' AND 'z')
              AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
              AND (   char_3 BETWEEN 'a' AND 'z'
                   OR char_3 BETWEEN 'A' AND 'Z'
                   OR char_3 BETWEEN '0' AND '9')
```

```
              AND date_1 IN
                     (SELECT date_1
                        FROM master_2
                       WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                     'DD-MON-YYYY')
                                        AND SYSDATE)
              AND date_2 IN
                     (SELECT date_2
                        FROM master_2
                       WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                     'DD-MON-YYYY')
                                        AND SYSDATE))

/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_ALL_JOIN_5%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
--------------------------------------------
--INDEXES
M1_L2_T1_PK_IDX
M1_L2_T2_PK_IDX
M1_L2_T3_PK_IDX
M1_L3_T1_PK_IDX
M1_L3_T2_PK_IDX
M1_L3_T3_PK_IDX
M2_L2_T1_PK_IDX
M2_L2_T3_PK_IDX
M2_L2_TAB2_PK_IDX
M2_L3_T1_PK_IDX
M2_L3_T2_PK_IDX
M3_L2_T3_PK_IDX


--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
select counter
```

```
from (SELECT COUNT (*) counter
  FROM master_3
 WHERE     num_1 BETWEEN 1 AND 98768000
       AND num_2 BETWEEN 28 AND 291381400
       AND num_3 >= (SELECT MIN (num_3) FROM master_3)
       AND num_4 <= (SELECT AVG (num_4) FROM master_3)
       AND num_5 <= (SELECT MAX (num_5) FROM master_3)
       AND (char_1 BETWEEN 'a' AND 'z')
       AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A' AND 'Z')
       AND (    char_3 BETWEEN 'a' AND 'z'
            OR char_3 BETWEEN 'A' AND 'Z'
            OR char_3 BETWEEN '0' AND '9')
       AND value_1 IN (SELECT value_1
                         FROM master_3
                        WHERE value_1 LIKE '%A%' OR value_1 LIKE
'%B%')
       AND value_2 IN (SELECT value_2
                         FROM master_3
                        WHERE value_2 LIKE '%C%')
UNION
SELECT SUM (counter) counter
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
                  ON (B.PK_VALUE = C.PK_VALUE)
               JOIN master_1_l2_tab1 d
                  ON (d.pk_value = a.pk_value)
               JOIN master_1_l3_tab1 e
                  ON (e.pk_value = a.pk_value)
               JOIN master_1_l2_tab2 f
                  ON (f.pk_value = a.pk_value)
               JOIN master_1_l3_tab2 g
                  ON (g.pk_value = a.pk_value)
               JOIN master_1_l2_tab3 h
                  ON (h.pk_value = a.pk_value)
               JOIN master_1_l3_tab3 i
                  ON (i.pk_value = a.pk_value)
               JOIN master_2_l2_tab1 j
                  ON (j.pk_value = b.pk_value)
               JOIN master_2_l3_tab1 k
                  ON (k.pk_value = j.pk_value)
               JOIN master_2_l2_tab2 l
                  ON (l.pk_value = b.pk_value)
               JOIN master_2_l3_tab2 m
                  ON (m.pk_value = l.pk_value)
               JOIN master_2_l2_tab3 n
                  ON (n.pk_value = b.pk_value)
               JOIN master_2_l3_tab3 o
                  ON (o.pk_value = n.pk_value)
               JOIN master_3_l2_tab1 p
```

```
                 ON (p.parent_link = c.pk_value)
              JOIN master_3_l3_tab1 q
                 ON (q.parent_link = p.pk_value)
              JOIN master_3_l2_tab2 r
                 ON (r.parent_link = c.pk_value)
              JOIN master_3_l3_tab2 s
                 ON (s.parent_link = r.pk_value)
              JOIN master_3_l2_tab3 t
                 ON (t.parent_link = c.pk_value)
              JOIN master_3_l3_tab3 u
                 ON (u.parent_link = t.pk_value)
        WHERE     u.char_1 BETWEEN 'A' AND 'Z'
              AND (   u.char_2 BETWEEN 'A' AND 'Z'
                   OR u.char_2 BETWEEN '1' AND '9')
              AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
                                AND SYSDATE
       UNION
       SELECT COUNT (*) counter
         FROM master_2
        WHERE     value_1 IN (SELECT value_1
                                 FROM master_2
                                WHERE value_1 LIKE '%A%')
              AND value_2 IN (SELECT value_2
                                 FROM master_2
                                WHERE value_2 LIKE '%B%')
              AND value_3 IN (SELECT value_3
                                 FROM master_2
                                WHERE value_3 LIKE '%C%')
              AND (   value_4 LIKE 'A%'
                   OR value_4 LIKE 'B%'
                   OR value_4 LIKE 'C%'
                   OR value_4 LIKE 'D%'
                   OR value_4 LIKE 'E%')
              AND (   value_5 LIKE '0%'
                   OR value_5 LIKE 'b%'
                   OR value_5 LIKE '0%'
                   OR value_5 LIKE '7%'
                   OR value_5 LIKE 'Z%')
              AND (char_1 BETWEEN 'a' AND 'z')
              AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
              AND (   char_3 BETWEEN 'a' AND 'z'
                   OR char_3 BETWEEN 'A' AND 'Z'
                   OR char_3 BETWEEN '0' AND '9')
              AND date_1 IN
                     (SELECT date_1
                        FROM master_2
                       WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                     'DD-MON-YYYY')
                                        AND SYSDATE)
              AND date_2 IN
```

```
                      (SELECT date_2
                         FROM master_2
                        WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                      'DD-MON-YYYY')
                                         AND SYSDATE)))

/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_ALL_JOIN_6*/
counter
from (SELECT COUNT (*) counter
  FROM master_3
 WHERE     num_1 BETWEEN 1 AND 98768000
       AND num_2 BETWEEN 28 AND 291381400
       AND num_3 >= (SELECT MIN (num_3) FROM master_3)
       AND num_4 <= (SELECT AVG (num_4) FROM master_3)
       AND num_5 <= (SELECT MAX (num_5) FROM master_3)
       AND (char_1 BETWEEN 'a' AND 'z')
       AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A' AND 'Z')
       AND (   char_3 BETWEEN 'a' AND 'z'
            OR char_3 BETWEEN 'A' AND 'Z'
            OR char_3 BETWEEN '0' AND '9')
       AND value_1 IN (SELECT value_1
                         FROM master_3
                        WHERE value_1 LIKE '%A%' OR value_1 LIKE
'%B%')
       AND value_2 IN (SELECT value_2
                         FROM master_3
                        WHERE value_2 LIKE '%C%')
UNION
SELECT SUM (counter) counter
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
                  ON (B.PK_VALUE = C.PK_VALUE)
               JOIN master_1_l2_tab1 d
                  ON (d.pk_value = a.pk_value)
               JOIN master_1_l3_tab1 e
                  ON (e.pk_value = a.pk_value)
               JOIN master_1_l2_tab2 f
                  ON (f.pk_value = a.pk_value)
```

```
              JOIN master_1_l3_tab2 g
                 ON (g.pk_value = a.pk_value)
              JOIN master_1_l2_tab3 h
                 ON (h.pk_value = a.pk_value)
              JOIN master_1_l3_tab3 i
                 ON (i.pk_value = a.pk_value)
              JOIN master_2_l2_tab1 j
                 ON (j.pk_value = b.pk_value)
              JOIN master_2_l3_tab1 k
                 ON (k.pk_value = j.pk_value)
              JOIN master_2_l2_tab2 l
                 ON (l.pk_value = b.pk_value)
              JOIN master_2_l3_tab2 m
                 ON (m.pk_value = l.pk_value)
              JOIN master_2_l2_tab3 n
                 ON (n.pk_value = b.pk_value)
              JOIN master_2_l3_tab3 o
                 ON (o.pk_value = n.pk_value)
              JOIN master_3_l2_tab1 p
                 ON (p.parent_link = c.pk_value)
              JOIN master_3_l3_tab1 q
                 ON (q.parent_link = p.pk_value)
              JOIN master_3_l2_tab2 r
                 ON (r.parent_link = c.pk_value)
              JOIN master_3_l3_tab2 s
                 ON (s.parent_link = r.pk_value)
              JOIN master_3_l2_tab3 t
                 ON (t.parent_link = c.pk_value)
              JOIN master_3_l3_tab3 u
                 ON (u.parent_link = t.pk_value)
        WHERE     u.char_1 BETWEEN 'A' AND 'Z'
              AND (   u.char_2 BETWEEN 'A' AND 'Z'
                   OR u.char_2 BETWEEN '1' AND '9')
              AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
                                   AND SYSDATE
       UNION
       SELECT COUNT (*) counter
         FROM master_2
        WHERE     value_1 IN (SELECT value_1
                                FROM master_2
                               WHERE value_1 LIKE '%A%')
              AND value_2 IN (SELECT value_2
                                FROM master_2
                               WHERE value_2 LIKE '%B%')
              AND value_3 IN (SELECT value_3
                                FROM master_2
                               WHERE value_3 LIKE '%C%')
              AND (   value_4 LIKE 'A%'
                   OR value_4 LIKE 'B%'
                   OR value_4 LIKE 'C%'
                   OR value_4 LIKE 'D%'
```

```
                     OR value_4 LIKE 'E%')
               AND (    value_5 LIKE '0%'
                     OR value_5 LIKE 'b%'
                     OR value_5 LIKE '0%'
                     OR value_5 LIKE '7%'
                     OR value_5 LIKE 'Z%')
               AND (char_1 BETWEEN 'a' AND 'z')
               AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
               AND (    char_3 BETWEEN 'a' AND 'z'
                     OR char_3 BETWEEN 'A' AND 'Z'
                     OR char_3 BETWEEN '0' AND '9')
               AND date_1 IN
                       (SELECT date_1
                          FROM master_2
                         WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                       'DD-MON-YYYY')
                                          AND SYSDATE)
               AND date_2 IN
                       (SELECT date_2
                          FROM master_2
                         WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                       'DD-MON-YYYY')
                                          AND SYSDATE)))

/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_ALL_JOIN_6%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
------------------------------------------
--INDEXES
M1_L2_T3_PK_IDX
M1_L3_T1_PK_IDX
M1_L3_T2_PK_IDX
M1_L3_T3_PK_IDX
M2_L2_T1_PK_IDX
M2_L2_T3_PK_IDX
M2_L2_TAB2_PK_IDX
M2_L3_T1_PK_IDX
```

```
M2_L3_T2_PK_IDX
M3_L2_T3_PK_IDX
MASTER_1_PK_IDX
MASTER_2_PK_IDX


--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
select counter
from (SELECT COUNT (*) counter
  FROM master_3
 WHERE     num_1 BETWEEN 1 AND 98768000
       AND num_2 BETWEEN 28 AND 291381400
       AND num_3 >= (SELECT MIN (num_3) FROM master_3)
       AND num_4 <= (SELECT AVG (num_4) FROM master_3)
       AND num_5 <= (SELECT MAX (num_5) FROM master_3)
       AND (char_1 BETWEEN 'a' AND 'z')
       AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A' AND 'Z')
       AND (   char_3 BETWEEN 'a' AND 'z'
            OR char_3 BETWEEN 'A' AND 'Z'
            OR char_3 BETWEEN '0' AND '9')
       AND value_1 IN (SELECT value_1
                         FROM master_3
                        WHERE value_1 LIKE '%A%' OR value_1 LIKE
'%B%')
       AND value_2 IN (SELECT value_2
                         FROM master_3
                        WHERE value_2 LIKE '%C%')
UNION
SELECT SUM (counter) counter
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
                  ON (B.PK_VALUE = C.PK_VALUE)
               JOIN master_1_l2_tab1 d
                  ON (d.pk_value = a.pk_value)
               JOIN master_1_l3_tab1 e
                  ON (e.pk_value = a.pk_value)
               JOIN master_1_l2_tab2 f
                  ON (f.pk_value = a.pk_value)
               JOIN master_1_l3_tab2 g
                  ON (g.pk_value = a.pk_value)
               JOIN master_1_l2_tab3 h
                  ON (h.pk_value = a.pk_value)
               JOIN master_1_l3_tab3 i
                  ON (i.pk_value = a.pk_value)
               JOIN master_2_l2_tab1 j
```

```
              ON (j.pk_value = b.pk_value)
           JOIN master_2_l3_tab1 k
              ON (k.pk_value = j.pk_value)
           JOIN master_2_l2_tab2 l
              ON (l.pk_value = b.pk_value)
           JOIN master_2_l3_tab2 m
              ON (m.pk_value = l.pk_value)
           JOIN master_2_l2_tab3 n
              ON (n.pk_value = b.pk_value)
           JOIN master_2_l3_tab3 o
              ON (o.pk_value = n.pk_value)
           JOIN master_3_l2_tab1 p
              ON (p.parent_link = c.pk_value)
           JOIN master_3_l3_tab1 q
              ON (q.parent_link = p.pk_value)
           JOIN master_3_l2_tab2 r
              ON (r.parent_link = c.pk_value)
           JOIN master_3_l3_tab2 s
              ON (s.parent_link = r.pk_value)
           JOIN master_3_l2_tab3 t
              ON (t.parent_link = c.pk_value)
           JOIN master_3_l3_tab3 u
              ON (u.parent_link = t.pk_value)
     WHERE     u.char_1 BETWEEN 'A' AND 'Z'
           AND (   u.char_2 BETWEEN 'A' AND 'Z'
                OR u.char_2 BETWEEN '1' AND '9')
           AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
                            AND SYSDATE
    UNION
    SELECT COUNT (*) counter
      FROM master_2
     WHERE     value_1 IN (SELECT value_1
                              FROM master_2
                             WHERE value_1 LIKE '%A%')
           AND value_2 IN (SELECT value_2
                              FROM master_2
                             WHERE value_2 LIKE '%B%')
           AND value_3 IN (SELECT value_3
                              FROM master_2
                             WHERE value_3 LIKE '%C%')
           AND (   value_4 LIKE 'A%'
                OR value_4 LIKE 'B%'
                OR value_4 LIKE 'C%'
                OR value_4 LIKE 'D%'
                OR value_4 LIKE 'E%')
           AND (   value_5 LIKE '0%'
                OR value_5 LIKE 'b%'
                OR value_5 LIKE '0%'
                OR value_5 LIKE '7%'
                OR value_5 LIKE 'Z%')
           AND (char_1 BETWEEN 'a' AND 'z')
```

```
               AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
               AND (    char_3 BETWEEN 'a' AND 'z'
                    OR char_3 BETWEEN 'A' AND 'Z'
                    OR char_3 BETWEEN '0' AND '9')
               AND date_1 IN
                      (SELECT date_1
                         FROM master_2
                        WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                      'DD-MON-YYYY')
                                         AND SYSDATE)
               AND date_2 IN
                      (SELECT date_2
                         FROM master_2
                        WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                      'DD-MON-YYYY')
                                         AND SYSDATE)))

/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_ALL_JOIN_7*/
counter
from (SELECT COUNT (*) counter
  FROM master_3
 WHERE     num_1 BETWEEN 1 AND 98768000
       AND num_2 BETWEEN 28 AND 291381400
       AND num_3 >= (SELECT MIN (num_3) FROM master_3)
       AND num_4 <= (SELECT AVG (num_4) FROM master_3)
       AND num_5 <= (SELECT MAX (num_5) FROM master_3)
       AND (char_1 BETWEEN 'a' AND 'z')
       AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A' AND 'Z')
       AND (    char_3 BETWEEN 'a' AND 'z'
            OR char_3 BETWEEN 'A' AND 'Z'
            OR char_3 BETWEEN '0' AND '9')
       AND value_1 IN (SELECT value_1
                         FROM master_3
                        WHERE value_1 LIKE '%A%' OR value_1 LIKE
'%B%')
       AND value_2 IN (SELECT value_2
                         FROM master_3
                        WHERE value_2 LIKE '%C%')
UNION
SELECT SUM (counter) counter
```

```
  FROM (SELECT COUNT (*) counter
          FROM master_1 a
               JOIN master_2 b
                  ON (A.PK_VALUE = B.PK_VALUE)
               JOIN master_3 c
                  ON (B.PK_VALUE = C.PK_VALUE)
               JOIN master_1_l2_tab1 d
                  ON (d.pk_value = a.pk_value)
               JOIN master_1_l3_tab1 e
                  ON (e.pk_value = a.pk_value)
               JOIN master_1_l2_tab2 f
                  ON (f.pk_value = a.pk_value)
               JOIN master_1_l3_tab2 g
                  ON (g.pk_value = a.pk_value)
               JOIN master_1_l2_tab3 h
                  ON (h.pk_value = a.pk_value)
               JOIN master_1_l3_tab3 i
                  ON (i.pk_value = a.pk_value)
               JOIN master_2_l2_tab1 j
                  ON (j.pk_value = b.pk_value)
               JOIN master_2_l3_tab1 k
                  ON (k.pk_value = j.pk_value)
               JOIN master_2_l2_tab2 l
                  ON (l.pk_value = b.pk_value)
               JOIN master_2_l3_tab2 m
                  ON (m.pk_value = l.pk_value)
               JOIN master_2_l2_tab3 n
                  ON (n.pk_value = b.pk_value)
               JOIN master_2_l3_tab3 o
                  ON (o.pk_value = n.pk_value)
               JOIN master_3_l2_tab1 p
                  ON (p.parent_link = c.pk_value)
               JOIN master_3_l3_tab1 q
                  ON (q.parent_link = p.pk_value)
               JOIN master_3_l2_tab2 r
                  ON (r.parent_link = c.pk_value)
               JOIN master_3_l3_tab2 s
                  ON (s.parent_link = r.pk_value)
               JOIN master_3_l2_tab3 t
                  ON (t.parent_link = c.pk_value)
               JOIN master_3_l3_tab3 u
                  ON (u.parent_link = t.pk_value)
         WHERE     u.char_1 BETWEEN 'A' AND 'Z'
               AND (   u.char_2 BETWEEN 'A' AND 'Z'
                    OR u.char_2 BETWEEN '1' AND '9')
               AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-
YYYY')
                                    AND SYSDATE
        UNION
        SELECT COUNT (*) counter
          FROM master_2
         WHERE     value_1 IN (SELECT value_1
```

```
                                FROM master_2
                               WHERE value_1 LIKE '%A%')
               AND value_2 IN (SELECT value_2
                                FROM master_2
                               WHERE value_2 LIKE '%B%')
               AND value_3 IN (SELECT value_3
                                FROM master_2
                               WHERE value_3 LIKE '%C%')
               AND (    value_4 LIKE 'A%'
                     OR value_4 LIKE 'B%'
                     OR value_4 LIKE 'C%'
                     OR value_4 LIKE 'D%'
                     OR value_4 LIKE 'E%')
               AND (    value_5 LIKE '0%'
                     OR value_5 LIKE 'b%'
                     OR value_5 LIKE '0%'
                     OR value_5 LIKE '7%'
                     OR value_5 LIKE 'Z%')
               AND (char_1 BETWEEN 'a' AND 'z')
               AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
               AND (    char_3 BETWEEN 'a' AND 'z'
                     OR char_3 BETWEEN 'A' AND 'Z'
                     OR char_3 BETWEEN '0' AND '9')
               AND date_1 IN
                       (SELECT date_1
                          FROM master_2
                         WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                        'DD-MON-YYYY')
                                          AND SYSDATE)
               AND date_2 IN
                       (SELECT date_2
                          FROM master_2
                         WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                        'DD-MON-YYYY')
                                          AND SYSDATE)))

/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_ALL_JOIN_7%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
```

```
/
----------------------------------------------------------------------
------------------------------------------
--INDEXES
M2_L2_T3_PK_IDX
M3_L3_T3_CHAR_1_IDX
M1_L2_T1_PK_IDX
MASTER_1_PK_IDX
MASTER_3_PK_IDX
MASTER_2_CHAR_1_IDX
MASTER_2_CHAR_3_IDX
MASTER_2_DATE_2_IDX
MASTER_2_VALUE_2_IDX
MASTER_2_VALUE_4_IDX
MASTER_2_VALUE_5_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT a.pk_value,
        b.pk_value,
        c.pk_value,
        d.pk_value,
        e.pk_value,
        f.pk_value,
        g.pk_value,
        h.pk_value,
        i.pk_value,
        j.pk_value,
        k.pk_value,
        l.pk_value,
        m.pk_value,
        n.pk_value,
        o.pk_value,
        p.pk_value,
        q.pk_value,
        r.pk_value,
        s.pk_value,
        t.pk_value
   FROM master_1 a
        JOIN master_2 b
           ON (A.PK_VALUE = B.PK_VALUE)
        JOIN master_3 c
           ON (B.PK_VALUE = C.PK_VALUE)
        JOIN master_1_l2_tab1 d
           ON (d.pk_value = a.pk_value)
        JOIN master_1_l3_tab1 e
           ON (e.pk_value = a.pk_value)
        JOIN master_1_l2_tab2 f
           ON (f.pk_value = a.pk_value)
```

```
        JOIN master_1_l3_tab2 g
           ON (g.pk_value = a.pk_value)
        JOIN master_1_l2_tab3 h
           ON (h.pk_value = a.pk_value)
        JOIN master_1_l3_tab3 i
           ON (i.pk_value = a.pk_value)
        JOIN master_2_l2_tab1 j
           ON (j.pk_value = b.pk_value)
        JOIN master_2_l3_tab1 k
           ON (k.pk_value = j.pk_value)
        JOIN master_2_l2_tab2 l
           ON (l.pk_value = b.pk_value)
        JOIN master_2_l3_tab2 m
           ON (m.pk_value = l.pk_value)
        JOIN master_2_l2_tab3 n
           ON (n.pk_value = b.pk_value)
        JOIN master_2_l3_tab3 o
           ON (o.pk_value = n.pk_value)
        JOIN master_3_l2_tab1 p
           ON (p.parent_link = c.pk_value)
        JOIN master_3_l3_tab1 q
           ON (q.parent_link = p.pk_value)
        JOIN master_3_l2_tab2 r
           ON (r.parent_link = c.pk_value)
        JOIN master_3_l3_tab2 s
           ON (s.parent_link = r.pk_value)
        JOIN master_3_l2_tab3 t
           ON (t.parent_link = c.pk_value)
        JOIN master_3_l3_tab3 u
           ON (u.parent_link = t.pk_value)
  WHERE     u.char_1 BETWEEN 'A' AND 'Z'
        AND (u.char_2 BETWEEN 'A' AND 'Z' OR u.char_2 BETWEEN '1' AND
'9')
        AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-YYYY')
                         AND SYSDATE
 UNION
 SELECT NULL,
        pk_value,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
        NULL,
```

```
        NULL,
        NULL,
        NULL,
        NULL
   FROM master_2
  WHERE     value_1 IN (SELECT value_1
                          FROM master_2
                         WHERE value_1 LIKE '%A%')
        AND value_2 IN (SELECT value_2
                          FROM master_2
                         WHERE value_2 LIKE '%B%')
        AND value_3 IN (SELECT value_3
                          FROM master_2
                         WHERE value_3 LIKE '%C%')
        AND (   value_4 LIKE 'A%'
             OR value_4 LIKE 'B%'
             OR value_4 LIKE 'C%'
             OR value_4 LIKE 'D%'
             OR value_4 LIKE 'E%')
        AND (   value_5 LIKE '0%'
             OR value_5 LIKE 'b%'
             OR value_5 LIKE '0%'
             OR value_5 LIKE '7%'
             OR value_5 LIKE 'Z%')
        AND (char_1 BETWEEN 'a' AND 'z')
        AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A' AND 'Z')
        AND (   char_3 BETWEEN 'a' AND 'z'
             OR char_3 BETWEEN 'A' AND 'Z'
             OR char_3 BETWEEN '0' AND '9')
        AND date_1 IN
               (SELECT date_1
                  FROM master_2
                 WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970', 'DD-MON-
YYYY')
                                  AND SYSDATE)
        AND date_2 IN
               (SELECT date_2
                  FROM master_2
                 WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971', 'DD-MON-
YYYY')
                                  AND SYSDATE)
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
```

```
select /*+ MASTER_ALL_JOIN_8*/
        a.pk_value,
        b.pk_value,
        c.pk_value,
        d.pk_value,
        e.pk_value,
        f.pk_value,
        g.pk_value,
        h.pk_value,
        i.pk_value,
        j.pk_value,
        k.pk_value,
        l.pk_value,
        m.pk_value,
        n.pk_value,
        o.pk_value,
        p.pk_value,
        q.pk_value,
        r.pk_value,
        s.pk_value,
        t.pk_value
   FROM master_1 a
        JOIN master_2 b
           ON (A.PK_VALUE = B.PK_VALUE)
        JOIN master_3 c
           ON (B.PK_VALUE = C.PK_VALUE)
        JOIN master_1_l2_tab1 d
           ON (d.pk_value = a.pk_value)
        JOIN master_1_l3_tab1 e
           ON (e.pk_value = a.pk_value)
        JOIN master_1_l2_tab2 f
           ON (f.pk_value = a.pk_value)
        JOIN master_1_l3_tab2 g
           ON (g.pk_value = a.pk_value)
        JOIN master_1_l2_tab3 h
           ON (h.pk_value = a.pk_value)
        JOIN master_1_l3_tab3 i
           ON (i.pk_value = a.pk_value)
        JOIN master_2_l2_tab1 j
           ON (j.pk_value = b.pk_value)
        JOIN master_2_l3_tab1 k
           ON (k.pk_value = j.pk_value)
        JOIN master_2_l2_tab2 l
           ON (l.pk_value = b.pk_value)
        JOIN master_2_l3_tab2 m
           ON (m.pk_value = l.pk_value)
        JOIN master_2_l2_tab3 n
           ON (n.pk_value = b.pk_value)
        JOIN master_2_l3_tab3 o
           ON (o.pk_value = n.pk_value)
        JOIN master_3_l2_tab1 p
           ON (p.parent_link = c.pk_value)
```

```
       JOIN master_3_l3_tab1 q
          ON (q.parent_link = p.pk_value)
       JOIN master_3_l2_tab2 r
          ON (r.parent_link = c.pk_value)
       JOIN master_3_l3_tab2 s
          ON (s.parent_link = r.pk_value)
       JOIN master_3_l2_tab3 t
          ON (t.parent_link = c.pk_value)
       JOIN master_3_l3_tab3 u
          ON (u.parent_link = t.pk_value)
 WHERE     u.char_1 BETWEEN 'A' AND 'Z'
       AND (u.char_2 BETWEEN 'A' AND 'Z' OR u.char_2 BETWEEN '1' AND
'9')
       AND a.date_1 BETWEEN TO_DATE ('01-FEB-1983', 'DD-MON-YYYY')
                        AND SYSDATE
UNION
SELECT NULL,
       pk_value,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL,
       NULL
  FROM master_2
 WHERE     value_1 IN (SELECT value_1
                         FROM master_2
                        WHERE value_1 LIKE '%A%')
       AND value_2 IN (SELECT value_2
                         FROM master_2
                        WHERE value_2 LIKE '%B%')
       AND value_3 IN (SELECT value_3
                         FROM master_2
                        WHERE value_3 LIKE '%C%')
       AND (   value_4 LIKE 'A%'
            OR value_4 LIKE 'B%'
            OR value_4 LIKE 'C%'
            OR value_4 LIKE 'D%'
            OR value_4 LIKE 'E%')
       AND (   value_5 LIKE '0%'
```

```
              OR value_5 LIKE 'b%'
              OR value_5 LIKE '0%'
              OR value_5 LIKE '7%'
              OR value_5 LIKE 'Z%')
        AND (char_1 BETWEEN 'a' AND 'z')
        AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A' AND 'Z')
        AND (    char_3 BETWEEN 'a' AND 'z'
              OR char_3 BETWEEN 'A' AND 'Z'
              OR char_3 BETWEEN '0' AND '9')
        AND date_1 IN
               (SELECT date_1
                  FROM master_2
                 WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970', 'DD-MON-
YYYY')
                                  AND SYSDATE)
        AND date_2 IN
               (SELECT date_2
                  FROM master_2
                 WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971', 'DD-MON-
YYYY')
                                  AND SYSDATE)
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_ALL_JOIN_8%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
MASTER_2_CHAR_1_IDX
MASTER_2_CHAR_2_IDX
MASTER_2_CHAR_3_IDX
MASTER_2_DATE_1_IDX
MASTER_2_DATE_2_IDX
MASTER_2_VALUE_1_IDX
MASTER_2_VALUE_2_IDX
MASTER_2_VALUE_3_IDX
MASTER_2_VALUE_4_IDX
MASTER_2_VALUE_5_IDX

--GET_COST
```

```
delete from plan_table
/
explain plan
set statement_id = 'A' for
select count(*)
from master_2
where value_1 in (select value_1 from master_2 where value_1 like
'%A%')
  and value_2 in (select value_2 from master_2 where value_2 like
'%B%')
  and value_3 in (select value_3 from master_2 where value_3 like
'%C%')
  and (value_4 like 'A%' or
       value_4 like 'B%' or
       value_4 like 'C%' or
       value_4 like 'D%' or
       value_4 like 'E%')
  and (value_5 like '0%' or value_5 like 'b%' or value_5 like '0%' or
value_5 like '7%' or value_5 like 'Z%')
  and (char_1 between 'a' and 'z')
  and (char_2 between 'a' and 'z' or char_2 between 'A' and 'Z')
  and (char_3 between 'a' and 'z' or char_3 between 'A' and 'Z' or
char_3 between '0' and '9')
  and date_1 in (select date_1 from master_2 where date_1 between
to_date('01-JAN-1970','DD-MON-YYYY') and sysdate)
  and date_2 in (select date_2 from master_2 where date_2 between
to_date('01-JAN-1971','DD-MON-YYYY') and sysdate)
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/
--TIME_COST
set timing on
select /*+ MASTER_2_QUERY*/
count(*)
from master_2
where value_1 in (select value_1 from master_2 where value_1 like
'%A%')
  and value_2 in (select value_2 from master_2 where value_2 like
'%B%')
  and value_3 in (select value_3 from master_2 where value_3 like
'%C%')
  and (value_4 like 'A%' or
       value_4 like 'B%' or
       value_4 like 'C%' or
       value_4 like 'D%' or
       value_4 like 'E%')
```

```
  and (value_5 like '0%' or value_5 like 'b%' or value_5 like '0%' or
value_5 like '7%' or value_5 like 'Z%')
  and (char_1 between 'a' and 'z')
  and (char_2 between 'a' and 'z' or char_2 between 'A' and 'Z')
  and (char_3 between 'a' and 'z' or char_3 between 'A' and 'Z' or
char_3 between '0' and '9')
  and date_1 in (select date_1 from master_2 where date_1 between
to_date('01-JAN-1970','DD-MON-YYYY') and sysdate)
  and date_2 in (select date_2 from master_2 where date_2 between
to_date('01-JAN-1971','DD-MON-YYYY') and sysdate)
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_2_QUERY%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
MASTER_3_CHAR_1_IDX
MASTER_3_CHAR_2_IDX
MASTER_3_CHAR_3_IDX
MASTER_3_NUM_1_IDX
MASTER_3_NUM_2_IDX
MASTER_3_NUM_3_IDX
MASTER_3_NUM_4_IDX
MASTER_3_NUM_5_IDX
MASTER_3_VALUE_1_IDX
MASTER_3_VALUE_2_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
select count(*)
from master_3
where num_1 between 1 and 98768000
  and num_2 between 28 and 291381400
  and num_3 >= (select min(num_3) from master_3)
  and num_4 <= (select avg(num_4) from master_3)
  and num_5 <= (select max(num_5) from master_3)
```

```
  and (char_1 between 'a' and 'z')
  and (char_2 between 'a' and 'z' or char_2 between 'A' and 'Z')
  and (char_3 between 'a' and 'z' or char_3 between 'A' and 'Z' or
char_3 between '0' and '9')
  and value_1 in (select value_1 from master_3 where value_1 like
'%A%' or value_1 like '%B%')
  and value_2 in (select value_2 from master_3 where value_2 like
'%C%')
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/
--TIME_COST
set timing on
select /*+ MASTER_3_QUERY*/
count(*)
from master_3
where num_1 between 1 and 98768000
  and num_2 between 28 and 291381400
  and num_3 >= (select min(num_3) from master_3)
  and num_4 <= (select avg(num_4) from master_3)
  and num_5 <= (select max(num_5) from master_3)
  and (char_1 between 'a' and 'z')
  and (char_2 between 'a' and 'z' or char_2 between 'A' and 'Z')
  and (char_3 between 'a' and 'z' or char_3 between 'A' and 'Z' or
char_3 between '0' and '9')
  and value_1 in (select value_1 from master_3 where value_1 like
'%A%' or value_1 like '%B%')
  and value_2 in (select value_2 from master_3 where value_2 like
'%C%')
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_3_QUERY%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
```

```
--INDEXES
MASTER_1_CHAR_1_IDX
M1_L2_T1_CHAR_1_IDX
M3_L3_T1_CHAR_1_IDX
M1_L2_T2_VALUE_1_IDX
M1_L3_T3_VALUE_2_IDX
MASTER_2_DATE_1_IDX
M2_L2_T1_DATE_2_IDX
M2_L3_T3_CHAR_1_IDX
MASTER_3_NUM_1_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT MAX (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1
         WHERE char_1 BETWEEN 'A' AND 'Z'
        UNION
        SELECT COUNT (*) counter
          FROM master_1_l2_tab1
         WHERE char_1 BETWEEN 'A' AND 'Z'
        UNION
        SELECT COUNT (*) counter
          FROM master_1_l3_tab1
         WHERE char_1 BETWEEN 'A' AND 'Z'
        UNION
        SELECT COUNT (value_1) counter
          FROM master_1_l2_tab2
         WHERE value_1 IN
                  ('8vI57wmjS',
                   'HQyF36601',
                   'wYSrlb3EB',
                   'x91rimHk4',
                   'nFN1A0s9l',
                   'gx513eSJH',
                   'e5lCRBSpr',
                   '1mV9SIBJp',
                   'TJ95Q2j87',
                   'phl2Jg5Q3')
        UNION
        SELECT COUNT (value_2)
          FROM master_1_l3_tab3
         WHERE value_2 LIKE '%a%' OR value_2 LIKE '%b%'
        UNION
        SELECT COUNT (date_1)
          FROM master_2
         WHERE date_1 BETWEEN TO_DATE ('14-JUL-2010', 'DD-MON-YYYY')
                          AND TO_DATE ('13-AUG-2011', 'DD-MON-YYYY')
        UNION
```

```
        SELECT COUNT (date_2)
          FROM master_2_l2_tab1
         WHERE date_2 BETWEEN TO_DATE ('14-JAN-2010', 'DD-MON-YYYY')
                          AND TO_DATE ('13-FEB-2011', 'DD-MON-YYYY')
        UNION
        SELECT COUNT (*)
          FROM master_2_l3_tab3
         WHERE char_1 IN ('A', '1', '2', '3')
        UNION
        SELECT COUNT (*)
          FROM    master_3 a
               JOIN
                  master_3_l2_tab3 b
               ON (a.pk_value = b.parent_link)
         WHERE    a.num_1 BETWEEN 1123 AND 3829193
               OR b.comp_1 BETWEEN 'A' AND 'Z')
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ GENERAL_JOIN_1*/
MAX (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1
         WHERE char_1 BETWEEN 'A' AND 'Z'
        UNION
        SELECT COUNT (*) counter
          FROM master_1_l2_tab1
         WHERE char_1 BETWEEN 'A' AND 'Z'
        UNION
        SELECT COUNT (*) counter
          FROM master_1_l3_tab1
         WHERE char_1 BETWEEN 'A' AND 'Z'
        UNION
        SELECT COUNT (value_1) counter
          FROM master_1_l2_tab2
         WHERE value_1 IN
                  ('8vI57wmjS',
                   'HQyF36601',
                   'wYSrlb3EB',
                   'x91rimHk4',
                   'nFN1A0s9l',
                   'gx513eSJH',
                   'e5lCRBSpr',
                   '1mV9SIBJp',
```

```
                    'TJ95Q2j87',
                    'phl2Jg5Q3')
        UNION
        SELECT COUNT (value_2)
          FROM master_1_l3_tab3
         WHERE value_2 LIKE '%a%' OR value_2 LIKE '%b%'
        UNION
        SELECT COUNT (date_1)
          FROM master_2
         WHERE date_1 BETWEEN TO_DATE ('14-JUL-2010', 'DD-MON-YYYY')
                          AND TO_DATE ('13-AUG-2011', 'DD-MON-YYYY')
        UNION
        SELECT COUNT (date_2)
          FROM master_2_l2_tab1
         WHERE date_2 BETWEEN TO_DATE ('14-JAN-2010', 'DD-MON-YYYY')
                          AND TO_DATE ('13-FEB-2011', 'DD-MON-YYYY')
        UNION
        SELECT COUNT (*)
          FROM master_2_l3_tab3
         WHERE char_1 IN ('A', '1', '2', '3')
        UNION
        SELECT COUNT (*)
          FROM    master_3 a
               JOIN
                  master_3_l2_tab3 b
               ON (a.pk_value = b.parent_link)
         WHERE    a.num_1 BETWEEN 1123 AND 3829193
               OR b.comp_1 BETWEEN 'A' AND 'Z')
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%GENERAL_JOIN_1%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
MASTER_1_VALUE_1_IDX
MASTER_1_CHAR_1_IDX
MASTER_1_DATE_1_IDX
MASTER_1_DATE_2_IDX
MASTER_2_VALUE_1_IDX
```

```
MASTER_3_PK_IDX
M3_L3_T1_CHAR_1_IDX
M3_L2_T2_PK_IDX
M3_L3_T3_PK_IDX
MASTER_3_NUM_1_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT MAX (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1 a JOIN master_2 b ON (a.master_2_link =
b.pk_value)
         WHERE     a.date_1 IN
                      (SELECT date_1
                         FROM master_1
                        WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                      'DD-MON-YYYY')
                                         AND SYSDATE)
               AND a.date_2 IN
                      (SELECT date_2
                         FROM master_1
                        WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                      'DD-MON-YYYY')
                                         AND SYSDATE)
               AND a.char_1 IN (SELECT char_1
                                  FROM master_1
                                 WHERE char_1 BETWEEN 'A' AND 'Z')
               AND (a.value_1 LIKE 'a%' OR a.value_1 LIKE 'b%')
               AND (B.VALUE_1 LIKE '%A%' OR B.VALUE_1 LIKE '%b%')
        UNION
        SELECT COUNT (*) counter
          FROM    master_3_l2_tab1 a
               JOIN
                  master_3_l3_tab1 b
               ON (b.parent_link = a.pk_value)
         WHERE     a.parent_link NOT IN (SELECT pk_value FROM
master_3)
               AND B.CHAR_1 = 'a'
        UNION
        SELECT COUNT (*) counter
          FROM    master_3_l2_tab2 a
               JOIN
                  master_3_l3_tab2 b
               ON (b.parent_link = a.pk_value)
         WHERE a.pk_value NOT IN (SELECT pk_value
                                   FROM master_3_l2_tab2
                                  WHERE MOD (pk_value, 2) = 1)
        UNION
        SELECT COUNT (*) COUNTER
```

```
          FROM master_3_l2_tab3 a
               JOIN master_3 b
                  ON (a.parent_link = b.pk_value)
               JOIN master_3_l3_tab3 c
                  ON (C.PARENT_LINK = b.pk_value)
         WHERE     a.pk_value NOT IN (SELECT pk_value
                                        FROM master_3_l3_tab3
                                       WHERE MOD (pk_value, 2) = 4)
               AND b.num_1 BETWEEN 2355 AND 234445)
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ GENERAL_JOIN_2*/
MAX (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_1 a JOIN master_2 b ON (a.master_2_link =
b.pk_value)
         WHERE     a.date_1 IN
                      (SELECT date_1
                         FROM master_1
                        WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                      'DD-MON-YYYY')
                                         AND SYSDATE)
               AND a.date_2 IN
                      (SELECT date_2
                         FROM master_1
                        WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                      'DD-MON-YYYY')
                                         AND SYSDATE)
               AND a.char_1 IN (SELECT char_1
                                  FROM master_1
                                 WHERE char_1 BETWEEN 'A' AND 'Z')
               AND (a.value_1 LIKE 'a%' OR a.value_1 LIKE 'b%')
               AND (B.VALUE_1 LIKE '%A%' OR B.VALUE_1 LIKE '%b%')
        UNION
        SELECT COUNT (*) counter
          FROM    master_3_l2_tab1 a
               JOIN
                  master_3_l3_tab1 b
               ON (b.parent_link = a.pk_value)
         WHERE     a.parent_link NOT IN (SELECT pk_value FROM
master_3)
               AND B.CHAR_1 = 'a'
        UNION
```

```
        SELECT COUNT (*) counter
          FROM    master_3_l2_tab2 a
               JOIN
                  master_3_l3_tab2 b
               ON (b.parent_link = a.pk_value)
         WHERE a.pk_value NOT IN (SELECT pk_value
                                    FROM master_3_l2_tab2
                                   WHERE MOD (pk_value, 2) = 1)
        UNION
        SELECT COUNT (*) COUNTER
          FROM master_3_l2_tab3 a
               JOIN master_3 b
                  ON (a.parent_link = b.pk_value)
               JOIN master_3_l3_tab3 c
                  ON (C.PARENT_LINK = b.pk_value)
         WHERE     a.pk_value NOT IN (SELECT pk_value
                                        FROM master_3_l3_tab3
                                       WHERE MOD (pk_value, 2) = 4)
               AND b.num_1 BETWEEN 2355 AND 234445)
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%GENERAL_JOIN_2%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
M1_L2_T3_DATE_1_IDX
M1_L3_T3_CHAR_1_IDX
M1_L2_T3_DATE_1_IDX
MASTER_1_PK_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT COUNT (*) counter
  FROM master_1 a
       JOIN master_1_l2_tab3 b
          ON (a.pk_value = b.parent_link)
```

```
       JOIN master_1_l3_tab3 c
          ON (b.pk_value = c.parent_link)
 where C.PK_VALUE NOT IN (SELECT DISTINCT master_2_link FROM master_1)
 and c.char_1 between 'A' and 'B'
 AND b.date_1 = to_date('JAN-01-2012','MON-DD-YYYY')
 and mod(A.PK_VALUE, 2) = 6
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ GENERAL_JOIN_3*/
COUNT (*) counter
  FROM master_1 a
       JOIN master_1_l2_tab3 b
          ON (a.pk_value = b.parent_link)
       JOIN master_1_l3_tab3 c
          ON (b.pk_value = c.parent_link)
 where C.PK_VALUE NOT IN (SELECT DISTINCT master_2_link FROM master_1)
 and c.char_1 between 'A' and 'B'
 AND b.date_1 = to_date('JAN-01-2012','MON-DD-YYYY')
 and mod(A.PK_VALUE, 2) = 6
/
exit
/
--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%GENERAL_JOIN_3%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
M1_L2_T1_CHAR_1_IDX
M1_L3_T1_DATE_1_IDX
M2_L2_T1_DATE_7_IDX
M2_L3_T1_CHAR_2_IDX
M3_L2_T1_NUM_COMP_1_IDX
M3_L3_T1_VALUE_1_IDX
```

```
M3_L3_T2_VALUE_1_IDX
M3_L3_T2_VALUE_2_IDX
MASTER_3_NUM_1_IDX
MASTER_3_NUM_2_IDX
MASTER_3_NUM_3_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT MAX (counter)
  FROM (SELECT COUNT (*) counter
          FROM    master_1_l2_tab1 a
               JOIN
                  master_1_l3_tab1 b
               ON (a.pk_value = b.parent_link)
         WHERE     A.CHAR_1 BETWEEN 'a' AND 'x'
               AND B.DATE_1 BETWEEN TO_DATE ('31-MAR-2012', 'DD-MON-
YYYY')
                                AND SYSDATE
        UNION
        SELECT COUNT (*) counter
          FROM    master_2_l2_tab1 a
               JOIN
                  master_2_l3_tab1 b
               ON (a.pk_value = b.parent_link)
         WHERE     A.DATE_7 BETWEEN TO_DATE ('31-MAR-2012', 'DD-MON-
YYYY')
                                AND SYSDATE
               AND B.CHAR_2 IN ('a', 'b', 'c', 'd', 'e', 'A', 'B',
'C', 'D')
        UNION
        SELECT COUNT (*) counter
          FROM    master_3_l2_tab1 a
               JOIN
                  master_3_l3_tab1 b
               ON (a.pk_value = b.parent_link)
         WHERE     A.NUM_COMP_1 BETWEEN 39283 AND 283914
               AND b.value_1 IN
                      ('3K7139V6ai',
                       '9JUPS61wI0',
                       '7vS97etdUD',
                       'xd5Insf25s',
                       '5606Y4QVT7',
                       'g3fwyao83f',
                       'jTnCRknVIl',
                       'IjjIc1dnkP',
                       '5EE969PSdh',
                       'QTjGnmN2Qt',
                       '6f318Xnytf',
                       'Edpf5S3ql8',
```

```
                'Hm1I30DT8f',
                'uR25h2u03F',
                'tV3nDnUuUu')
UNION
SELECT COUNT (*) counter
  FROM    master_3_l2_tab2 a
       JOIN
          master_3_l3_tab2 b
       ON (a.pk_value = b.parent_link)
 WHERE     b.value_1 IN
              ('IY9UaY1N98',
               '6cHk2mN8L5',
               'UbrrO55FUW',
               '9WVAyW2y6u',
               'k4VG9uAmJ5',
               '20otsii38m',
               'RNQ9irc26R',
               'l7VdgWe22R',
               'KI382JQ1kP',
               'jMcGoxWuGM',
               'tlM3g02MTu',
               '7iImx0h2b5',
               '3ywB17VLhl',
               'iq1eAID9HI',
               '30A1p977M3')
       AND b.value_2 IN
              ('NRqr6oSh4E',
               'Sc1p45Hyh1',
               'nHl3k615G7',
               'rFLvlqBfWp',
               '5TFBekA13m',
               'xb7leIyAqn',
               'BJ5iG187FE',
               'dC8370aL5K',
               'bO3TRLK0Bq',
               '107R65vFJd',
               'sXeIi9pyHg',
               'X5BXSEA6A1',
               'We5SWDd5pW',
               '5g5A0F71e4',
               '3CQvfkhIe6',
               'xbatnyv0IR')
UNION
SELECT COUNT (*) counter
  FROM master_3 a
       JOIN master_3_l2_tab3 b
          ON (a.pk_value = b.parent_link)
       JOIN master_3_l3_tab3 c
          ON (b.pk_value = c.parent_link)
 WHERE (   A.NUM_1 BETWEEN 6000 AND 7000
        OR A.NUM_2 BETWEEN 7000 AND 8000
        OR A.NUM_3 BETWEEN 8000 AND 9000))
```

```
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ GENERAL_JOIN_4*/
MAX (counter)
  FROM (SELECT COUNT (*) counter
          FROM    master_1_l2_tab1 a
                 JOIN
                    master_1_l3_tab1 b
                 ON (a.pk_value = b.parent_link)
         WHERE     A.CHAR_1 BETWEEN 'a' AND 'x'
                 AND B.DATE_1 BETWEEN TO_DATE ('31-MAR-2012', 'DD-MON-
YYYY')
                                 AND SYSDATE
        UNION
        SELECT COUNT (*) counter
          FROM    master_2_l2_tab1 a
                 JOIN
                    master_2_l3_tab1 b
                 ON (a.pk_value = b.parent_link)
         WHERE     A.DATE_7 BETWEEN TO_DATE ('31-MAR-2012', 'DD-MON-
YYYY')
                                 AND SYSDATE
                 AND B.CHAR_2 IN ('a', 'b', 'c', 'd', 'e', 'A', 'B',
'C', 'D')
        UNION
        SELECT COUNT (*) counter
          FROM    master_3_l2_tab1 a
                 JOIN
                    master_3_l3_tab1 b
                 ON (a.pk_value = b.parent_link)
         WHERE     A.NUM_COMP_1 BETWEEN 39283 AND 283914
                 AND b.value_1 IN
                        ('3K7139V6ai',
                         '9JUPS61wI0',
                         '7vS97etdUD',
                         'xd5Insf25s',
                         '5606Y4QVT7',
                         'g3fwyao83f',
                         'jTnCRknVIl',
                         'IjjIc1dnkP',
                         '5EE969PSdh',
                         'QTjGnmN2Qt',
                         '6f318Xnytf',
```

```
               'Edpf5S3ql8',
               'Hm1I30DT8f',
               'uR25h2u03F',
               'tV3nDnUuUu')
UNION
SELECT COUNT (*) counter
  FROM    master_3_l2_tab2 a
       JOIN
          master_3_l3_tab2 b
       ON (a.pk_value = b.parent_link)
 WHERE     b.value_1 IN
              ('IY9UaY1N98',
               '6cHk2mN8L5',
               'UbrrO55FUW',
               '9WVAyW2y6u',
               'k4VG9uAmJ5',
               '20otsii38m',
               'RNQ9irc26R',
               'l7VdgWe22R',
               'KI382JQ1kP',
               'jMcGoxWuGM',
               'tlM3g02MTu',
               '7iImx0h2b5',
               '3ywB17VLhl',
               'iq1eAID9HI',
               '30A1p977M3')
       AND b.value_2 IN
              ('NRqr6oSh4E',
               'Sc1p45Hyh1',
               'nHl3k615G7',
               'rFLvlqBfWp',
               '5TFBekA13m',
               'xb7leIyAqn',
               'BJ5iG187FE',
               'dC8370aL5K',
               'bO3TRLK0Bq',
               '107R65vFJd',
               'sXeIi9pyHg',
               'X5BXSEA6A1',
               'We5SWDd5pW',
               '5g5A0F71e4',
               '3CQvfkhIe6',
               'xbatnyv0IR')
UNION
SELECT COUNT (*) counter
  FROM master_3 a
       JOIN master_3_l2_tab3 b
          ON (a.pk_value = b.parent_link)
       JOIN master_3_l3_tab3 c
          ON (b.pk_value = c.parent_link)
 WHERE (   A.NUM_1 BETWEEN 6000 AND 7000
        OR A.NUM_2 BETWEEN 7000 AND 8000
```

```
                OR A.NUM_3 BETWEEN 8000 AND 9000))
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%GENERAL_JOIN_4%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------------
-------------------------------------------
--INDEXES
MASTER_1_CHAR_3_IDX
MASTER_1_DATE_1_IDX
MASTER_1_DATE_3_IDX
MASTER_1_DATE_4_IDX
MASTER_1_VALUE_1_IDX
MASTER_1_VALUE_2_IDX
MASTER_3_NUM_1_IDX
MASTER_3_NUM_2_IDX
MASTER_3_NUM_3_IDX
MASTER_3_NUM_4_IDX
MASTER_3_NUM_5_IDX
MASTER_3_CHAR_3_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT SUM (counter),
       AVG (counter),
       MAX (counter),
       MIN (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_3
         WHERE     (   num_1 BETWEEN 1 AND 100
                    OR num_2 BETWEEN 1 AND 1000
                    OR num_3 BETWEEN 1 AND 10000
                    OR num_4 BETWEEN 1 AND 100000
                    OR num_5 BETWEEN 1 AND 1000000)
               AND (   char_1 BETWEEN 'A' AND 'D'
                    OR char_2 BETWEEN 'A' AND 'Y'
                    OR char_3 BETWEEN 'A' AND 'M')
```

```
       UNION
       SELECT COUNT (*) counter
         FROM master_2
        WHERE     value_1 IN (SELECT value_1
                                FROM master_2
                               WHERE value_1 LIKE '%A%')
              AND value_2 IN (SELECT value_2
                                FROM master_2
                               WHERE value_2 LIKE '%B%')
              AND value_3 IN (SELECT value_3
                                FROM master_2
                               WHERE value_3 LIKE '%C%')
              AND (   value_4 LIKE 'A%'
                   OR value_4 LIKE 'B%'
                   OR value_4 LIKE 'C%'
                   OR value_4 LIKE 'D%'
                   OR value_4 LIKE 'E%')
              AND (   value_5 LIKE '0%'
                   OR value_5 LIKE 'b%'
                   OR value_5 LIKE '0%'
                   OR value_5 LIKE '7%'
                   OR value_5 LIKE 'Z%')
              AND (char_1 BETWEEN 'a' AND 'z')
              AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
              AND (   char_3 BETWEEN 'a' AND 'z'
                   OR char_3 BETWEEN 'A' AND 'Z'
                   OR char_3 BETWEEN '0' AND '9')
              AND date_1 IN
                     (SELECT date_1
                        FROM master_2
                       WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                     'DD-MON-YYYY')
                                        AND SYSDATE)
              AND date_2 IN
                     (SELECT date_2
                        FROM master_2
                       WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
                                                     'DD-MON-YYYY')
                                        AND SYSDATE))
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ GENERAL_QUERY_5*/
```

```
       SUM (counter),
       AVG (counter),
       MAX (counter),
       MIN (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_3
         WHERE      (    num_1 BETWEEN 1 AND 100
                      OR num_2 BETWEEN 1 AND 1000
                      OR num_3 BETWEEN 1 AND 10000
                      OR num_4 BETWEEN 1 AND 100000
                      OR num_5 BETWEEN 1 AND 1000000)
               AND (    char_1 BETWEEN 'A' AND 'D'
                     OR char_2 BETWEEN 'A' AND 'Y'
                     OR char_3 BETWEEN 'A' AND 'M')
        UNION
        SELECT COUNT (*) counter
          FROM master_2
         WHERE     value_1 IN (SELECT value_1
                                  FROM master_2
                                 WHERE value_1 LIKE '%A%')
               AND value_2 IN (SELECT value_2
                                  FROM master_2
                                 WHERE value_2 LIKE '%B%')
               AND value_3 IN (SELECT value_3
                                  FROM master_2
                                 WHERE value_3 LIKE '%C%')
               AND (    value_4 LIKE 'A%'
                     OR value_4 LIKE 'B%'
                     OR value_4 LIKE 'C%'
                     OR value_4 LIKE 'D%'
                     OR value_4 LIKE 'E%')
               AND (    value_5 LIKE '0%'
                     OR value_5 LIKE 'b%'
                     OR value_5 LIKE '0%'
                     OR value_5 LIKE '7%'
                     OR value_5 LIKE 'Z%')
               AND (char_1 BETWEEN 'a' AND 'z')
               AND (char_2 BETWEEN 'a' AND 'z' OR char_2 BETWEEN 'A'
AND 'Z')
               AND (    char_3 BETWEEN 'a' AND 'z'
                     OR char_3 BETWEEN 'A' AND 'Z'
                     OR char_3 BETWEEN '0' AND '9')
               AND date_1 IN
                      (SELECT date_1
                         FROM master_2
                        WHERE date_1 BETWEEN TO_DATE ('01-JAN-1970',
                                                       'DD-MON-YYYY')
                                         AND SYSDATE)
               AND date_2 IN
                      (SELECT date_2
                         FROM master_2
                        WHERE date_2 BETWEEN TO_DATE ('01-JAN-1971',
```

```
                                                 'DD-MON-YYYY')
                                    AND SYSDATE))
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%GENERAL_QUERY_5%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
MASTER_2_PK_IDX
MASTER_3_PK_IDX
M3_L2_T1_NUM_COMP_1_IDX
MASTER_3_NUM_1_IDX
MASTER_2_CHAR_1_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT count(*)
  FROM master_3_l2_tab1 a, master_3 b, master_2 c
 WHERE     A.PARENT_LINK = b.pk_value
       AND B.MASTER_2_LINK = c.pk_value
       AND A.NUM_COMP_1 BETWEEN 90 AND 95000
       AND B.NUM_1 IN
              (155449580,
               643538844,
               267005443,
               149103222,
               134317259,
               226229117,
               975327780,
               811720886,
               73512047)
       AND C.CHAR_1 IN
              ('L',
               'i',
               '5',
               '9',
```

```
               'm',
               'W',
               'J',
               'i',
               'l',
               'A',
               'B',
               'C',
               'D',
               'E',
               'F',
               'G')
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ GENERAL_QUERY_6*/
count(*)
  FROM master_3_l2_tab1 a, master_3 b, master_2 c
 WHERE     A.PARENT_LINK = b.pk_value
       AND B.MASTER_2_LINK = c.pk_value
       AND A.NUM_COMP_1 BETWEEN 90 AND 95000
       AND B.NUM_1 IN
              (155449580,
               643538844,
               267005443,
               149103222,
               134317259,
               226229117,
               975327780,
               811720886,
               73512047)
       AND C.CHAR_1 IN
              ('L',
               'i',
               '5',
               '9',
               'm',
               'W',
               'J',
               'i',
               'l',
               'A',
               'B',
               'C',
```

```
              'D',
              'E',
              'F',
              'G')
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%GENERAL_QUERY_6%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
MASTER_2_DATE_2_IDX
MASTER_3_CHAR_1_IDX
M1_L3_T1_VALUE_1_IDX
MASTER_3_VALUE_2_IDX
MASTER_3_PK_IDX
MASTER_2_PK_IDX
MASTER_1_DATE_2_IDX
MASTER_1_DATE_1_IDX
MASTER_3_LINKER

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT COUNT (*)
  FROM master_1 d JOIN master_3 e ON (d.pk_value = e.pk_value)
 WHERE     (d.date_1) = TO_DATE ('27-JUN-2010', 'DD-MON-YYYY')
       AND (d.date_2) = TO_DATE ('22-DEC-1993', 'DD-MON-YYYY')
       AND EXISTS
              (SELECT *
                 FROM master_2
                WHERE (date_2) = TO_DATE ('25-JAN-2009', 'DD-MON-
YYYY'))
       AND EXISTS
              (SELECT *
                 FROM master_3
                WHERE char_1 = 'B')
       AND EXISTS
```

```
              (SELECT *
                 FROM    master_1_l2_tab1 a
                      JOIN
                         master_1_l3_tab1 b
                      ON (a.pk_value = b.parent_link)
                WHERE B.VALUE_1 IN
                         ('WvNR99AULn',
                          'R75Bv2TRef',
                          '4cEThl2Mm6',
                          '7u3X162vjl',
                          '1NoqnQaYXe',
                          'uU43uKAUUL',
                          '0T63MKf858',
                          '4MG191Y3TC',
                          '6Lww0T7NRo',
                          'O1w6WkXDU5'))
       AND EXISTS
              (SELECT value_2
                 FROM master_3
                WHERE value_2 IN
                         ('e2Qcse48Gp',
                          '6e8nTb1FyH',
                          'BA15ACOxO0',
                          '24Fg07idnW',
                          '5D91o0M1U9',
                          'twWaKStLGh',
                          'UY8u54Pu0P',
                          'ScUTql3w88'))
       AND d.pk_value IN (SELECT pk_value FROM master_2)
       AND E.MASTER_1_LINK IN (SELECT master_2_link FROM master_3)
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ GENERAL_QUERY_7*/
COUNT (*)
  FROM master_1 d JOIN master_3 e ON (d.pk_value = e.pk_value)
 WHERE     (d.date_1) = TO_DATE ('27-JUN-2010', 'DD-MON-YYYY')
       AND (d.date_2) = TO_DATE ('22-DEC-1993', 'DD-MON-YYYY')
       AND EXISTS
              (SELECT *
                 FROM master_2
                WHERE (date_2) = TO_DATE ('25-JAN-2009', 'DD-MON-
YYYY'))
       AND EXISTS
```

```
              (SELECT *
                 FROM master_3
                WHERE char_1 = 'B')
       AND EXISTS
              (SELECT *
                 FROM    master_1_l2_tab1 a
                      JOIN
                         master_1_l3_tab1 b
                      ON (a.pk_value = b.parent_link)
                WHERE B.VALUE_1 IN
                         ('WvNR99AULn',
                          'R75Bv2TRef',
                          '4cEThl2Mm6',
                          '7u3X162vjl',
                          '1NoqnQaYXe',
                          'uU43uKAUUL',
                          '0T63MKf858',
                          '4MG191Y3TC',
                          '6Lww0T7NRo',
                          'O1w6WkXDU5'))
       AND EXISTS
              (SELECT value_2
                 FROM master_3
                WHERE value_2 IN
                         ('e2Qcse48Gp',
                          '6e8nTb1FyH',
                          'BA15ACOxO0',
                          '24Fg07idnW',
                          '5D91o0M1U9',
                          'twWaKStLGh',
                          'UY8u54Pu0P',
                          'ScUTql3w88'))
       AND d.pk_value IN (SELECT pk_value FROM master_2)
       AND E.MASTER_1_LINK IN (SELECT master_2_link FROM master_3)
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%GENERAL_QUERY_7%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
------------------------------------------------------------------------
--------------------------------------------
```

```
--INDEXES
MASTER_3_PK_IDX
MASTER_2_PK_IDX
MASTER_1_PK_IDX
MASTER_3_LINKER
MASTER_3_CHAR_1_IDX
M3_L2_T1_NUM_COMP_1_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT COUNT (*)
  FROM master_3 d join master_3_l2_tab1 e on (d.pk_value =
e.parent_link)
 WHERE d.pk_value >=
          (SELECT AVG (C.MASTER_2_LINK)
             FROM master_1 c
            WHERE c.pk_value <=
                     (SELECT AVG (B.MASTER_1_LINK)
                        FROM master_2 b
                       WHERE b.pk_value >= (SELECT AVG (a.pk_value)
                                              FROM master_3 a
                                             WHERE a.pk_value / 2 >
5000)))
  and D.MASTER_2_LINK between 5000 and 5098
  and d.char_1 between '1' and '9'
 and e.num_comp_1 between 343144008 and 343144015
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ GENERAL_QUERY_8*/
COUNT (*)
  FROM master_3 d join master_3_l2_tab1 e on (d.pk_value =
e.parent_link)
 WHERE d.pk_value >=
          (SELECT AVG (C.MASTER_2_LINK)
             FROM master_1 c
            WHERE c.pk_value <=
                     (SELECT AVG (B.MASTER_1_LINK)
                        FROM master_2 b
                       WHERE b.pk_value >= (SELECT AVG (a.pk_value)
                                              FROM master_3 a
```

```
                                            WHERE a.pk_value / 2 >
5000)))
  and D.MASTER_2_LINK between 5000 and 5098
  and d.char_1 between '1' and '9'
 and e.num_comp_1 between 343144008 and 343144015
/
exit
/
--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%GENERAL_QUERY_8%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
MASTER_3_PK_IDX
MASTER_2_PK_IDX
MASTER_1_PK_IDX
MASTER_3_LINKER
MASTER_3_CHAR_1_IDX
M3_L2_T1_NUM_COMP_1_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT COUNT (*)
  FROM master_3 d join master_3_l2_tab1 e on (d.pk_value =
e.parent_link)
 WHERE d.pk_value >=
          (SELECT AVG (C.MASTER_2_LINK)
             FROM master_1 c
            WHERE c.pk_value <=
                     (SELECT AVG (B.MASTER_1_LINK)
                        FROM master_2 b
                       WHERE b.pk_value >= (SELECT AVG (a.pk_value)
                                             FROM master_3 a
                                            WHERE a.pk_value / 2 >
5000)))
  and D.MASTER_2_LINK between 5000 and 5098
  and d.char_1 between '1' and '9'
 and e.num_comp_1 between 343144008 and 343144015
/
```

```
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ GENERAL_QUERY_8*/
COUNT (*)
  FROM master_3 d join master_3_l2_tab1 e on (d.pk_value =
e.parent_link)
 WHERE d.pk_value >=
          (SELECT AVG (C.MASTER_2_LINK)
             FROM master_1 c
            WHERE c.pk_value <=
                     (SELECT AVG (B.MASTER_1_LINK)
                        FROM master_2 b
                       WHERE b.pk_value >= (SELECT AVG (a.pk_value)
                                              FROM master_3 a
                                             WHERE a.pk_value / 2 >
5000)))
  and D.MASTER_2_LINK between 5000 and 5098
  and d.char_1 between '1' and '9'
 and e.num_comp_1 between 343144008 and 343144015
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%GENERAL_QUERY_8%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
MASTER_4_CHAR_1_IDX
MASTER_4_CHAR_2_IDX
MASTER_4_CHAR_3_IDX

--GET_COST
delete from plan_table
```

```
/
explain plan
set statement_id = 'A' for
select sum(counter)
from
              (SELECT count(*) counter
                   FROM master_4
                  WHERE char_1 IN (SELECT char_1
                                     FROM master_4
                                    WHERE ASCII (char_3) BETWEEN 85
AND 90)
                 UNION
                 SELECT count(*) counter
                   FROM master_4
                  WHERE char_1 IN (SELECT char_2
                                     FROM master_4
                                    WHERE ASCII (char_1) BETWEEN 85
AND 90)
                 UNION
                 SELECT count(*) counter
                   FROM master_4
                  WHERE char_1 IN (SELECT char_3
                                     FROM master_4
                                    WHERE ASCII (char_2) BETWEEN 85
AND 90))
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
select /*+ MASTER_4_QUERY_1*/
sum(counter)
from
              (SELECT count(*) counter
                   FROM master_4
                  WHERE char_1 IN (SELECT char_1
                                     FROM master_4
                                    WHERE ASCII (char_3) BETWEEN 85
AND 90)
                 UNION
                 SELECT count(*) counter
                   FROM master_4
                  WHERE char_1 IN (SELECT char_2
                                     FROM master_4
                                    WHERE ASCII (char_1) BETWEEN 85
AND 90)
```

```
                UNION
                SELECT count(*) counter
                  FROM master_4
                 WHERE char_1 IN (SELECT char_3
                                    FROM master_4
                                   WHERE ASCII (char_2) BETWEEN 85
AND 90))
/
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_4_QUERY_1%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
MASTER_4_CHAR_1_IDX
MASTER_4_CHAR_2_IDX
MASTER_4_CHAR_3_IDX
MASTER_4_DATE_1_IDX
MASTER_4_DATE_2_IDX
MASTER_4_DATE_3_IDX
MASTER_4_VALUE_1_IDX
MASTER_4_VALUE_2_IDX
MASTER_4_VALUE_3_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT SUM (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_4
         WHERE char_1 IN (SELECT char_1
                            FROM master_4
                           WHERE ASCII (char_3) BETWEEN 85 AND 90)
        UNION
        SELECT COUNT (*) counter
          FROM master_4
         WHERE char_1 IN (SELECT char_2
                            FROM master_4
```

```
                       WHERE ASCII (char_1) BETWEEN 85 AND 90)
UNION
SELECT COUNT (*) counter
  FROM master_4
 WHERE char_1 IN (SELECT char_3
                    FROM master_4
                   WHERE ASCII (char_2) BETWEEN 85 AND 90)
UNION
SELECT COUNT (*) counter
  FROM master_4
 WHERE value_1 IN
          (SELECT value_1
             FROM master_4
            WHERE value_1 IN
                     ('Waneta',
                      'Queenie',
                      'Shante',
                      'Barbera',
                      'Thalia',
                      'Kasie',
                      'Courtney',
                      'Pearlene',
                      'Hattie',
                      'Darrin',
                      'Marcellus',
                      'Kandi',
                      'Jenifer',
                      'Chanel',
                      'Aurelio',
                     'Felicia'))
UNION
SELECT COUNT (*) counter
  FROM master_4
 WHERE value_3 IN
          (SELECT value_2
             FROM master_4
            WHERE value_2 IN
                     ('Waneta',
                      'Queenie',
                      'Shante',
                      'Barbera',
                      'Thalia',
                      'Kasie',
                      'Courtney',
                      'Pearlene',
                      'Hattie',
                      'Darrin',
                      'Marcellus',
                      'Kandi',
                      'Jenifer',
                      'Chanel',
                      'Aurelio',
```

```
                                  'Felicia'))
          UNION
          SELECT COUNT (*) counter
            FROM master_4
           WHERE (value_1, value_2) IN
                    (SELECT value_3, value_3
                       FROM master_4
                      WHERE value_3 IN
                              ('Waneta',
                               'Queenie',
                               'Shante',
                               'Barbera',
                               'Thalia',
                               'Kasie',
                               'Courtney',
                               'Pearlene',
                               'Hattie',
                               'Darrin',
                               'Marcellus',
                               'Kandi',
                               'Jenifer',
                               'Chanel',
                               'Aurelio',
                               'Felicia'))
          UNION
          SELECT COUNT (*) counter
            FROM master_4
           WHERE date_2 BETWEEN (SELECT MIN (date_2) FROM master_4)
                            AND (SELECT MIN (date_2) + 35 FROM master_4)
          UNION
          SELECT COUNT (*) counter
            FROM master_4
           WHERE date_2 NOT BETWEEN (SELECT MIN (date_2) FROM master_4)
                                AND (SELECT MIN (date_2) + 50 FROM
master_4)
          UNION
          SELECT COUNT (*) counter
            FROM master_4
           WHERE date_3 BETWEEN TO_DATE ('01-01-2000', 'DD-MM-YYYY')
                            AND SYSDATE - 25)
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
SELECT /*+ MASTER_4_QUERY_2*/
```

```
SUM (counter)
  FROM (SELECT COUNT (*) counter
          FROM master_4
         WHERE char_1 IN (SELECT char_1
                            FROM master_4
                           WHERE ASCII (char_3) BETWEEN 85 AND 90)
        UNION
        SELECT COUNT (*) counter
          FROM master_4
         WHERE char_1 IN (SELECT char_2
                            FROM master_4
                           WHERE ASCII (char_1) BETWEEN 85 AND 90)
        UNION
        SELECT COUNT (*) counter
          FROM master_4
         WHERE char_1 IN (SELECT char_3
                            FROM master_4
                           WHERE ASCII (char_2) BETWEEN 85 AND 90)
        UNION
        SELECT COUNT (*) counter
          FROM master_4
         WHERE value_1 IN
                 (SELECT value_1
                    FROM master_4
                   WHERE value_1 IN
                           ('Waneta',
                            'Queenie',
                            'Shante',
                            'Barbera',
                            'Thalia',
                            'Kasie',
                            'Courtney',
                            'Pearlene',
                            'Hattie',
                            'Darrin',
                            'Marcellus',
                            'Kandi',
                            'Jenifer',
                            'Chanel',
                            'Aurelio',
                            'Felicia'))
        UNION
        SELECT COUNT (*) counter
          FROM master_4
         WHERE value_3 IN
                 (SELECT value_2
                    FROM master_4
                   WHERE value_2 IN
                           ('Waneta',
                            'Queenie',
                            'Shante',
                            'Barbera',
```

```
                              'Thalia',
                              'Kasie',
                              'Courtney',
                              'Pearlene',
                              'Hattie',
                              'Darrin',
                              'Marcellus',
                              'Kandi',
                              'Jenifer',
                              'Chanel',
                              'Aurelio',
                              'Felicia'))
        UNION
        SELECT COUNT (*) counter
          FROM master_4
         WHERE (value_1, value_2) IN
                  (SELECT value_3, value_3
                     FROM master_4
                    WHERE value_3 IN
                             ('Waneta',
                              'Queenie',
                              'Shante',
                              'Barbera',
                              'Thalia',
                              'Kasie',
                              'Courtney',
                              'Pearlene',
                              'Hattie',
                              'Darrin',
                              'Marcellus',
                              'Kandi',
                              'Jenifer',
                              'Chanel',
                              'Aurelio',
                              'Felicia'))
        UNION
        SELECT COUNT (*) counter
          FROM master_4
         WHERE date_2 BETWEEN (SELECT MIN (date_2) FROM master_4)
                          AND (SELECT MIN (date_2) + 35 FROM master_4)
        UNION
        SELECT COUNT (*) counter
          FROM master_4
         WHERE date_2 NOT BETWEEN (SELECT MIN (date_2) FROM master_4)
                              AND (SELECT MIN (date_2) + 50 FROM
master_4)
        UNION
        SELECT COUNT (*) counter
          FROM master_4
         WHERE date_3 BETWEEN TO_DATE ('01-01-2000', 'DD-MM-YYYY')
                          AND SYSDATE - 25)
/
```

```
exit
/

--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_4_QUERY_2%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
--INDEXES
MASTER_4_CHAR_1_IDX
MASTER_4_CHAR_2_IDX
MASTER_4_CHAR_3_IDX
MASTER_4_DATE_1_IDX
MASTER_4_DATE_2_IDX
MASTER_4_DATE_3_IDX
MASTER_4_VALUE_1_IDX
MASTER_4_VALUE_2_IDX
MASTER_4_VALUE_3_IDX

--GET_COST
delete from plan_table
/
explain plan
set statement_id = 'A' for
SELECT COUNT (*)
  FROM Master_4
 WHERE     value_1 IN
              (SELECT value_1
                 FROM master_4
                WHERE value_1 IN
                         ('Jenee',
                          'Hassie',
                          'Andrea',
                          'Danyell',
                          'Laree',
                          'Juliette',
                          'Eliene',
                          'Marie',
                          'Terry',
                          'Susie',
                          'Tina',
                          'Kareem',
                          'Amie',
```

```
                        'Levi',
                        'Eddie',
                        'Fleta'))
       AND value_2 IN
              (SELECT value_2
                 FROM MASTER_4
                WHERE VALUE_2 NOT IN
                       ('Jenee',
                        'Hassie',
                        'Andrea',
                        'Danyell',
                        'Laree',
                        'Juliette',
                        'Eliene',
                        'Marie',
                        'Terry',
                        'Susie',
                        'Tina',
                        'Kareem',
                        'Amie',
                        'Levi',
                        'Eddie',
                        'Fleta'))
       AND value_3 IN (SELECT value_3
                        FROM master_4
                       WHERE SUBSTR (value_3, 1, 1) IN ('A', 'B',
'C'))
       AND date_1 IN
              (SELECT date_1
                 FROM master_4
                WHERE date_1 BETWEEN TO_DATE ('01-JAN-2001', 'DD-MON-
YYYY')
                                 AND TO_DATE ('02-MAR-2002', 'DD-MON-
YYYY')
               UNION
               SELECT date_2
                 FROM master_4
                WHERE date_2 NOT BETWEEN TO_DATE ('01-JAN-2001',
                                                  'DD-MON-YYYY')
                                     AND TO_DATE ('02-MAR-2002',
                                                  'DD-MON-YYYY')
               UNION
               SELECT date_3
                 FROM master_4
                WHERE date_3 = (SELECT MIN (date_3) + 35 FROM
master_4))
       AND char_1 IN ('A', 'B', 'C')
       AND char_2 IN ('1', '2', '3', '4', '5', '6', '7', '8')
       AND char_3 NOT IN
              (SELECT char_3
                 FROM master_4
                WHERE char_3 NOT IN ('A', 'B', 'C', '1', '2', '3'))
```

```
/
select cost
from plan_table
where statement_id = 'A'
and parent_id IS NULL
/
exit
/

--TIME_COST
set timing on
SELECT /*+ MASTER_4_QUERY_3*/
COUNT (*)
  FROM Master_4
 WHERE     value_1 IN
              (SELECT value_1
                 FROM master_4
                WHERE value_1 IN
                         ('Jenee',
                          'Hassie',
                          'Andrea',
                          'Danyell',
                          'Laree',
                          'Juliette',
                          'Eliene',
                          'Marie',
                          'Terry',
                          'Susie',
                          'Tina',
                          'Kareem',
                          'Amie',
                          'Levi',
                          'Eddie',
                          'Fleta'))
       AND value_2 IN
              (SELECT value_2
                 FROM MASTER_4
                WHERE VALUE_2 NOT IN
                         ('Jenee',
                          'Hassie',
                          'Andrea',
                          'Danyell',
                          'Laree',
                          'Juliette',
                          'Eliene',
                          'Marie',
                          'Terry',
                          'Susie',
                          'Tina',
                          'Kareem',
                          'Amie',
                          'Levi',
```

```
                          'Eddie',
                          'Fleta'))
       AND value_3 IN (SELECT value_3
                         FROM master_4
                        WHERE SUBSTR (value_3, 1, 1) IN ('A', 'B',
'C'))
       AND date_1 IN
              (SELECT date_1
                 FROM master_4
                WHERE date_1 BETWEEN TO_DATE ('01-JAN-2001', 'DD-MON-
YYYY')
                                 AND TO_DATE ('02-MAR-2002', 'DD-MON-
YYYY')
               UNION
               SELECT date_2
                 FROM master_4
                WHERE date_2 NOT BETWEEN TO_DATE ('01-JAN-2001',
                                                  'DD-MON-YYYY')
                                     AND TO_DATE ('02-MAR-2002',
                                                  'DD-MON-YYYY')
               UNION
               SELECT date_3
                 FROM master_4
                WHERE date_3 = (SELECT MIN (date_3) + 35 FROM
master_4))
       AND char_1 IN ('A', 'B', 'C')
       AND char_2 IN ('1', '2', '3', '4', '5', '6', '7', '8')
       AND char_3 NOT IN
              (SELECT char_3
                 FROM master_4
                WHERE char_3 NOT IN ('A', 'B', 'C', '1', '2', '3'))
/
exit
/
--CROSS_CHECK_COST
set heading off
set feedback off
set pagesize 0
select optimizer_cost
from v$sql
where sql_text like '%MASTER_4_QUERY_3%'
and optimizer_cost > 1
order by last_load_time desc
/
exit
/
----------------------------------------------------------------------
-------------------------------------------
alter index M2_L2_TAB2_PK_IDX invisible;
alter index M1_L2_T2_NUM_1_IDX invisible;
alter index M1_L2_T2_NUM_2_IDX invisible;
alter index M1_L2_T2_NUM_3_IDX invisible;
```

```
alter index M1_L2_T2_NUM_4_IDX invisible;
alter index M1_L2_T2_NUM_5_IDX invisible;
alter index M1_L2_T2_NUM_6_IDX invisible;
alter index M1_L2_T2_NUM_7_IDX invisible;
alter index M1_L2_T2_NUM_8_IDX invisible;
alter index M1_L2_T2_NUM_9_IDX invisible;
alter index M1_L2_T2_NUM_10_IDX invisible;
alter index M2_L2_T1_DATE_5_IDX invisible;
alter index M2_L2_T1_DATE_6_IDX invisible;
alter index M2_L2_T1_DATE_7_IDX invisible;
alter index M2_L2_T1_DATE_8_IDX invisible;
alter index M2_L2_T1_DATE_9_IDX invisible;
alter index M2_L2_T1_DATE_10_IDX invisible;
alter index M2_L2_T1_PK_IDX invisible;
alter index M2_L2_T1_DATE_1_IDX invisible;
alter index M2_L2_T1_DATE_2_IDX invisible;
alter index M2_L2_T1_DATE_3_IDX invisible;
alter index M2_L2_T1_DATE_4_IDX invisible;
alter index M1_L3_T3_CHAR_1_IDX invisible;
alter index M1_L3_T3_CHAR_2_IDX invisible;
alter index M1_L3_T3_DATE_1_IDX invisible;
alter index M1_L3_T3_DATE_2_IDX invisible;
alter index M1_L3_T3_DATE_3_IDX invisible;
alter index M1_L3_T3_DATE_4_IDX invisible;
alter index M1_L3_T3_DATE_5_IDX invisible;
alter index M1_L3_T3_PK_IDX invisible;
alter index M1_L3_T3_VALUE_1_IDX invisible;
alter index M1_L3_T3_VALUE_2_IDX invisible;
alter index M1_L3_T3_VALUE_3_IDX invisible;
alter index M1_L3_T2_CHAR_1_IDX invisible;
alter index M1_L3_T2_CHAR_2_IDX invisible;
alter index M1_L3_T2_DATE_1_IDX invisible;
alter index M1_L3_T2_DATE_2_IDX invisible;
alter index M1_L3_T2_DATE_3_IDX invisible;
alter index M1_L3_T2_DATE_4_IDX invisible;
alter index M1_L3_T2_DATE_5_IDX invisible;
alter index M1_L3_T2_PK_IDX invisible;
alter index M1_L3_T2_VALUE_1_IDX invisible;
alter index M1_L3_T2_VALUE_2_IDX invisible;
alter index M1_L3_T2_VALUE_3_IDX invisible;
alter index M1_L3_T1_PK_IDX invisible;
alter index M1_L3_T1_DATE_1_IDX invisible;
alter index M1_L3_T1_DATE_2_IDX invisible;
alter index M1_L3_T1_DATE_3_IDX invisible;
alter index M1_L3_T1_DATE_4_IDX invisible;
alter index M1_L3_T1_DATE_5_IDX invisible;
alter index M1_L3_T1_VALUE_1_IDX invisible;
alter index M1_L3_T1_VALUE_2_IDX invisible;
alter index M1_L3_T1_VALUE_3_IDX invisible;
alter index M1_L3_T1_CHAR_1_IDX invisible;
alter index M1_L3_T1_CHAR_2_IDX invisible;
alter index M1_L2_T3_CHAR_1_IDX invisible;
```

```
alter index M1_L2_T3_CHAR_2_IDX invisible;
alter index M1_L2_T3_CHAR_3_IDX invisible;
alter index M1_L2_T3_DATE_1_IDX invisible;
alter index M1_L2_T3_DATE_2_IDX invisible;
alter index M1_L2_T3_DATE_3_IDX invisible;
alter index M1_L2_T3_PK_IDX invisible;
alter index M1_L2_T3_VALUE_1_IDX invisible;
alter index M1_L2_T3_VALUE_2_IDX invisible;
alter index M1_L2_T3_VALUE_3_IDX invisible;
alter index M1_L2_T2_CHAR_1_IDX invisible;
alter index M1_L2_T2_CHAR_2_IDX invisible;
alter index M1_L2_T2_CHAR_3_IDX invisible;
alter index M1_L2_T2_DATE_1_IDX invisible;
alter index M1_L2_T2_DATE_2_IDX invisible;
alter index M1_L2_T2_DATE_3_IDX invisible;
alter index M1_L2_T2_PK_IDX invisible;
alter index M1_L2_T2_VALUE_1_IDX invisible;
alter index M1_L2_T2_VALUE_2_IDX invisible;
alter index M1_L2_T2_VALUE_3_IDX invisible;
alter index M1_L2_T1_PK_IDX invisible;
alter index M1_L2_T1_DATE_1_IDX invisible;
alter index M1_L2_T1_DATE_2_IDX invisible;
alter index M1_L2_T1_DATE_3_IDX invisible;
alter index M1_L2_T1_CHAR_1_IDX invisible;
alter index M1_L2_T1_CHAR_2_IDX invisible;
alter index M1_L2_T1_CHAR_3_IDX invisible;
alter index M1_L2_T1_VALUE_1_IDX invisible;
alter index M1_L2_T1_VALUE_2_IDX invisible;
alter index M1_L2_T1_VALUE_3_IDX invisible;
alter index MASTER_3_PK_IDX invisible;
alter index MASTER_3_NUM_1_IDX invisible;
alter index MASTER_3_NUM_2_IDX invisible;
alter index MASTER_3_NUM_3_IDX invisible;
alter index MASTER_3_NUM_4_IDX invisible;
alter index MASTER_3_NUM_5_IDX invisible;
alter index MASTER_3_CHAR_2_IDX invisible;
alter index MASTER_3_CHAR_1_IDX invisible;
alter index MASTER_3_CHAR_3_IDX invisible;
alter index MASTER_3_VALUE_1_IDX invisible;
alter index MASTER_3_VALUE_2_IDX invisible;
alter index MASTER_3_LINKER invisible;
alter index MASTER_2_PK_IDX invisible;
alter index MASTER_2_VALUE_1_IDX invisible;
alter index MASTER_2_VALUE_2_IDX invisible;
alter index MASTER_2_VALUE_3_IDX invisible;
alter index MASTER_2_VALUE_4_IDX invisible;
alter index MASTER_2_VALUE_5_IDX invisible;
alter index MASTER_2_CHAR_1_IDX invisible;
alter index MASTER_2_CHAR_2_IDX invisible;
alter index MASTER_2_CHAR_3_IDX invisible;
alter index MASTER_2_DATE_1_IDX invisible;
alter index MASTER_2_DATE_2_IDX invisible;
```

```
alter index MASTER_2_LINKER invisible;
alter index MASTER_1_PK_IDX invisible;
alter index MASTER_1_DATE_2_IDX invisible;
alter index MASTER_1_DATE_3_IDX invisible;
alter index MASTER_1_DATE_4_IDX invisible;
alter index MASTER_1_DATE_5_IDX invisible;
alter index MASTER_1_DATE_1_IDX invisible;
alter index MASTER_1_CHAR_1_IDX invisible;
alter index MASTER_1_CHAR_2_IDX invisible;
alter index MASTER_1_CHAR_3_IDX invisible;
alter index MASTER_1_VALUE_1_IDX invisible;
alter index MASTER_1_VALUE_2_IDX invisible;
alter index MASTER_1_LINKER invisible;
alter index M3_L3_T3_CHAR_1_IDX invisible;
alter index M3_L3_T3_CHAR_2_IDX invisible;
alter index M3_L3_T3_CHAR_3_IDX invisible;
alter index M3_L3_T3_DATE_1_IDX invisible;
alter index M3_L3_T3_DATE_2_IDX invisible;
alter index M3_L3_T3_DATE_3_IDX invisible;
alter index M3_L3_T3_DATE_4_IDX invisible;
alter index M3_L3_T3_PK_IDX invisible;
alter index M3_L3_T3_VALUE_1_IDX invisible;
alter index M3_L3_T3_VALUE_2_IDX invisible;
alter index M3_L3_T3_VALUE_3_IDX invisible;
alter index M3_L3_T2_CHAR_1_IDX invisible;
alter index M3_L3_T2_CHAR_2_IDX invisible;
alter index M3_L3_T2_CHAR_3_IDX invisible;
alter index M3_L3_T2_DATE_1_IDX invisible;
alter index M3_L3_T2_DATE_2_IDX invisible;
alter index M3_L3_T2_DATE_3_IDX invisible;
alter index M3_L3_T2_DATE_4_IDX invisible;
alter index M3_L3_T2_PK_IDX invisible;
alter index M3_L3_T2_VALUE_1_IDX invisible;
alter index M3_L3_T2_VALUE_2_IDX invisible;
alter index M3_L3_T2_VALUE_3_IDX invisible;
alter index M3_L3_T1_PK_IDX invisible;
alter index M3_L3_T1_CHAR_1_IDX invisible;
alter index M3_L3_T1_CHAR_2_IDX invisible;
alter index M3_L3_T1_CHAR_3_IDX invisible;
alter index M3_L3_T1_VALUE_1_IDX invisible;
alter index M3_L3_T1_VALUE_2_IDX invisible;
alter index M3_L3_T1_VALUE_3_IDX invisible;
alter index M3_L3_T1_DATE_1_IDX invisible;
alter index M3_L3_T1_DATE_2_IDX invisible;
alter index M3_L3_T1_DATE_3_IDX invisible;
alter index M3_L3_T1_DATE_4_IDX invisible;
alter index M3_L2_T3_COMP_1_IDX invisible;
alter index M3_L2_T3_DATE_COMP_1_IDX invisible;
alter index M3_L2_T3_NUM_COMP_1_IDX invisible;
alter index M3_L2_T3_PK_IDX invisible;
alter index M3_L2_T2_COMP_1_IDX invisible;
alter index M3_L2_T2_DATE_COMP_1_IDX invisible;
```

```
alter index M3_L2_T2_NUM_COMP_1_IDX invisible;
alter index M3_L2_T2_PK_IDX invisible;
alter index M3_L2_TB_PK_IDX invisible;
alter index M3_L2_T1_COMP_1_IDX invisible;
alter index M3_L2_T1_NUM_COMP_1_IDX invisible;
alter index M3_L2_T1_DATE_COMP_1_IDX invisible;
alter index M2_L3_T3_PK_IDX invisible;
alter index M2_L3_T3_CHAR_1_IDX invisible;
alter index M2_L3_T3_CHAR_2_IDX invisible;
alter index M2_L3_T3_CHAR_3_IDX invisible;
alter index M2_L3_T3_DATE_1_IDX invisible;
alter index M2_L3_T3_DATE_2_IDX invisible;
alter index M2_L3_T3_DATE_3_IDX invisible;
alter index M2_L3_T3_VALUE_1_IDX invisible;
alter index M2_L3_T3_VALUE_2_IDX invisible;
alter index M2_L3_T3_VALUE_3_IDX invisible;
alter index M2_L3_T3_VALUE_4_IDX invisible;
alter index M2_L3_T2_PK_IDX invisible;
alter index M2_L3_T2_CHAR_1_IDX invisible;
alter index M2_L3_T2_CHAR_2_IDX invisible;
alter index M2_L3_T2_CHAR_3_IDX invisible;
alter index M2_L3_T2_DATE_1_IDX invisible;
alter index M2_L3_T2_DATE_2_IDX invisible;
alter index M2_L3_T2_DATE_3_IDX invisible;
alter index M2_L3_T2_VALUE_1_IDX invisible;
alter index M2_L3_T2_VALUE_2_IDX invisible;
alter index M2_L3_T2_VALUE_3_IDX invisible;
alter index M2_L3_T2_VALUE_4_IDX invisible;
alter index M2_L3_T1_PK_IDX invisible;
alter index M2_L3_T1_CHAR_1_IDX invisible;
alter index M2_L3_T1_CHAR_2_IDX invisible;
alter index M2_L3_T1_CHAR_3_IDX invisible;
alter index M2_L3_T1_DATE_1_IDX invisible;
alter index M2_L3_T1_DATE_2_IDX invisible;
alter index M2_L3_T1_DATE_3_IDX invisible;
alter index M2_L3_T1_VALUE_1_IDX invisible;
alter index M2_L3_T1_VALUE_2_IDX invisible;
alter index M2_L3_T1_VALUE_3_IDX invisible;
alter index M2_L3_T1_VALUE_4_IDX invisible;
alter index M2_L2_T3_PK_IDX invisible;
alter index M2_L2_T3_VALUE_1_IDX invisible;
alter index M2_L2_T3_VALUE_2_IDX invisible;
alter index M2_L2_T3_VALUE_3_IDX invisible;
alter index M2_L2_T3_VALUE_4_IDX invisible;
alter index M2_L2_T3_VALUE_5_IDX invisible;
alter index M2_L2_T3_VALUE_6_IDX invisible;
alter index M2_L2_T3_VALUE_7_IDX invisible;
alter index M2_L2_T3_VALUE_8_IDX invisible;
alter index M2_L2_T3_VALUE_9_IDX invisible;
alter index M2_L2_T3_VALUE_10_IDX invisible;
alter index MASTER_4_CHAR_1_IDX invisible;
alter index MASTER_4_CHAR_2_IDX invisible;
```

```
alter index MASTER_4_CHAR_3_IDX invisible;
alter index MASTER_4_DATE_1_IDX invisible;
alter index MASTER_4_DATE_2_IDX invisible;
alter index MASTER_4_DATE_3_IDX invisible;
alter index MASTER_4_VALUE_1_IDX invisible;
alter index MASTER_4_VALUE_2_IDX invisible;
alter index MASTER_4_VALUE_3_IDX invisible;
exit
/
----------------------------------------------------------------------
------------------------------------------
```

Appendix 4

```
#!/usr/bin/perl
#-----------------Set Global Values-----------------------------
#---------------------------------------------------------------

$command_file     =
"\/export\/home\/oracle\/perl_scripts\/command1.sql";
$cost_capture_dir = "\/export\/home\/oracle\/perl_scripts\/";
$output_file      =
"\/export\/home\/oracle\/perl_scripts\/stat_data.txt";
@query_names = ('GENERAL_JOIN_2.sql', 'GENERAL_JOIN_1.sql',
'GENERAL_JOIN_3.sql',
                'MASTER_1_JOIN_3.sql', 'MASTER_1_QUERY.sql',
'MASTER_2_QUERY.sql', 'MASTER_3_QUERY.sql',
                'MASTER_ALL_JOIN_4.sql', 'MASTER_ALL_JOIN_5.sql',
'MASTER_ALL_JOIN_6.sql', 'MASTER_ALL_JOIN_7.sql',
                'GENERAL_JOIN_4.sql', 'GENERAL_JOIN_5.sql',
'GENERAL_JOIN_6.sql', 'GENERAL_JOIN_7.sql',
                'GENERAL_JOIN_8.sql', 'GENERAL_JOIN_9.sql',
'MASTER_1_JOIN_1.sql', 'MASTER_1_JOIN_2.sql',
                'MASTER_3_QUERY_2.sql', 'MASTER_ALL_JOIN_1.sql',
'MASTER_ALL_JOIN_2.sql', 'MASTER_ALL_JOIN_3.sql',

'MASTER_1_QUERY.sql','MASTER_ALL_JOIN_8.sql','MASTER_4_QUERY.sql','MAS
TER_4_QUERY_1.sql',
                'MASTER_4_QUERY_2.sql','MASTER_4_QUERY_3.sql');



#-------------------SUB ROUTINES--------------------------------
#---------------------------------------------------------------
sub initialize_nonfixed_globals {

   $index_count = 12;
   $total_combinations = 2 ** $index_count;
   @cost_array = ();

   $unique_cost_values = 1;
   @index_names = ();
   $get_cost_query = "";
   $time_cost_query = "";
   $cross_check_query = "";

   #--reset indexes
    $temp = sql("$cost_capture_dir" . "SQL_FILES\/RESET_INDEXES.sql");

}

sub sql {
```

```
   my $execute_file = shift;

   my $output = `sqlplus -S test_user\/test\@SANDBOX \@$execute_file`;
   return $output;
}

sub parse_sql_file {

   my $query_name = shift;
   my $index_flag = "false";
   my $get_cost_flag = "false";
   my $time_cost_flag = "false";
   my $cross_check_flag = "false";

   open (FILE,
"\/export\/home\/oracle\/perl_scripts\/SQL_FILES\/$query_name");
   my @file = <FILE>;
   close FILE;
  LOOP: foreach $line (@file){

      chomp($line);

      if ($line =~ /.*INDEXES.*/){

         $index_flag = "true";
         $get_cost_flag = "false";
         $time_cost_flag = "false";
         $cross_check_flag = "false";
         next LOOP;

      }
      elsif ($line =~ /.*GET_COST.*/){

         $index_flag = "false";
         $get_cost_flag = "true";
         $time_cost_flag = "false";
         $cross_check_flag = "false";
         next LOOP;

      }
      elsif ($line =~ /.*TIME_COST.*/){

         $index_flag = "false";
         $get_cost_flag = "false";
         $time_cost_flag = "true";
         $cross_check_flag = "false";
         next LOOP;

      }
      elsif ($line =~ /.*CROSS_CHECK_COST.*/){

         $index_flag = "false";
```

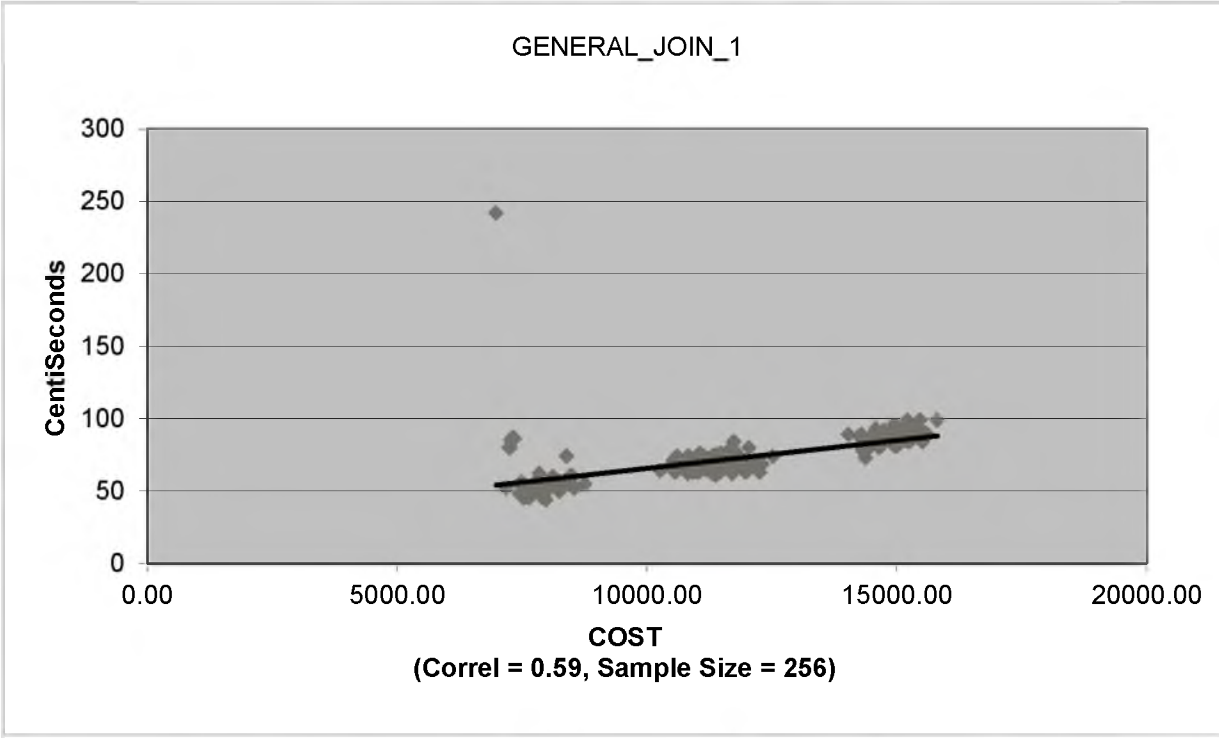
```
         $get_cost_flag = "false";
         $time_cost_flag = "false";
         $cross_check_flag = "true";
         next LOOP;

      }

      #--tank up some variables
      if ($index_flag eq "true"){

         push (@index_names, $line)

      }
      elsif ($get_cost_flag eq "true"){

         $get_cost_query .= "$line\n";

      }
      elsif ($time_cost_flag eq "true"){

         $time_cost_query .= "$line\n";

      }
      elsif ($cross_check_flag eq "true"){

         $cross_check_query .= "$line\n";
      }

   }

}

sub check_if_new_cost {

   my $test_cost = shift;
   my $new_value_flag = "TRUE";

   foreach $cost_value (@cost_array){

      if ($cost_value == $test_cost){
         $new_value_flag = "FALSE";
      }
   }

   if ($new_value_flag eq "TRUE"){

      push (@cost_array, $test_cost);
   }

   return $new_value_flag;
}
```

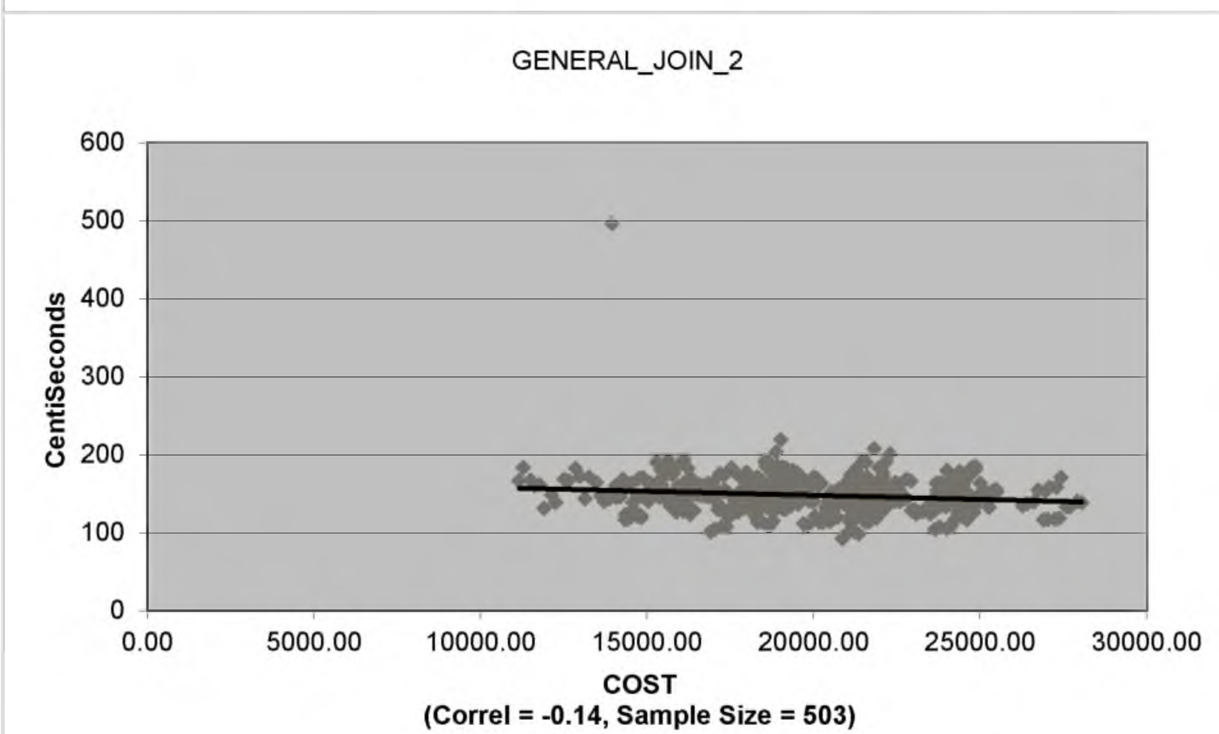
```
sub set_index_values_and_process_query {

   my $query_name = shift;

   #--This will set the indexes on and off according to how many we
want to work with,....
   #--and according to how many there are in the table.

   if (($#index_names) < $index_count){

      $index_count = $#index_names;
      $total_combinations = 2 ** $index_count;
   }


   for ($combinations = 0; $combinations < $total_combinations;
$combinations++){

      open (EXE, ">$command_file");
      print EXE "set heading off\n";
      print EXE "set feedback off\n";
      print EXE "set pagesize 0\n";


      for ($bit_index_check = 0; $bit_index_check < $index_count;
$bit_index_check++){

         $test_value = 2 ** $bit_index_check;

         if (($combinations & $test_value) != 0) {

            $command = qq{alter index $index_names[$bit_index_check]
invisible;};
            print EXE $command , "\n";

         }
         else{

            $command = qq{alter index $index_names[$bit_index_check]
visible;};
            print EXE $command , "\n";

         }
      }

    #--put the query on the end of the index changes
    print EXE "\n$get_cost_query\n";
    close EXE;


    chomp($cost = sql($command_file));
```

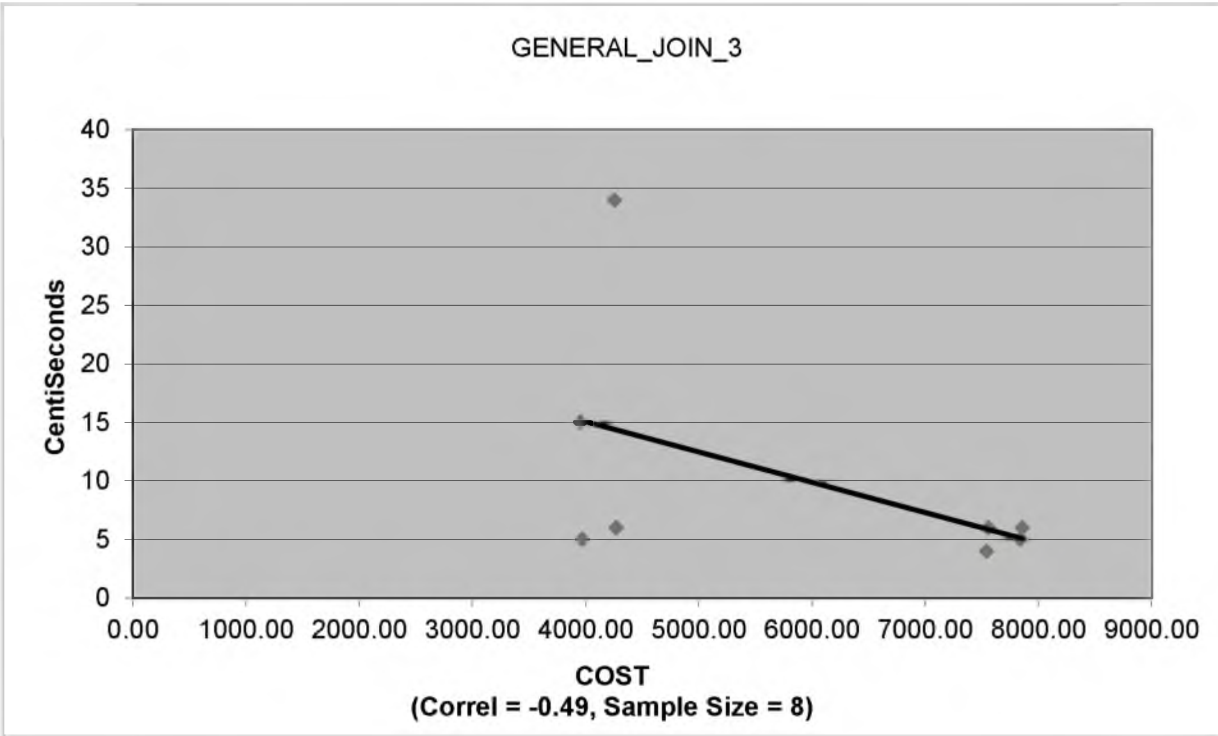
```
    $new_value = check_if_new_cost($cost);

    if ($new_value eq "TRUE"){

       #--Is a new cost, so need to run and find elapsed time.

       open (EXE, ">$command_file");
       print EXE "$time_cost_query";
       close EXE;

       $time = sql("$command_file");
       $time =~ /.*Elapsed: (.*:.*:.*.*).*/;
       chomp($time = $1);

       #--now, need to see if this was the same cost that was used
during execution.

       open (EXE, ">$command_file");
       print EXE $cross_check_query;
       close EXE;
       chomp($run_cost = sql("$command_file"));
       $~ = COST_OUTPUT;

       #-----------------------------------------------------
       write();

       open (OUTPUT, ">>$output_file");
       print OUTPUT
"$unique_cost_values!~!$query_name!~!$combinations!~!$cost!~!$run_cost
!~!$time\n";
       close OUTPUT;

       $unique_cost_values++;

     }
   }
}

#----------MAIN SECTION---------------------------------

foreach $query_name (@query_names){

   initialize_nonfixed_globals();

   parse_sql_file($query_name);


   system ("cls");
   print "Unique Number        Query Name
Combination     Cost Value      Run Cost        Elapsed Time\n";
```

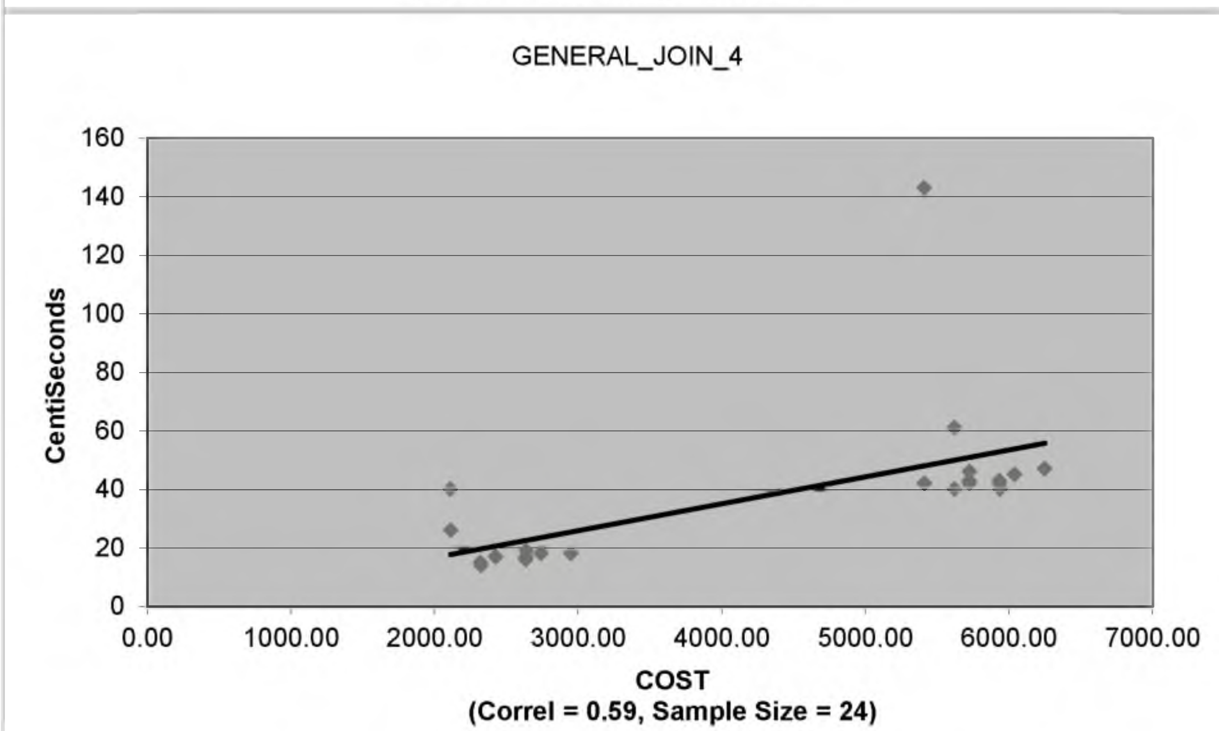
```
   set_index_values_and_process_query($query_name);

}

format COST_OUTPUT =
@<<<<<<
@<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<      @<<<<<<<<<
@<<<<<<<<<<<<<<<  @<<<<<<<<<<<<<<<  @<<<<<<<<<<<<<<<<
$unique_cost_values, $query_name,
$combinations,  $cost,          $run_cost,       $time
.
```

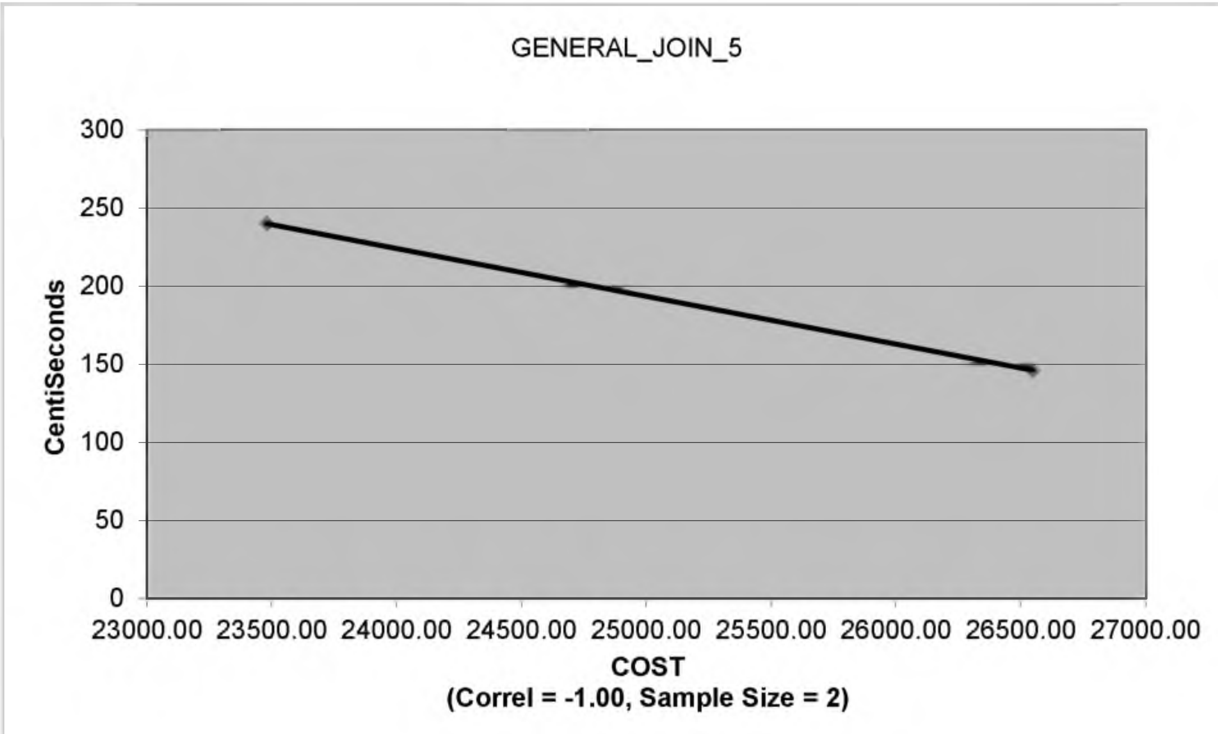
Appendix 5

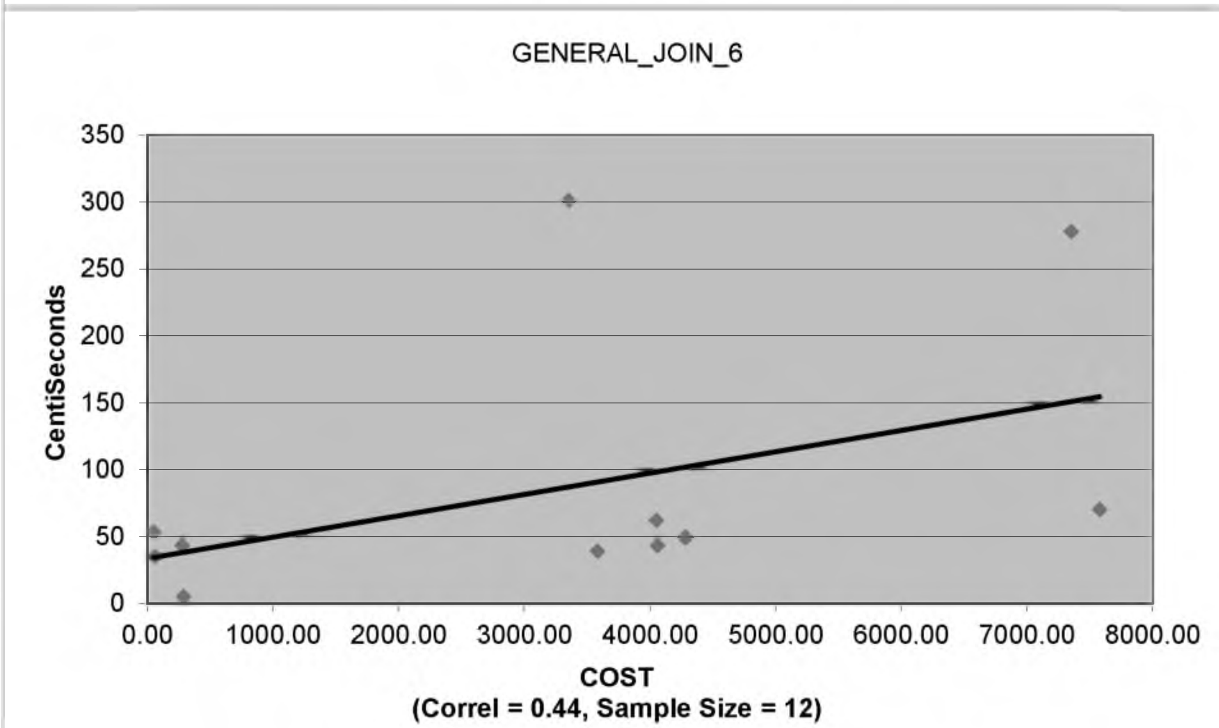

GENERAL_JOIN_1
CentiSeconds
300
250
200
150
100
50
0
0.00
5000.00
10000.00
15000.00
20000.00
COST
(Correl = 0.59, Sample Size = 256)

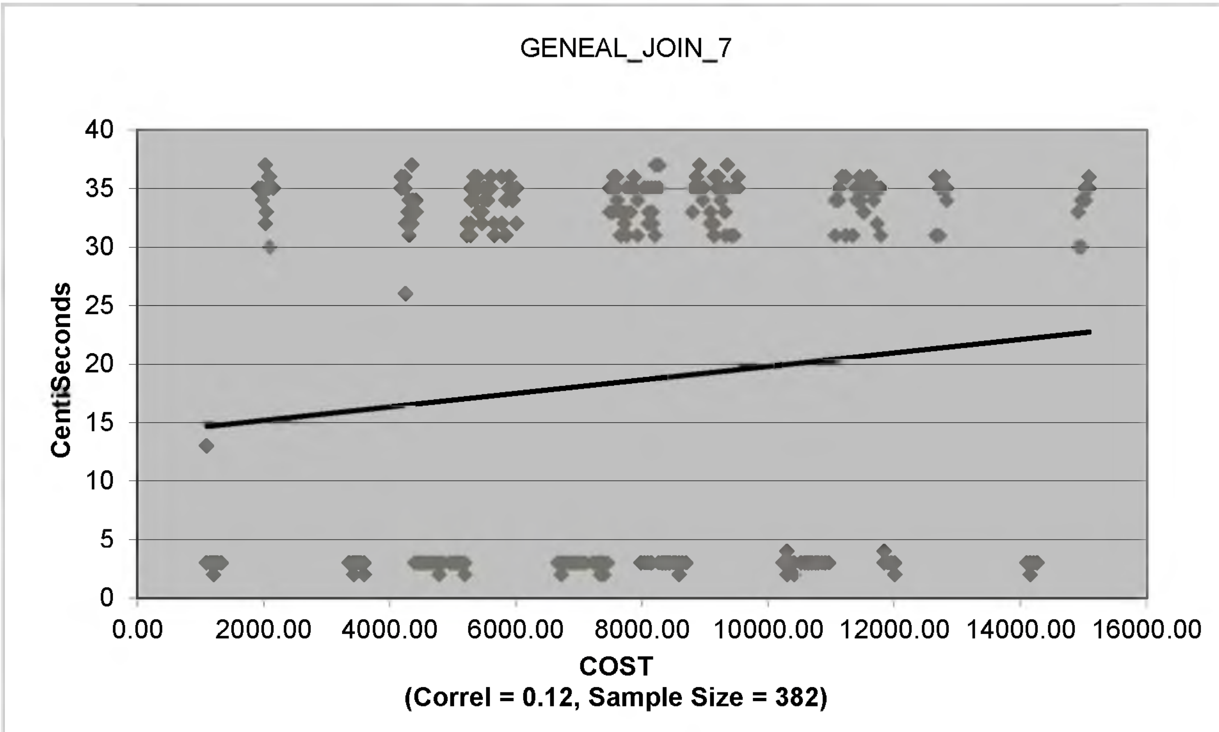

GENERAL_JOIN_2
CentiSeconds
600
500
400
300
200
100
0
0.00
5000.00
10000.00
15000.00
20000.00
25000.00
30000.00
COST
(Correl = -0.14, Sample Size = 503)

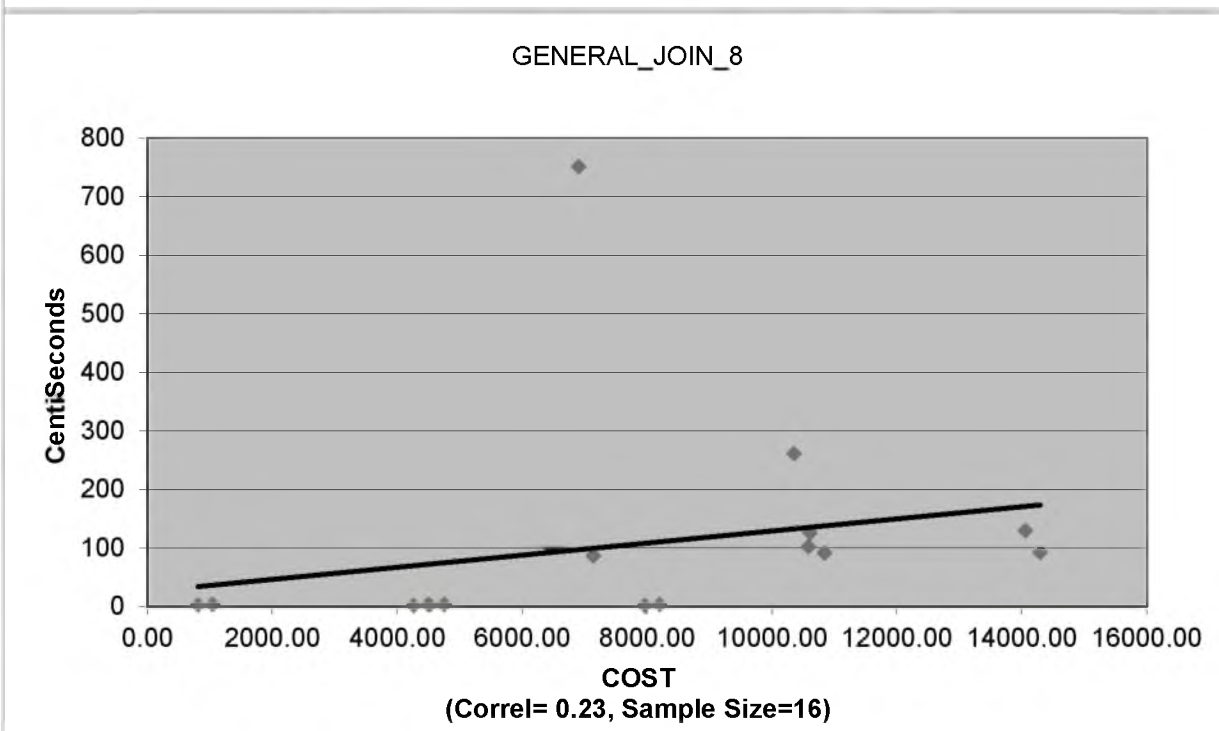

GENERAL_JOIN_3
40
35
30
25
20
15
10
5
0
CentiSeconds
0.00
1000.00
2000.00
3000.00
4000.00
5000.00
6000.00
7000.00
8000.00
9000.00
COST
(Correl = -0.49, Sample Size = 8)

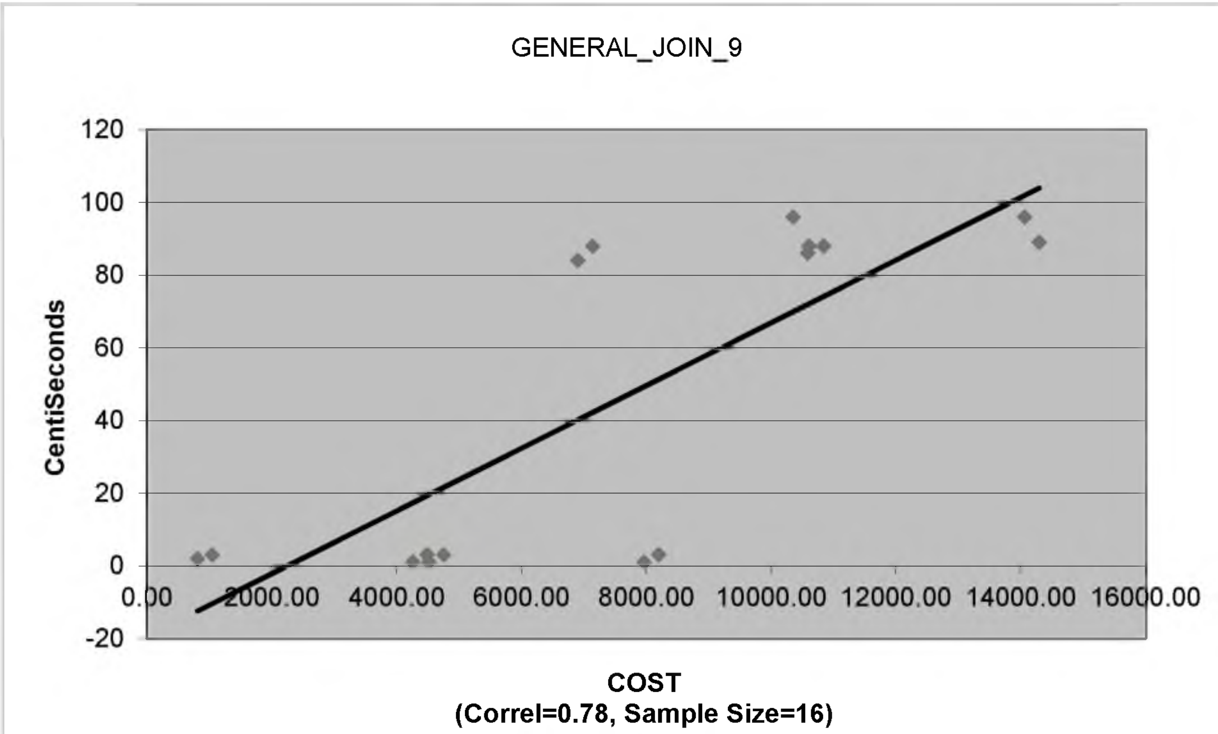

GENERAL_JOIN_4
160
140
120
100
80
60
40
20
0
CentiSeconds
0.00
1000.00
2000.00
3000.00
4000.00
5000.00
6000.00
7000.00
COST
(Correl = 0.59, Sample Size = 24)

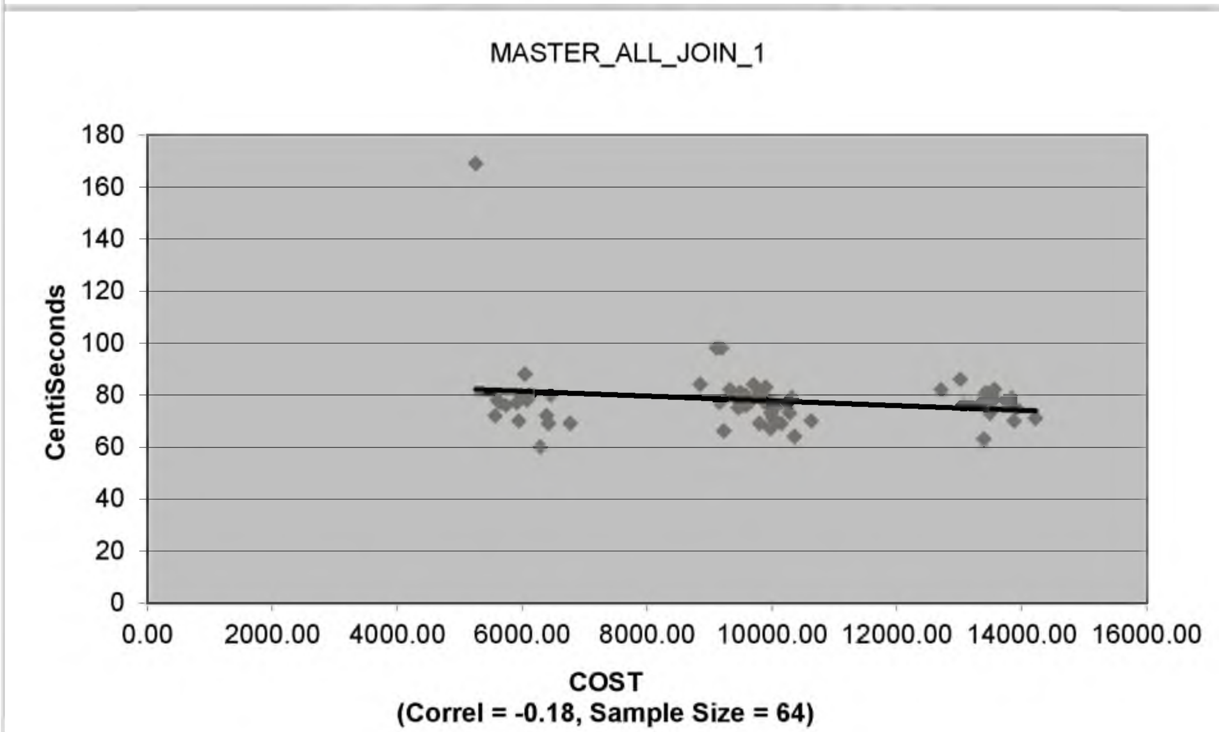

GENERAL_JOIN_5
CentiSeconds
300
250
200
150
100
50
0
23000.00 23500.00 24000.00 24500.00 25000.00 25500.00 26000.00 26500.00 27000.00
COST
(Correl = -1.00, Sample Size = 2)

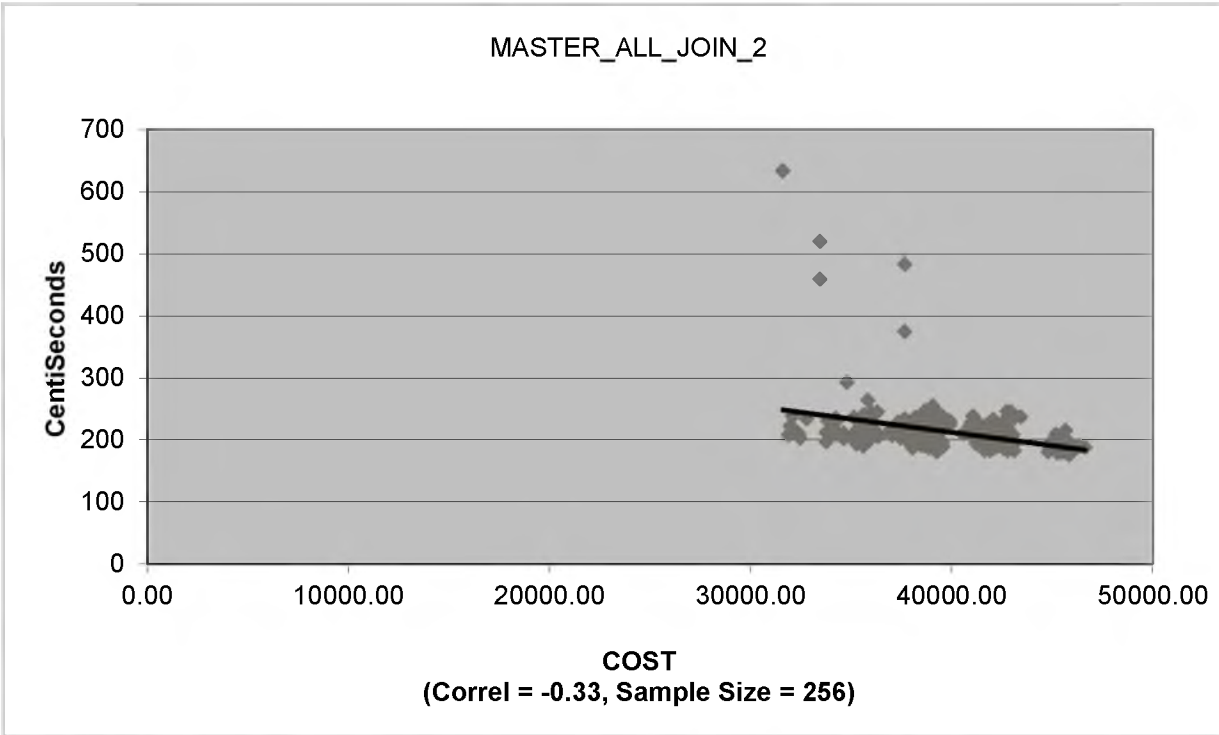

GENERAL_JOIN_6
CentiSeconds
350
300
250
200
150
100
50
0
0.00 1000.00 2000.00 3000.00 4000.00 5000.00 6000.00 7000.00 8000.00
COST
(Correl = 0.44, Sample Size = 12)

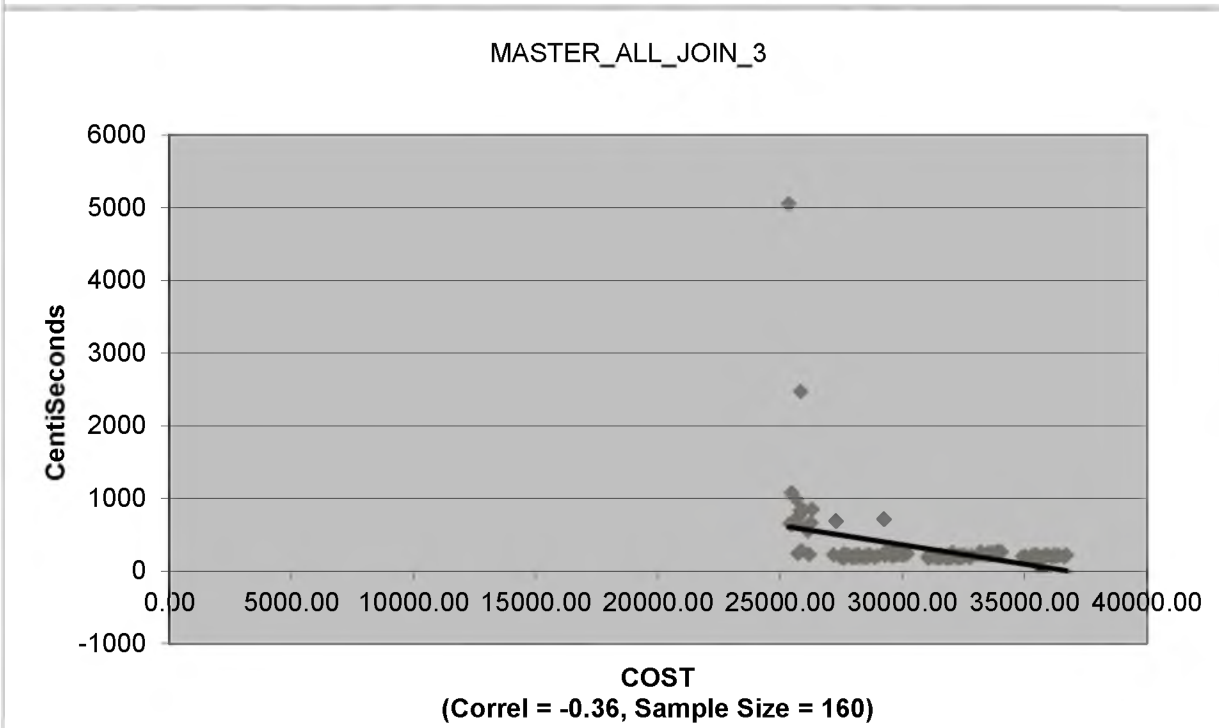

GENEAL_JOIN_7
40
35
30
25
20
15
10
5
0
CentiSeconds
0.00 2000.00 4000.00 6000.00 8000.00 10000.00 12000.00 14000.00 16000.00
COST
(Correl = 0.12, Sample Size = 382)

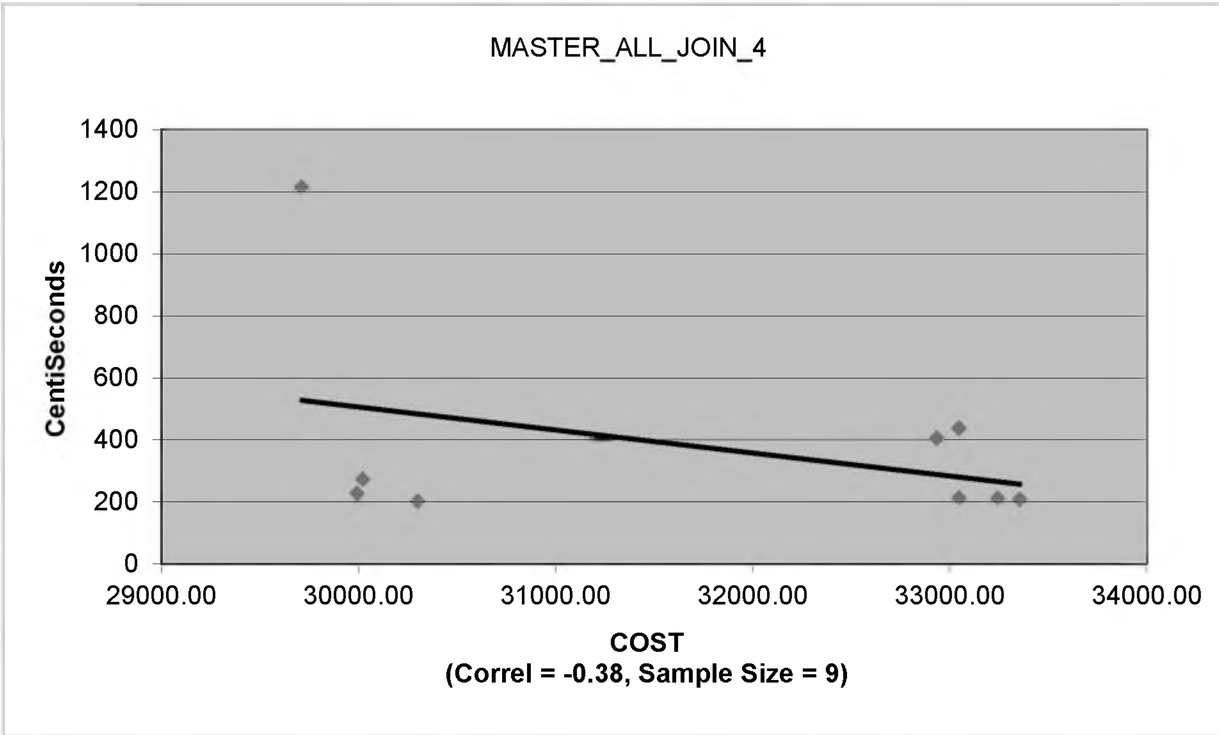

GENERAL_JOIN_8
800
700
600
500
400
300
200
100
0
CentiSeconds
0.00 2000.00 4000.00 6000.00 8000.00 10000.00 12000.00 14000.00 16000.00
COST
(Correl= 0.23, Sample Size=16)

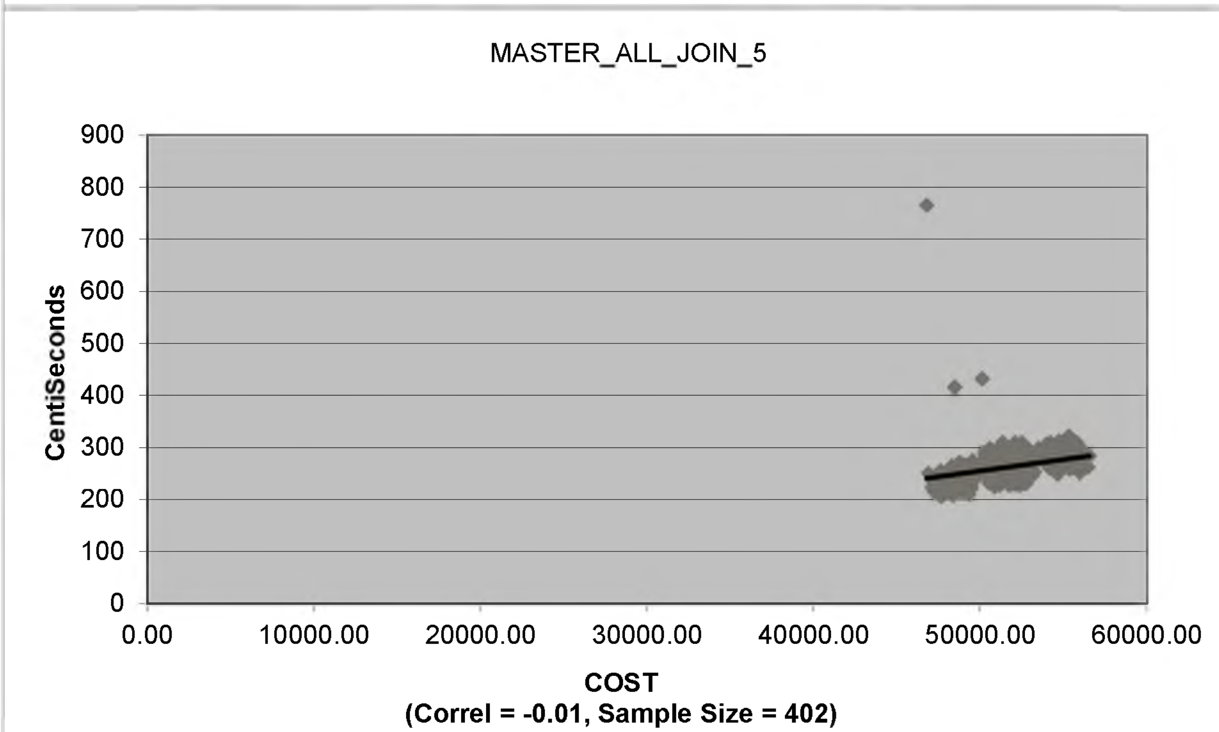

GENERAL_JOIN_9
120
100
80
60
40
20
0
-20
CentiSeconds
0.00
2000.00
4000.00
6000.00
8000.00
10000.00
12000.00
14000.00
16000.00
COST
(Correl=0.78, Sample Size=16)

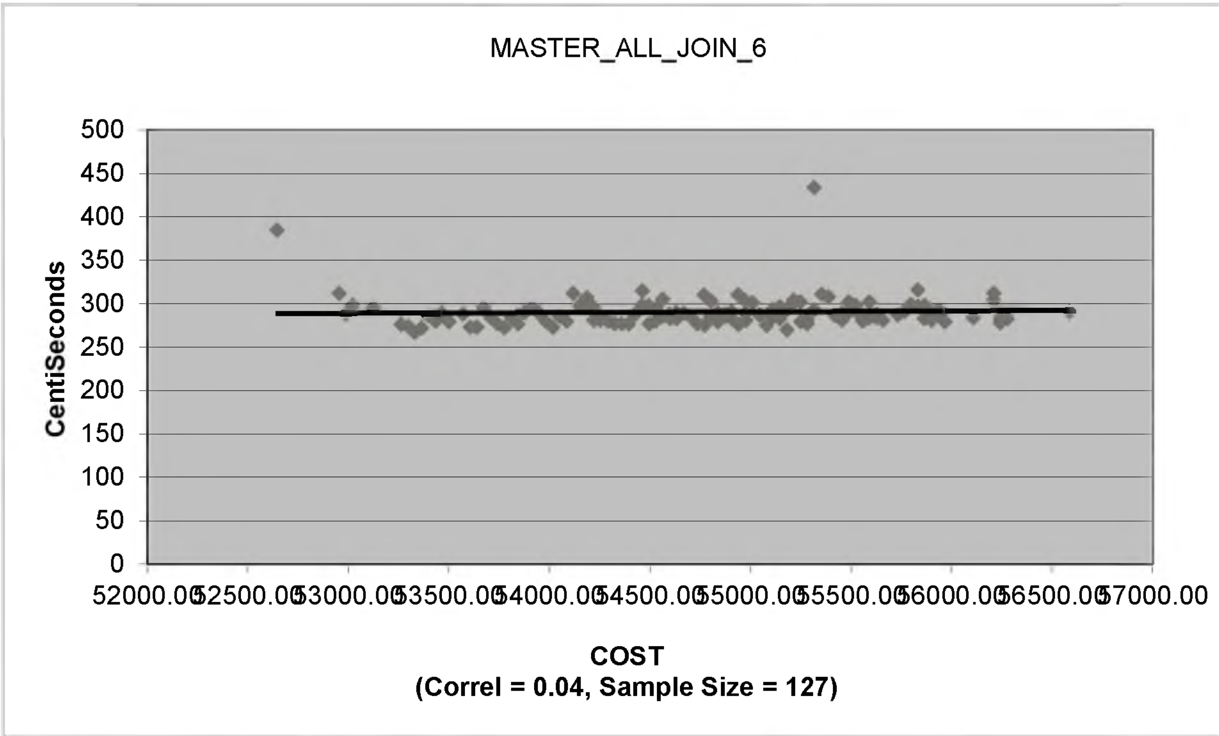

MASTER_ALL_JOIN_1
180
160
140
120
100
80
60
40
20
0
CentiSeconds
0.00
2000.00
4000.00
6000.00
8000.00
10000.00
12000.00
14000.00
16000.00
COST
(Correl = -0.18, Sample Size = 64)

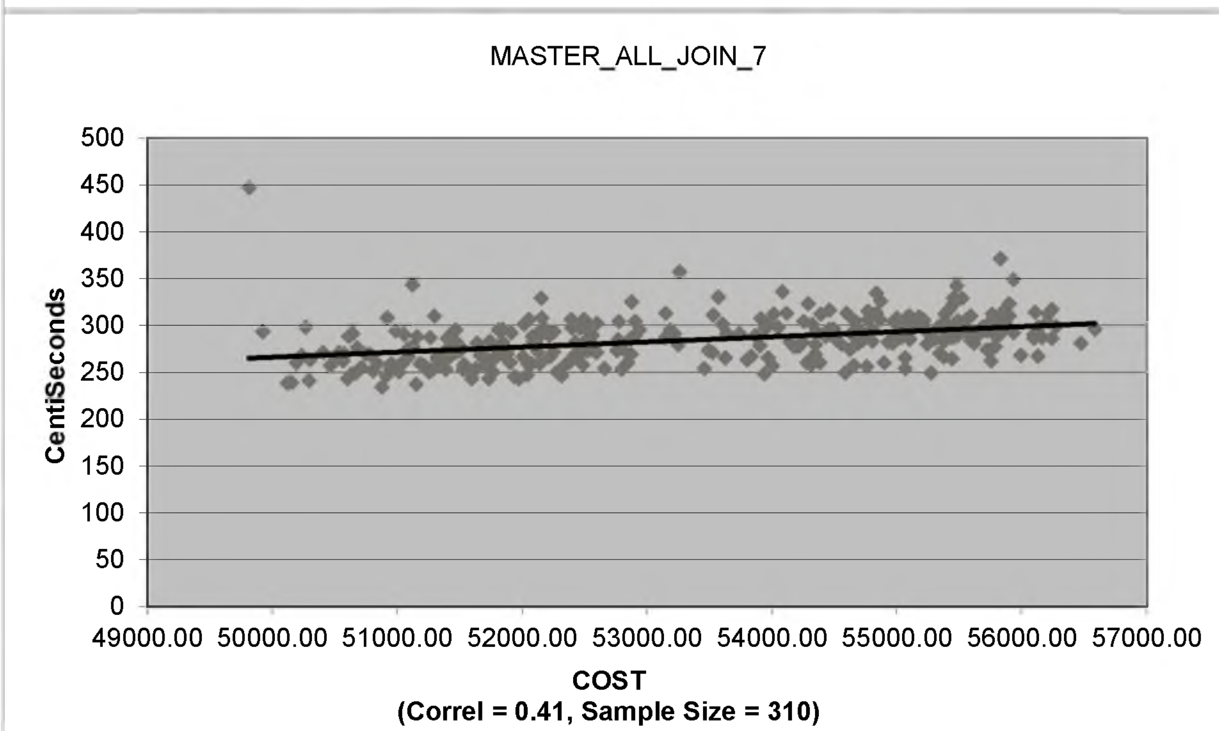

MASTER_ALL_JOIN_2
700
600
500
400
300
200
100
0
CentiSeconds
0.00
10000.00
20000.00
30000.00
40000.00
50000.00
COST
(Correl = -0.33, Sample Size = 256)

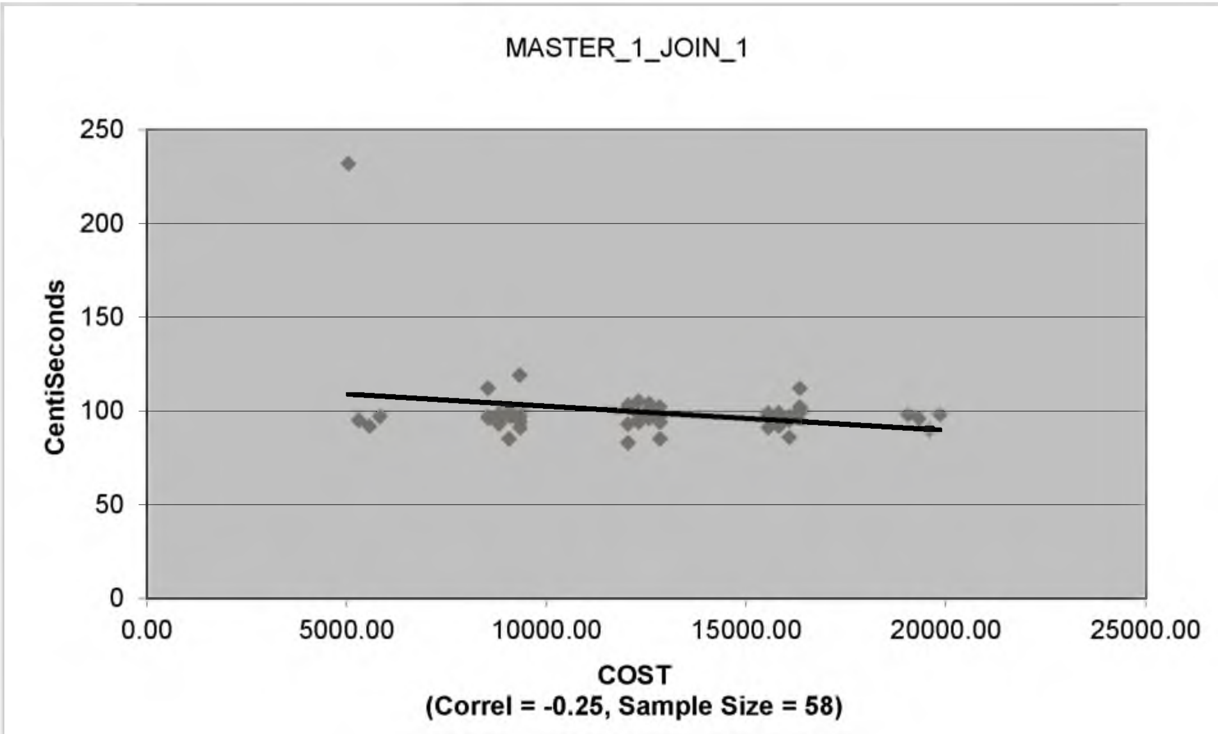

MASTER_ALL_JOIN_3
6000
5000
4000
3000
2000
1000
0
-1000
CentiSeconds
0.00
5000.00
10000.00
15000.00
20000.00
25000.00
30000.00
35000.00
40000.00
COST
(Correl = -0.36, Sample Size = 160)

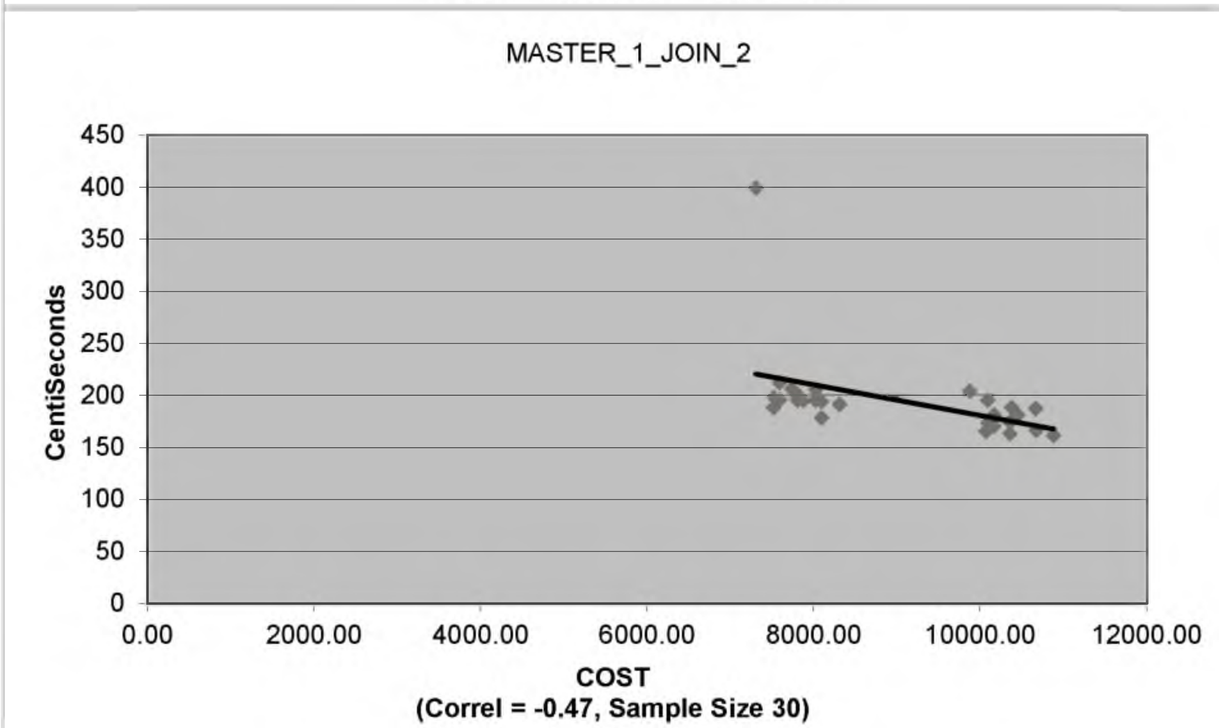

MASTER_ALL_JOIN_4
CentiSeconds
1400
1200
1000
800
600
400
200
0
29000.00
30000.00
31000.00
32000.00
33000.00
34000.00
COST
(Correl = -0.38, Sample Size = 9)

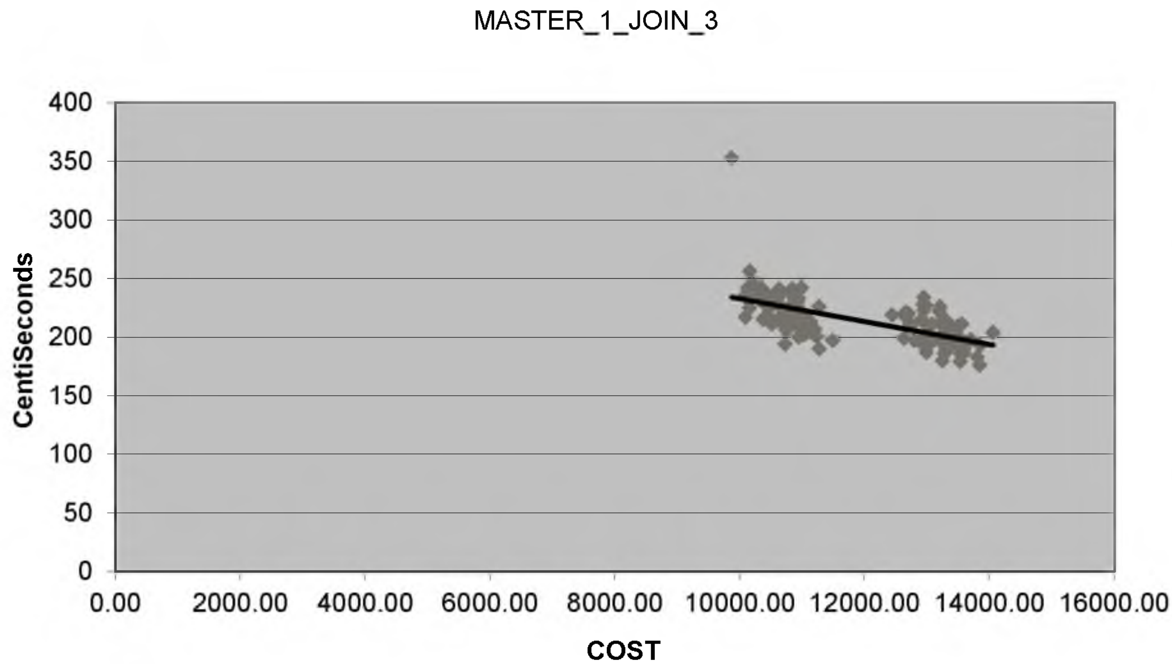

MASTER_ALL_JOIN_5
CentiSeconds
900
800
700
600
500
400
300
200
100
0
0.00
10000.00
20000.00
30000.00
40000.00
50000.00
60000.00
COST
(Correl = -0.01, Sample Size = 402)


MASTER_ALL_JOIN_6
CentiSeconds
500
450
400
350
300
250
200
150
100
50
0
52000.00 52500.00 53000.00 53500.00 54000.00 54500.00 55000.00 55500.00 56000.00 56500.00 57000.00
COST
(Correl = 0.04, Sample Size = 127)

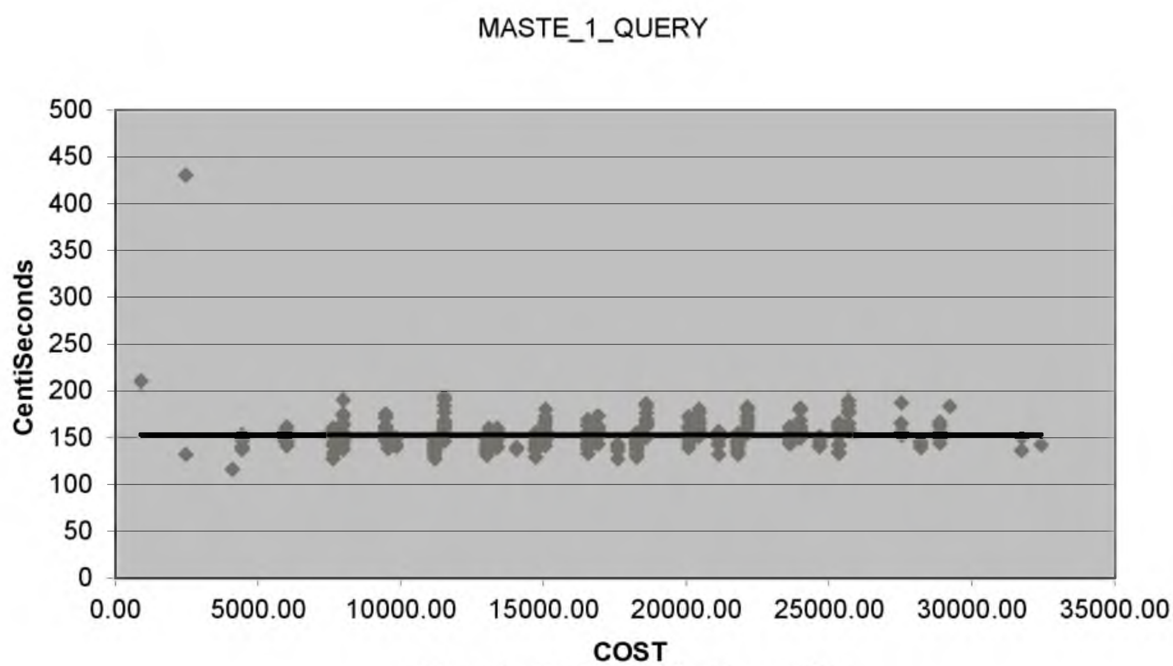

MASTER_ALL_JOIN_7
CentiSeconds
500
450
400
350
300
250
200
150
100
50
0
49000.00 50000.00 51000.00 52000.00 53000.00 54000.00 55000.00 56000.00 57000.00
COST
(Correl = 0.41, Sample Size = 310)


MASTER_1_JOIN_1
CentiSeconds
250
200
150
100
50
0
0.00
5000.00
10000.00
15000.00
20000.00
25000.00
COST
(Correl = -0.25, Sample Size = 58)

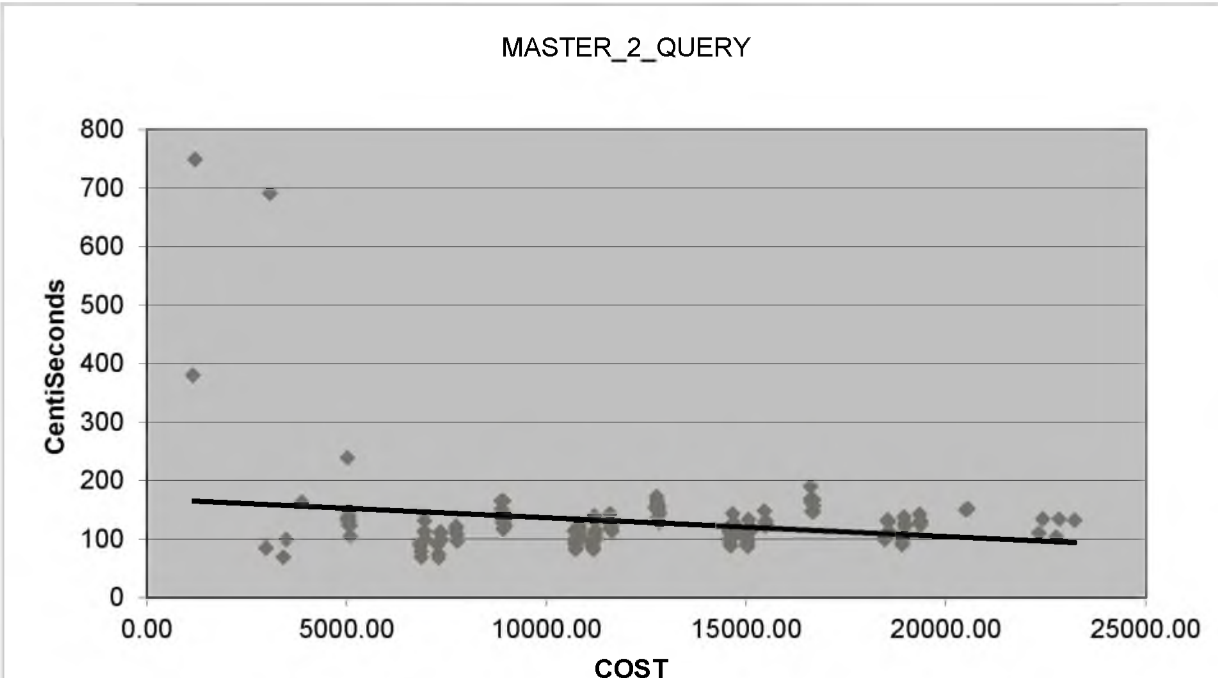

MASTER_1_JOIN_2
CentiSeconds
450
400
350
300
250
200
150
100
50
0
0.00
2000.00
4000.00
6000.00
8000.00
10000.00
12000.00
COST
(Correl = -0.47, Sample Size 30)



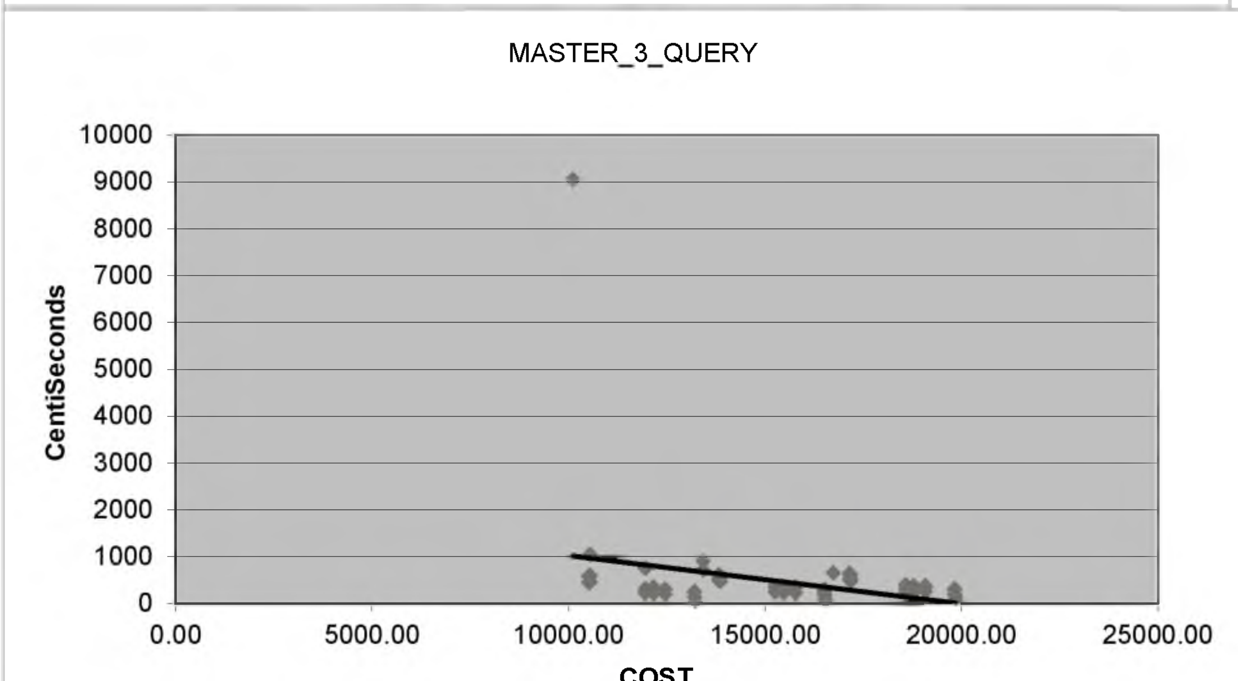
(Correl = -0.60, Sample Size = 119)



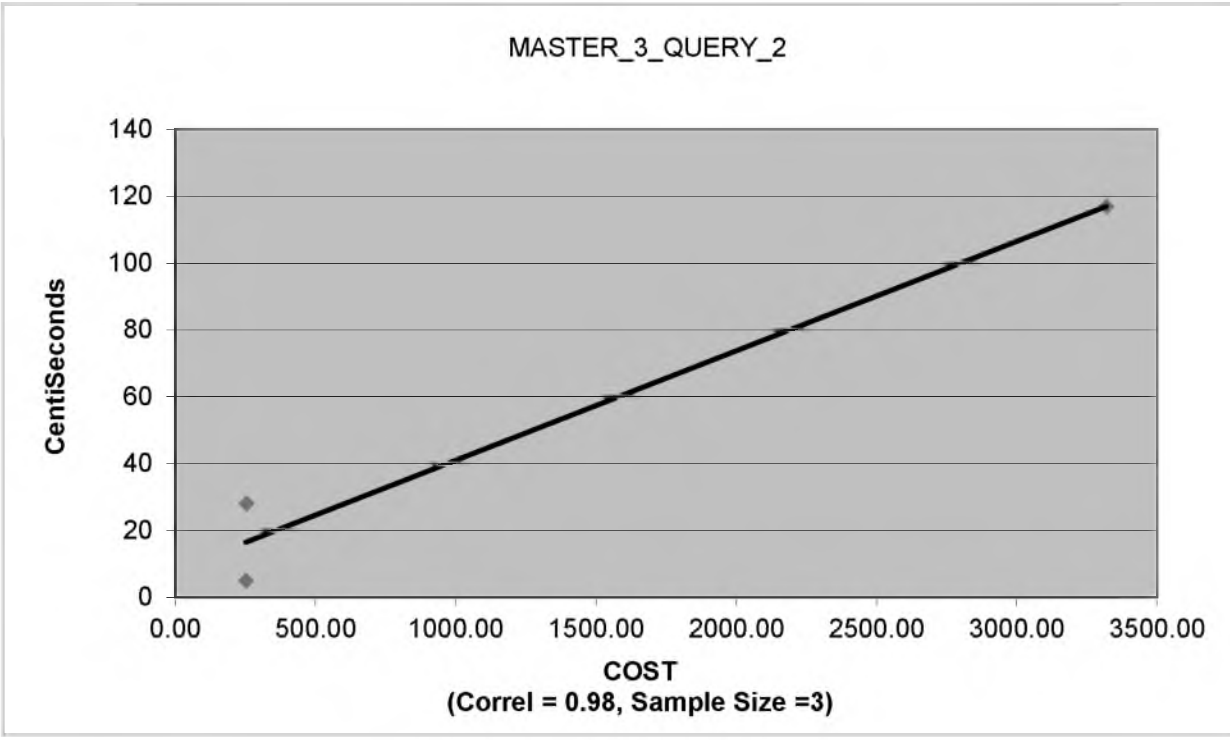
(Correl = 0.00, Sample Size = 311)



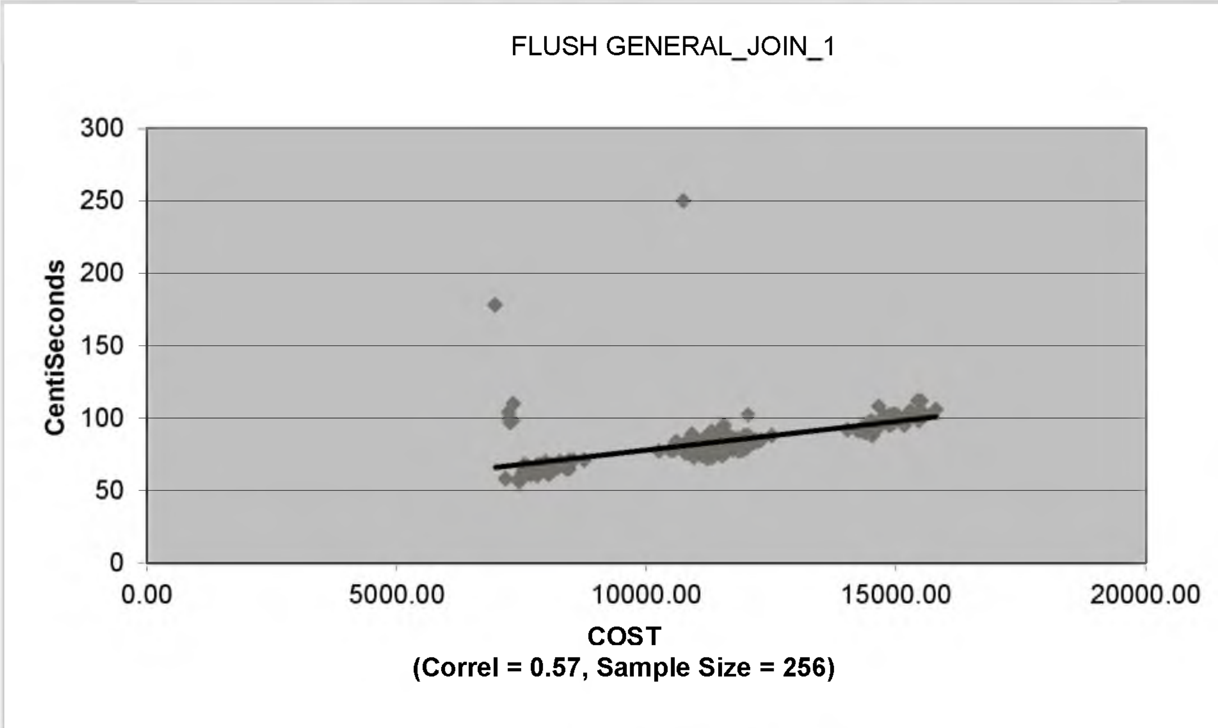
(Correl = -0.21, Sample Size = 174)

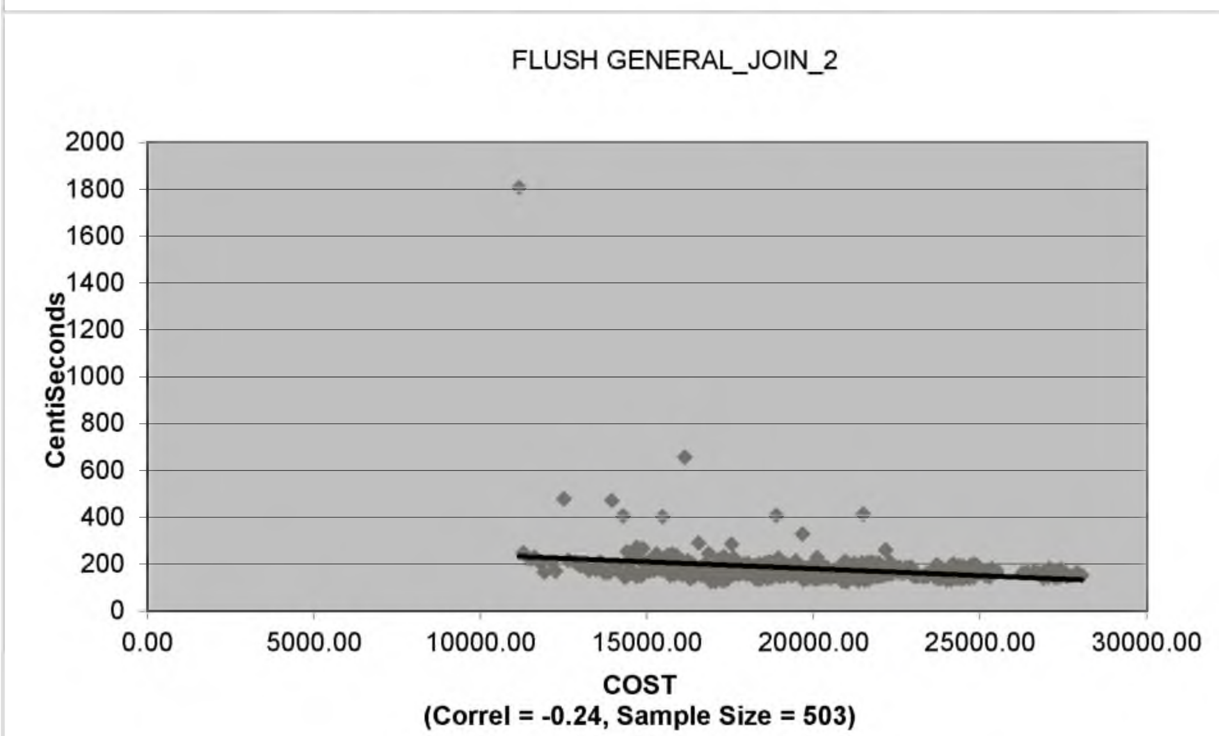


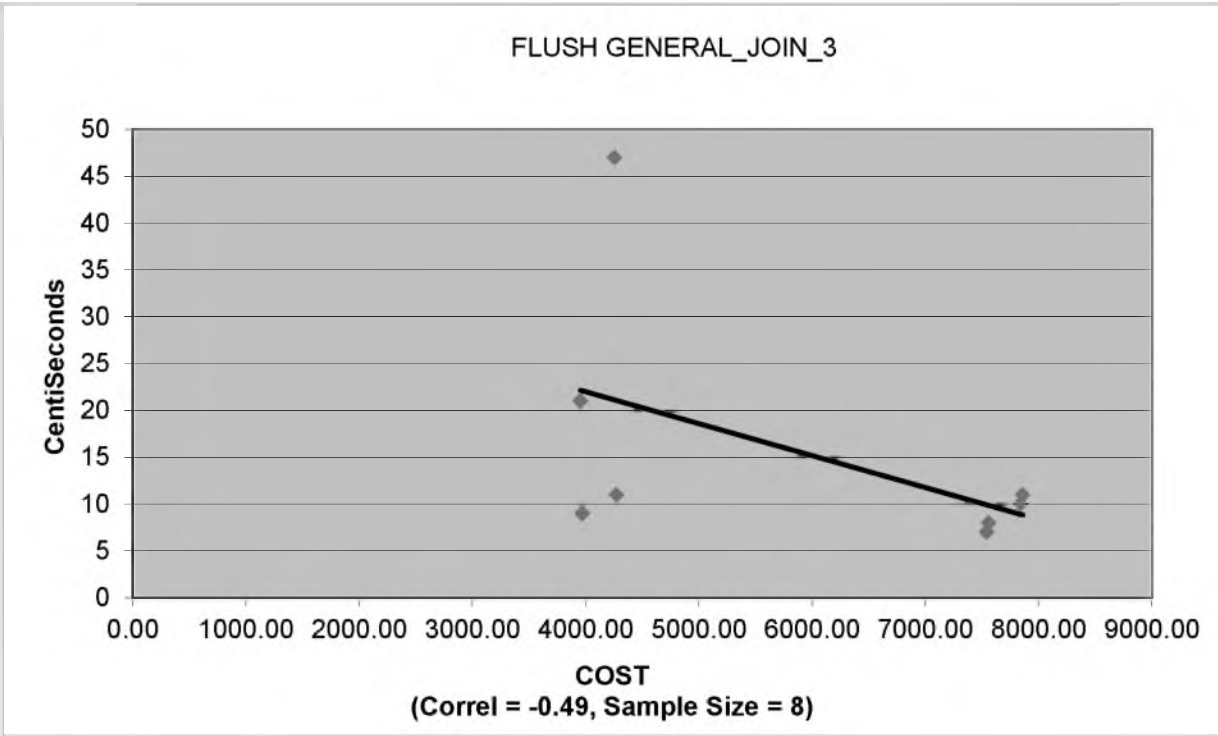
(Correl = -0.27, Sample Size = 80)

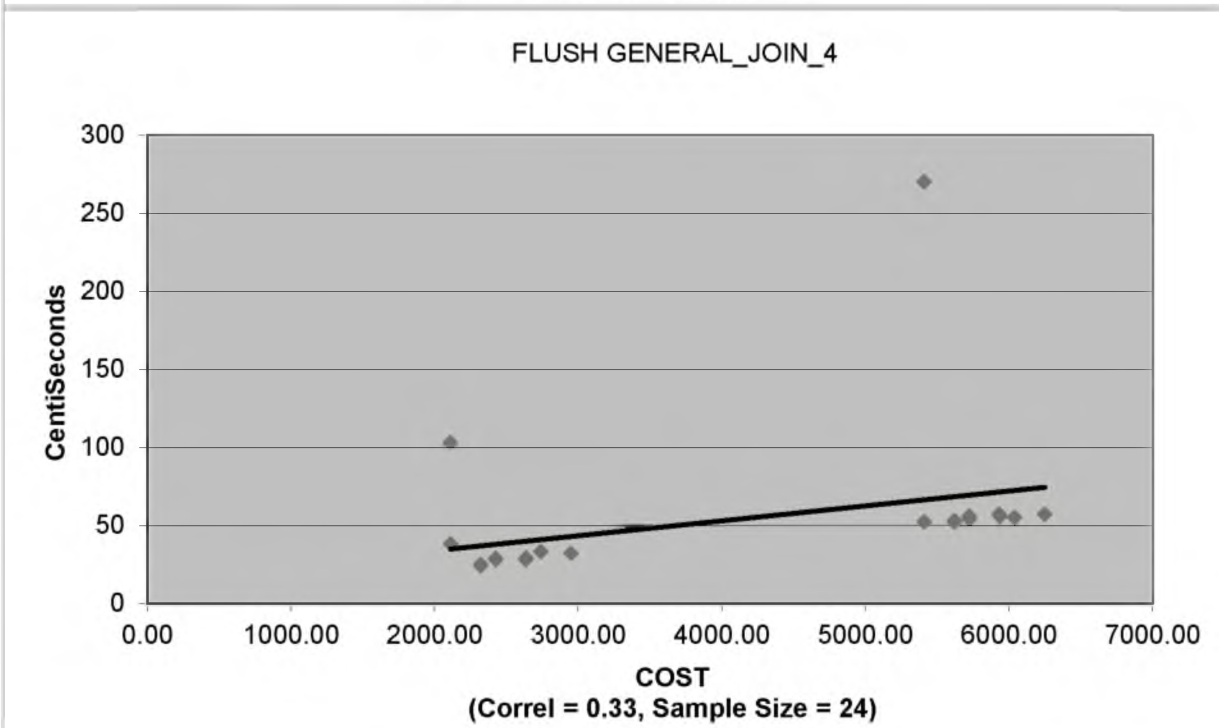

MASTER_3_QUERY_2
140
120
100
80
60
40
20
0
CentiSeconds
0.00
500.00
1000.00
1500.00
2000.00
2500.00
3000.00
3500.00
COST
(Correl = 0.98, Sample Size =3)

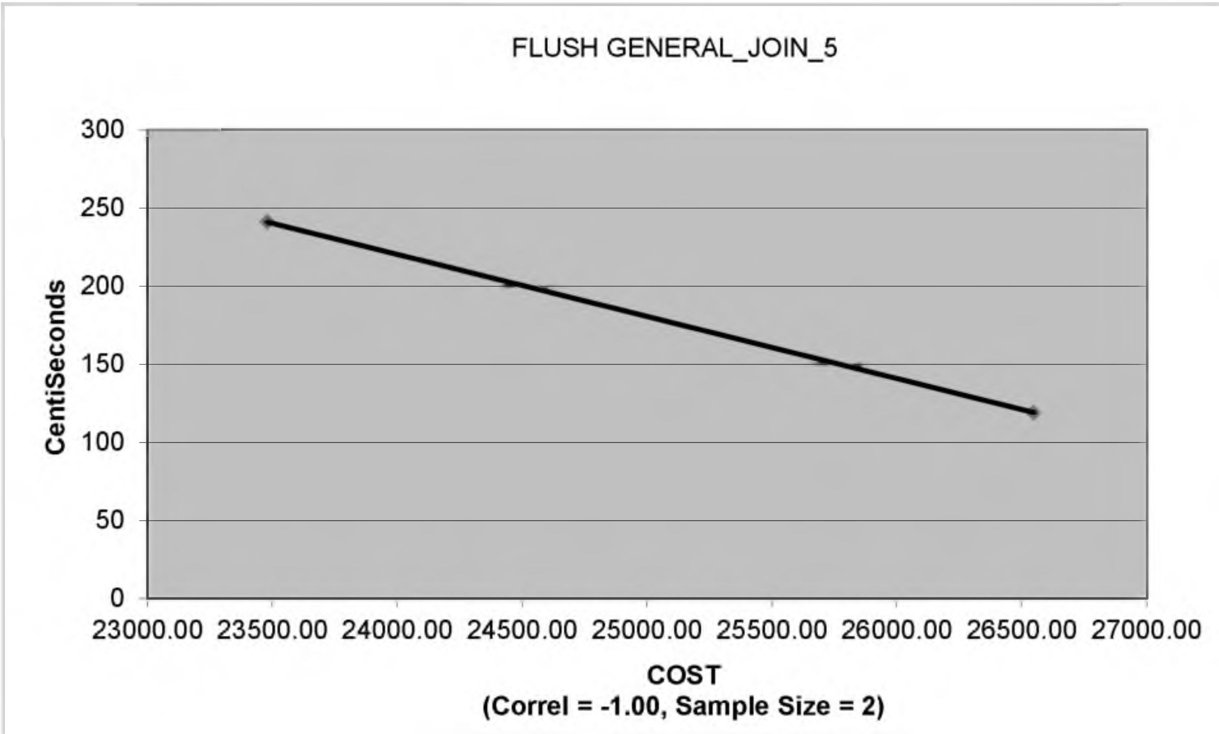
Appendix 6

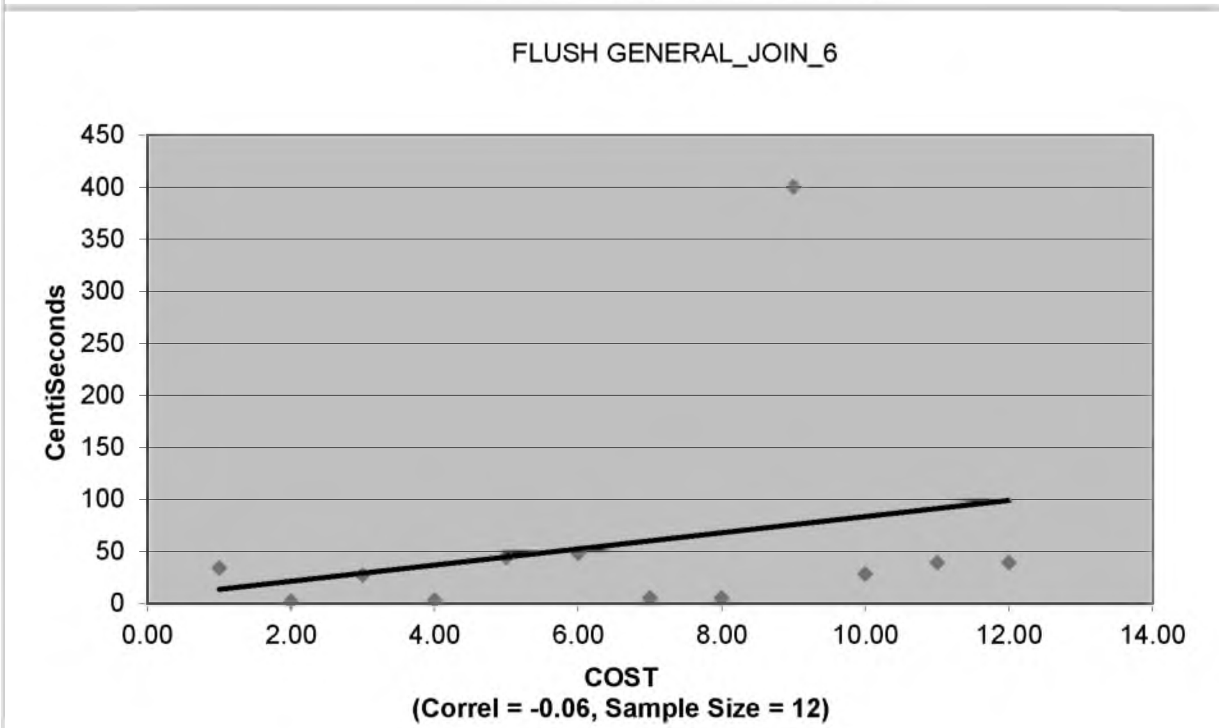

FLUSH GENERAL_JOIN_1
CentiSeconds
0
50
100
150
200
250
300
0.00
5000.00
10000.00
15000.00
20000.00
COST
(Correl = 0.57, Sample Size = 256)

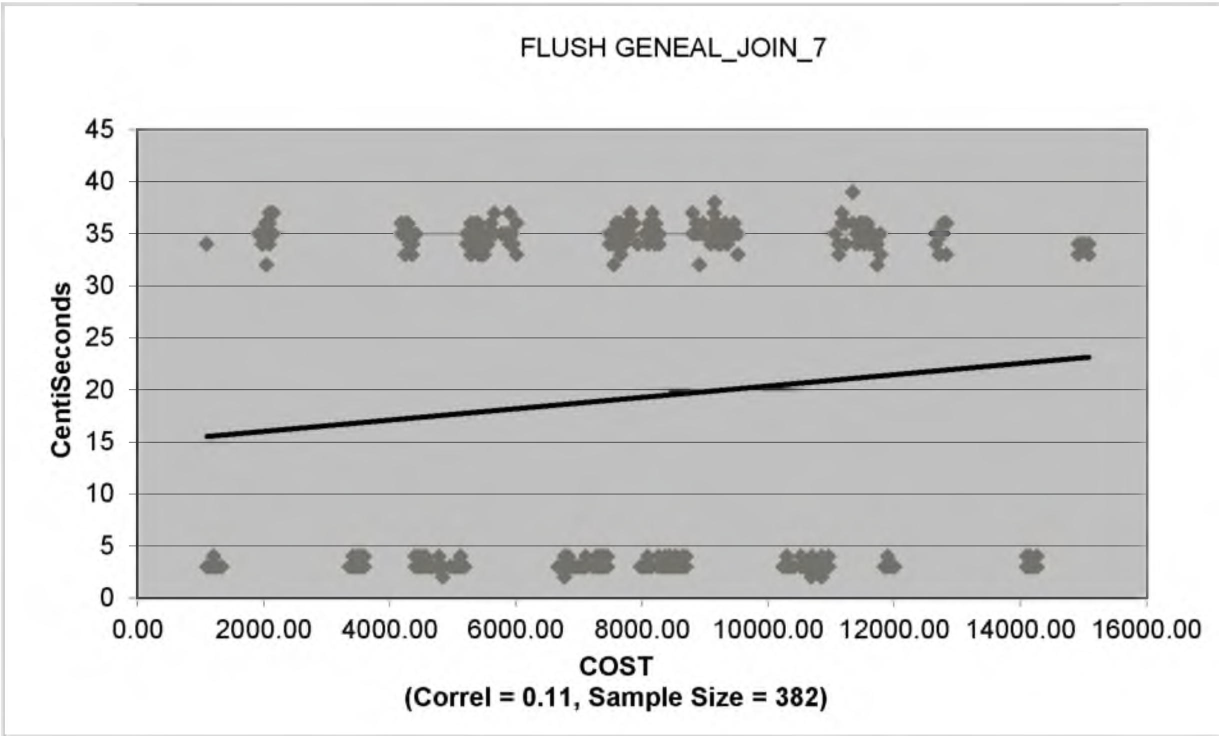

FLUSH GENERAL_JOIN_2
CentiSeconds
0
200
400
600
800
1000
1200
1400
1600
1800
2000
0.00
5000.00
10000.00
15000.00
20000.00
25000.00
30000.00
COST
(Correl = -0.24, Sample Size = 503)

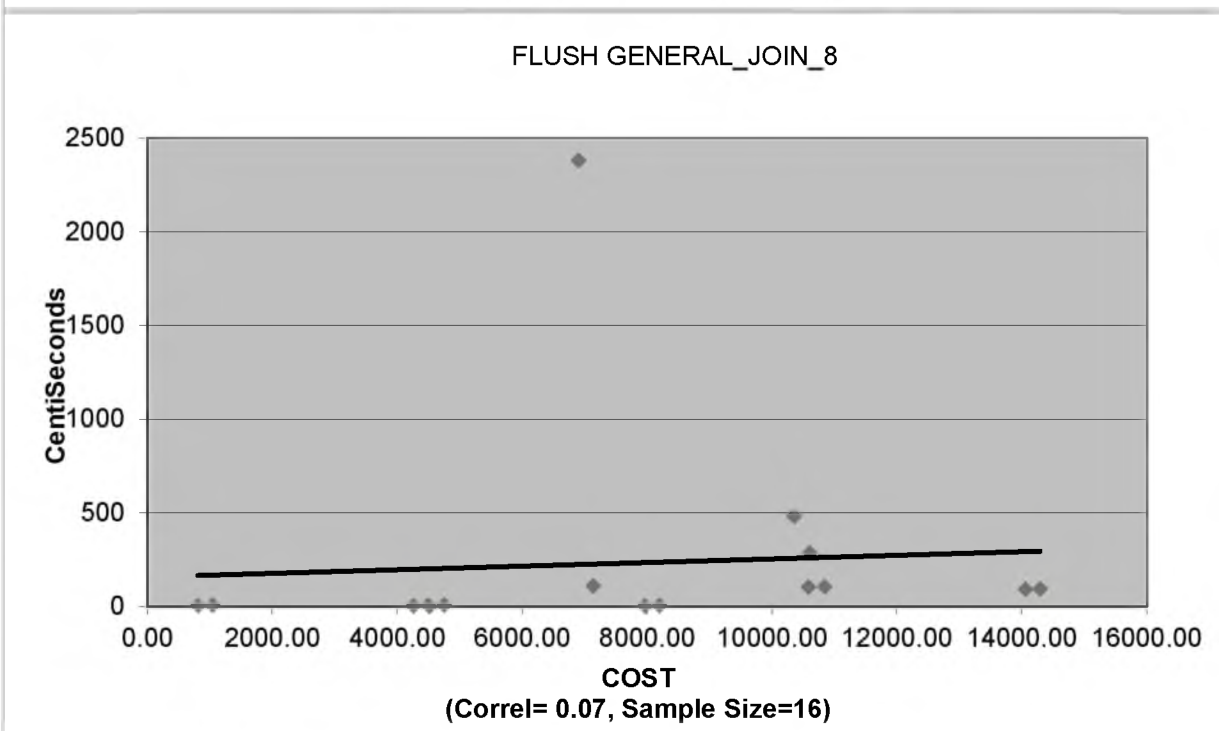

FLUSH GENERAL_JOIN_3
CentiSeconds
50
45
40
35
30
25
20
15
10
5
0
0.00
1000.00
2000.00
3000.00
4000.00
5000.00
6000.00
7000.00
8000.00
9000.00
COST
(Correl = -0.49, Sample Size = 8)

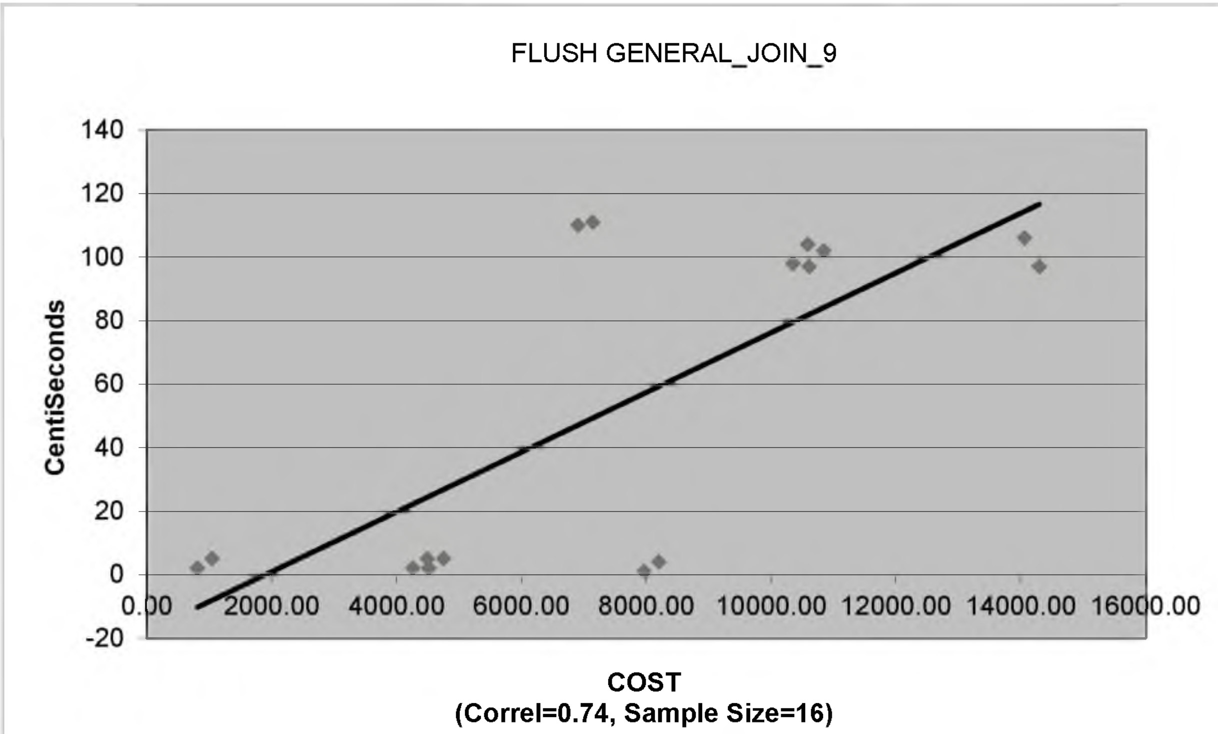

FLUSH GENERAL_JOIN_4
CentiSeconds
300
250
200
150
100
50
0
0.00
1000.00
2000.00
3000.00
4000.00
5000.00
6000.00
7000.00
COST
(Correl = 0.33, Sample Size = 24)

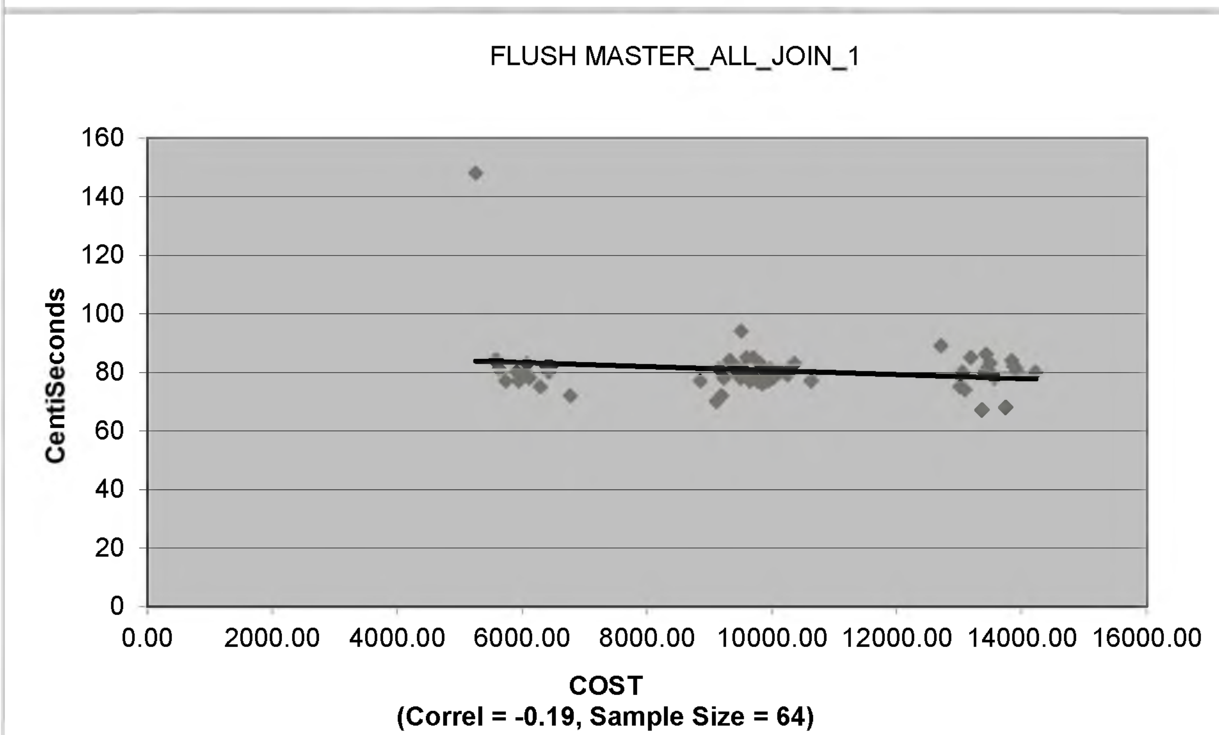

FLUSH GENERAL_JOIN_5
CentiSeconds
300
250
200
150
100
50
0
23000.00 23500.00 24000.00 24500.00 25000.00 25500.00 26000.00 26500.00 27000.00
COST
(Correl = -1.00, Sample Size = 2)

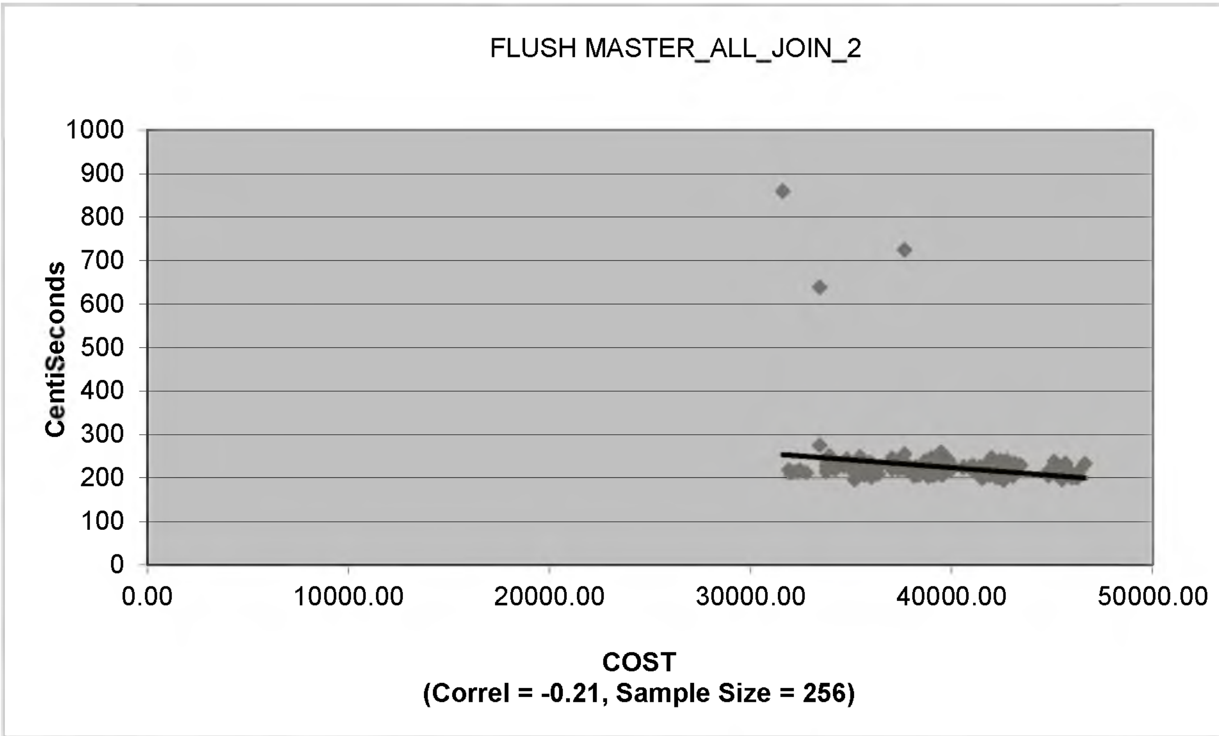

FLUSH GENERAL_JOIN_6
CentiSeconds
450
400
350
300
250
200
150
100
50
0
0.00 2.00 4.00 6.00 8.00 10.00 12.00 14.00
COST
(Correl = -0.06, Sample Size = 12)

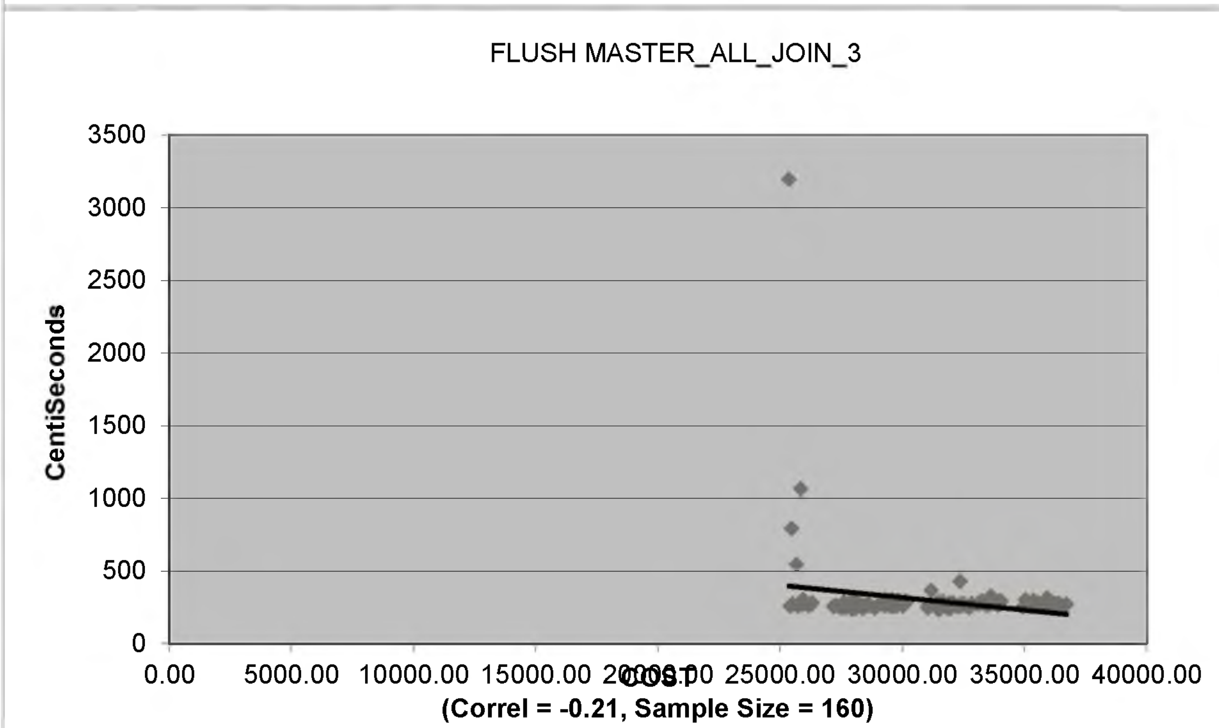

FLUSH GENEAL_JOIN_7
45
40
35
30
25
20
15
10
5
0
CentiSeconds
0.00
2000.00
4000.00
6000.00
8000.00
10000.00
12000.00
14000.00
16000.00
COST
(Correl = 0.11, Sample Size = 382)

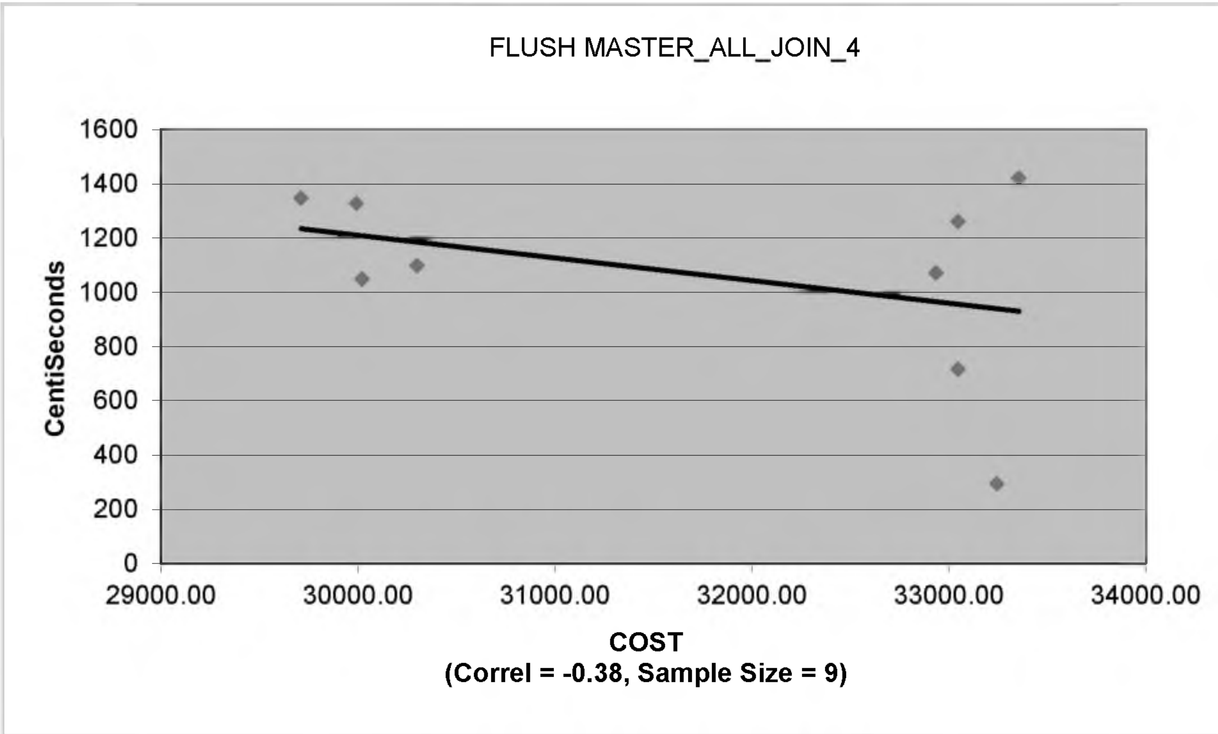

FLUSH GENERAL_JOIN_8
2500
2000
1500
1000
500
0
CentiSeconds
0.00
2000.00
4000.00
6000.00
8000.00
10000.00
12000.00
14000.00
16000.00
COST
(Correl= 0.07, Sample Size=16)

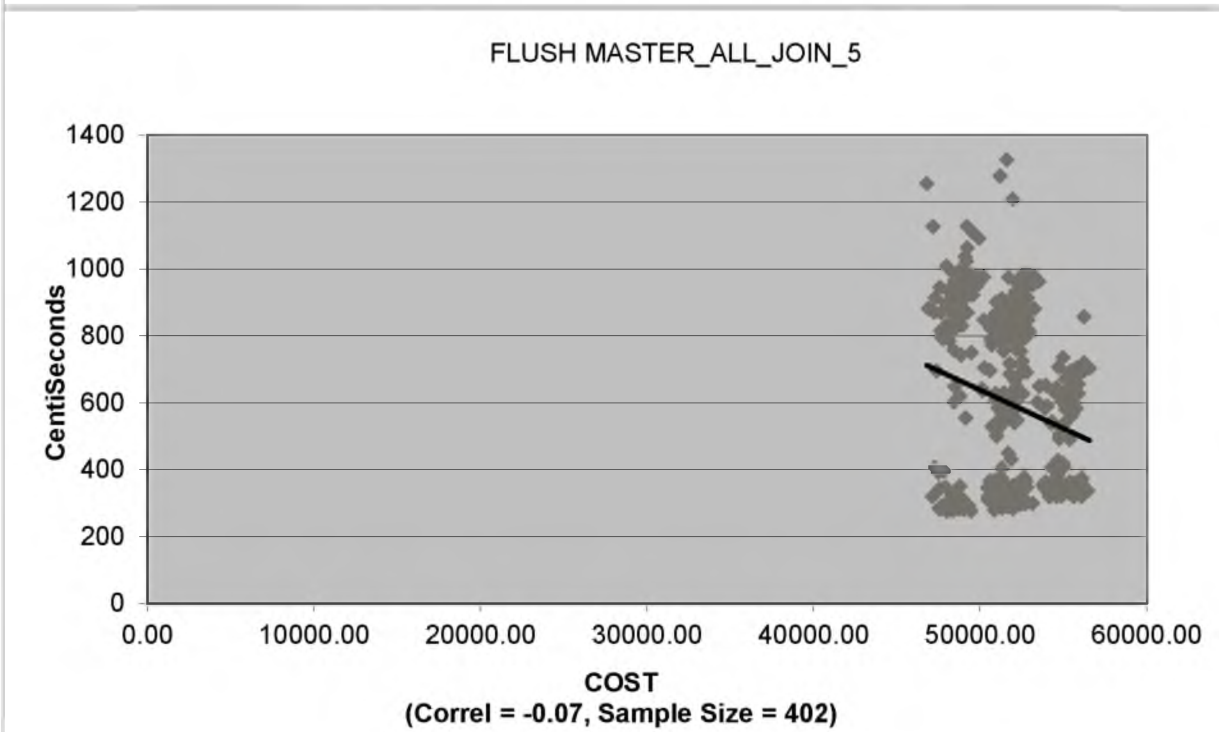

FLUSH GENERAL_JOIN_9
140
120
100
80
60
40
20
0
-20
CentiSeconds
0.00
2000.00
4000.00
6000.00
8000.00
10000.00
12000.00
14000.00
16000.00
COST
(Correl=0.74, Sample Size=16)

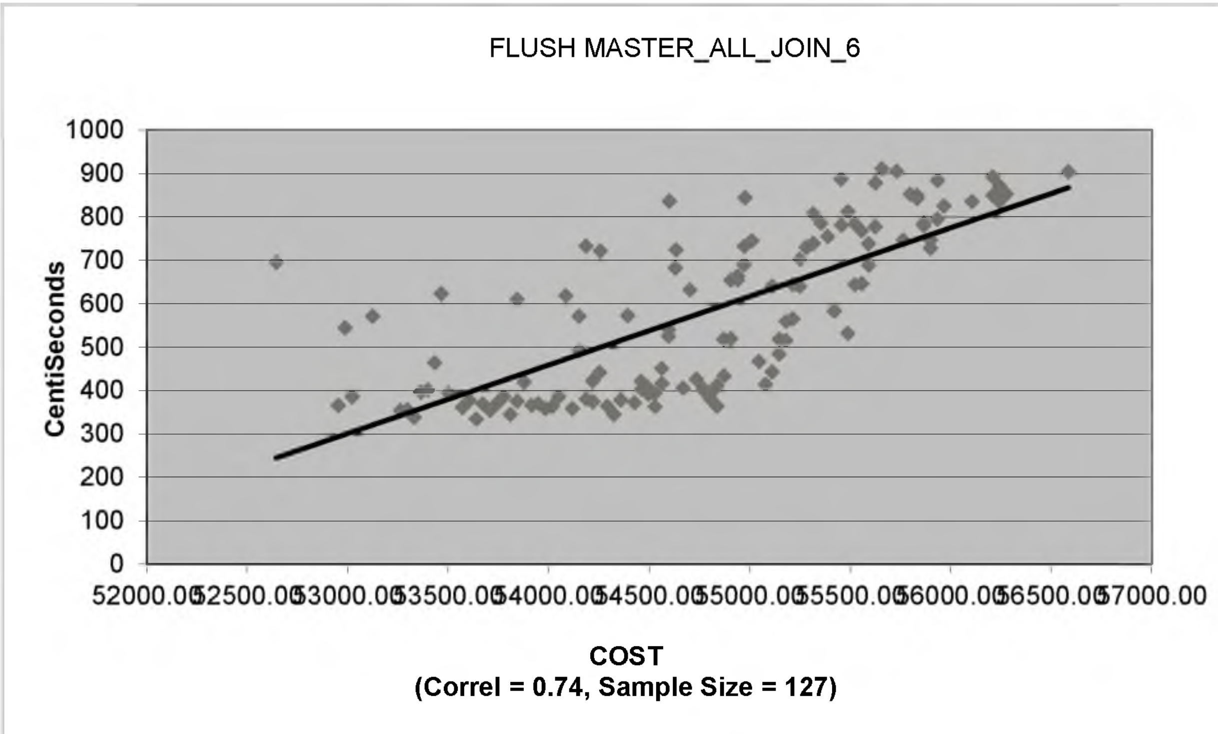

FLUSH MASTER_ALL_JOIN_1
160
140
120
100
80
60
40
20
0
CentiSeconds
0.00
2000.00
4000.00
6000.00
8000.00
10000.00
12000.00
14000.00
16000.00
COST
(Correl = -0.19, Sample Size = 64)

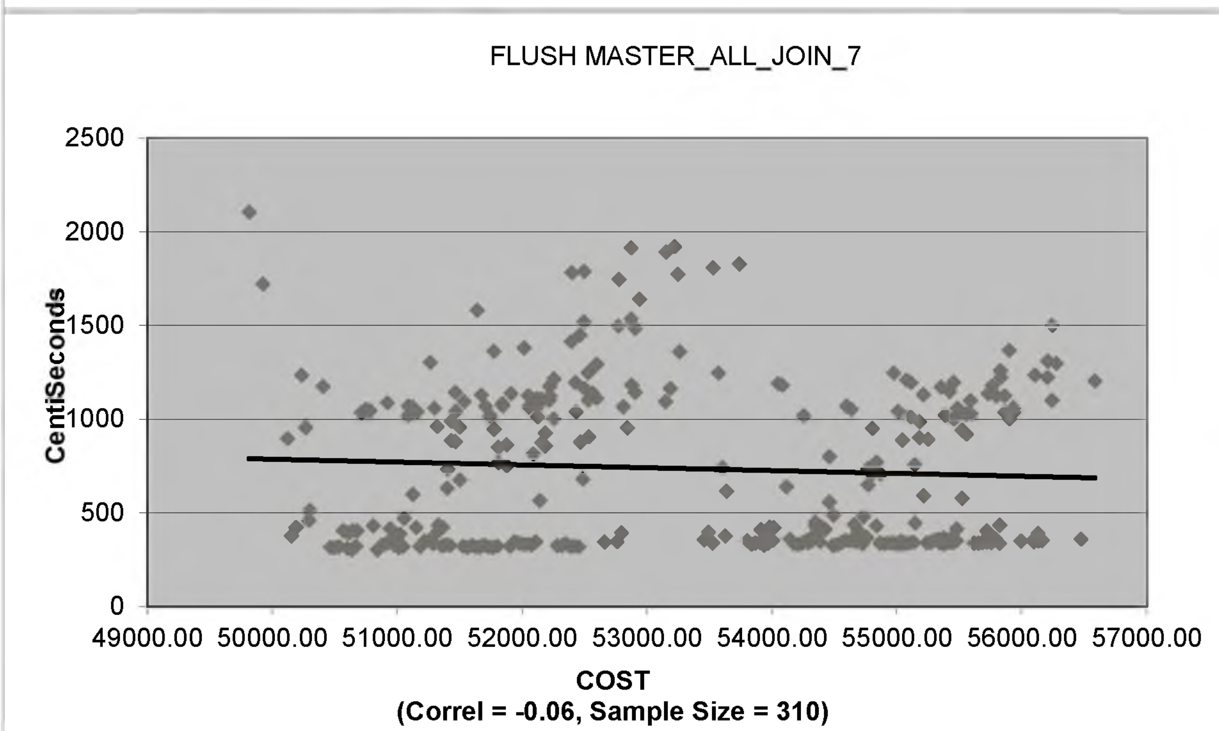

FLUSH MASTER_ALL_JOIN_2
CentiSeconds
1000
900
800
700
600
500
400
300
200
100
0
0.00
10000.00
20000.00
30000.00
40000.00
50000.00
COST
(Correl = -0.21, Sample Size = 256)

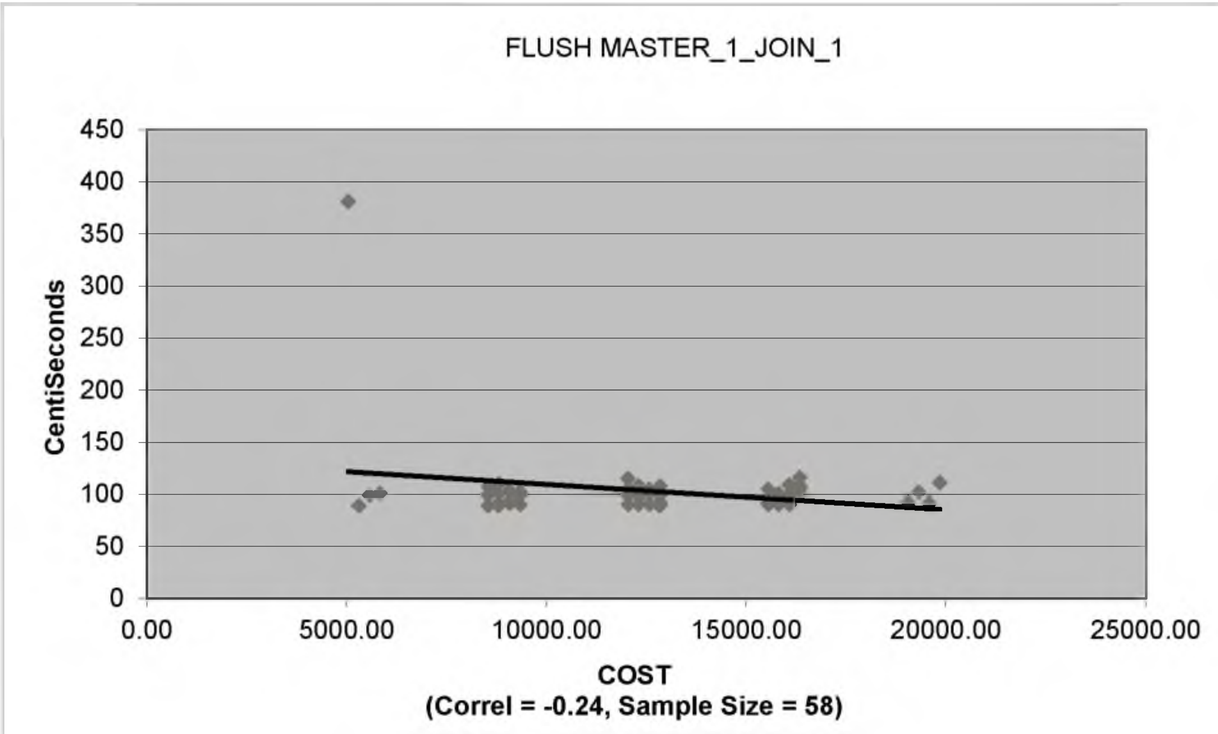

FLUSH MASTER_ALL_JOIN_3
CentiSeconds
3500
3000
2500
2000
1500
1000
500
0
0.00
5000.00
10000.00
15000.00
20000.00
25000.00
30000.00
35000.00
40000.00
COST
(Correl = -0.21, Sample Size = 160)

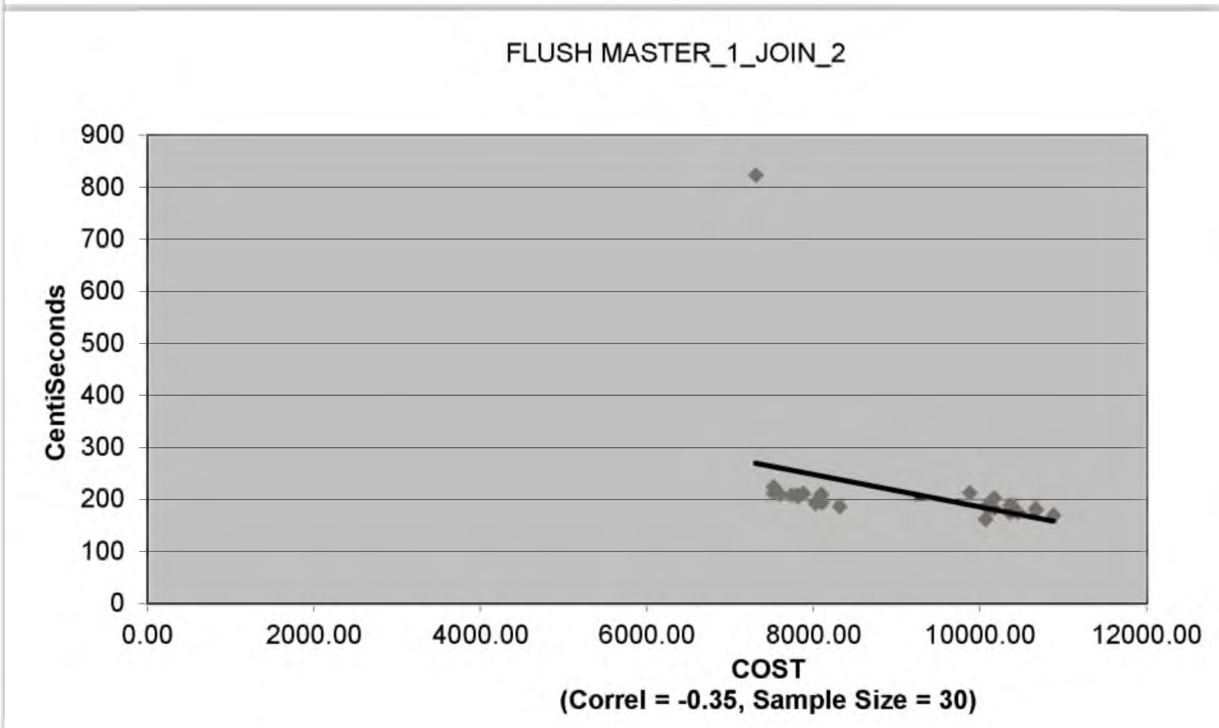

FLUSH MASTER_ALL_JOIN_4
CentiSeconds
1600
1400
1200
1000
800
600
400
200
0
29000.00
30000.00
31000.00
32000.00
33000.00
34000.00
COST
(Correl = -0.38, Sample Size = 9)

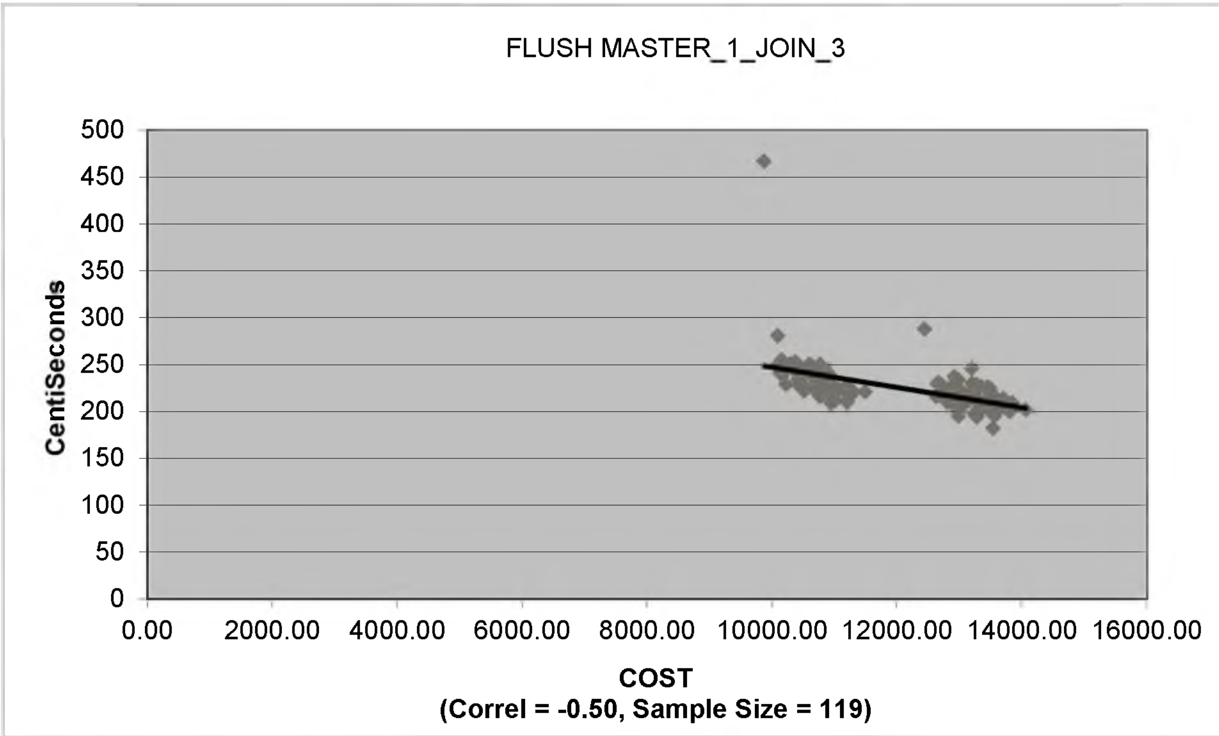

FLUSH MASTER_ALL_JOIN_5
CentiSeconds
1400
1200
1000
800
600
400
200
0
0.00
10000.00
20000.00
30000.00
40000.00
50000.00
60000.00
COST
(Correl = -0.07, Sample Size = 402)

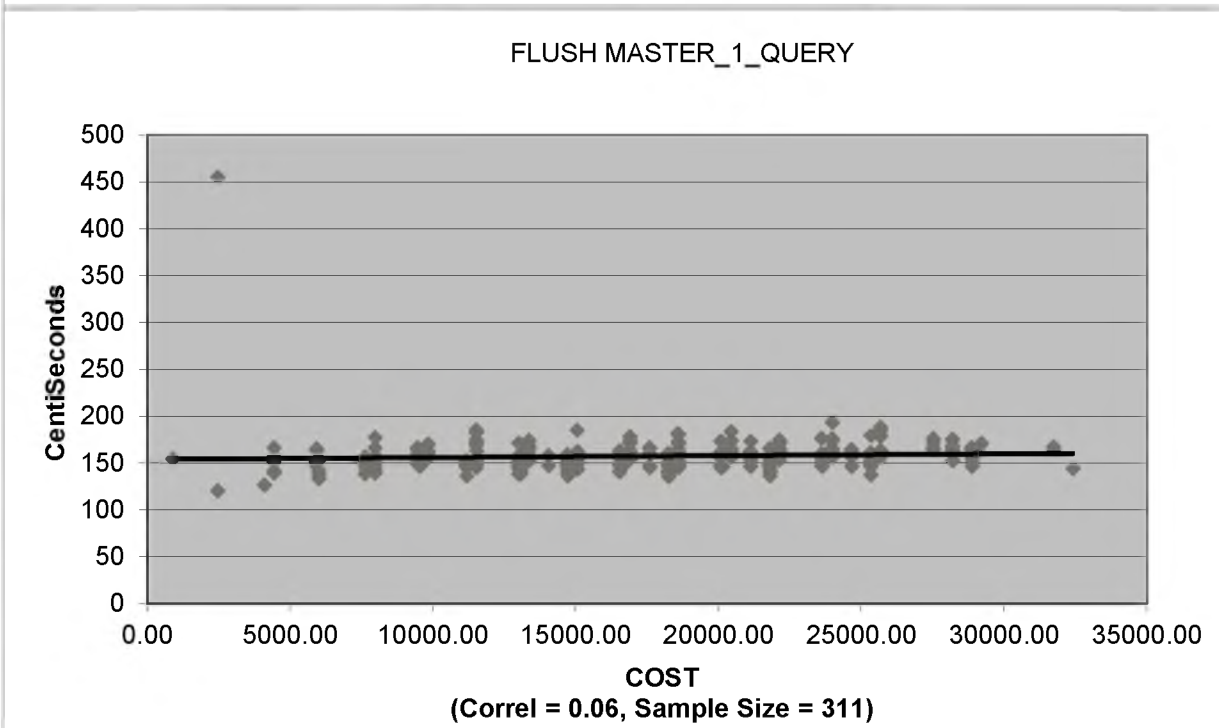

FLUSH MASTER_ALL_JOIN_6
CentiSeconds
1000
900
800
700
600
500
400
300
200
100
0
52000.00 52500.00 53000.00 53500.00 54000.00 54500.00 55000.00 55500.00 56000.00 56500.00 57000.00
COST
(Correl = 0.74, Sample Size = 127)

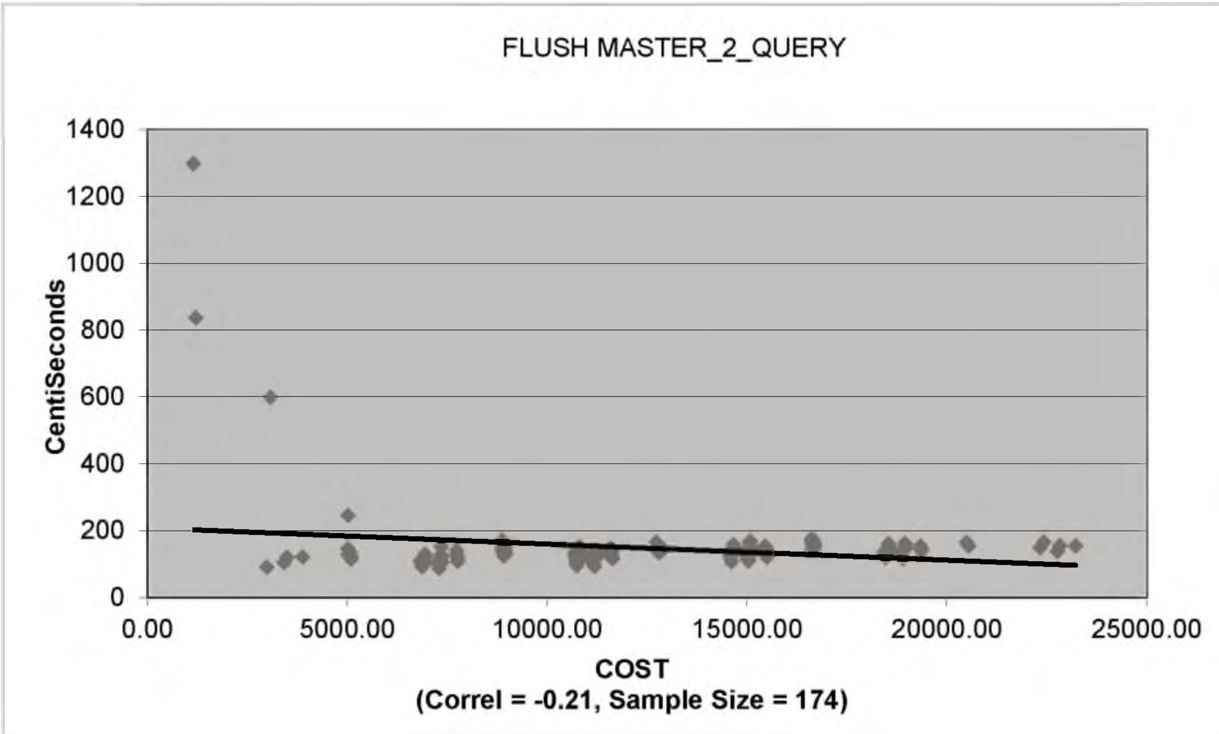

FLUSH MASTER_ALL_JOIN_7
CentiSeconds
2500
2000
1500
1000
500
0
49000.00 50000.00 51000.00 52000.00 53000.00 54000.00 55000.00 56000.00 57000.00
COST
(Correl = -0.06, Sample Size = 310)

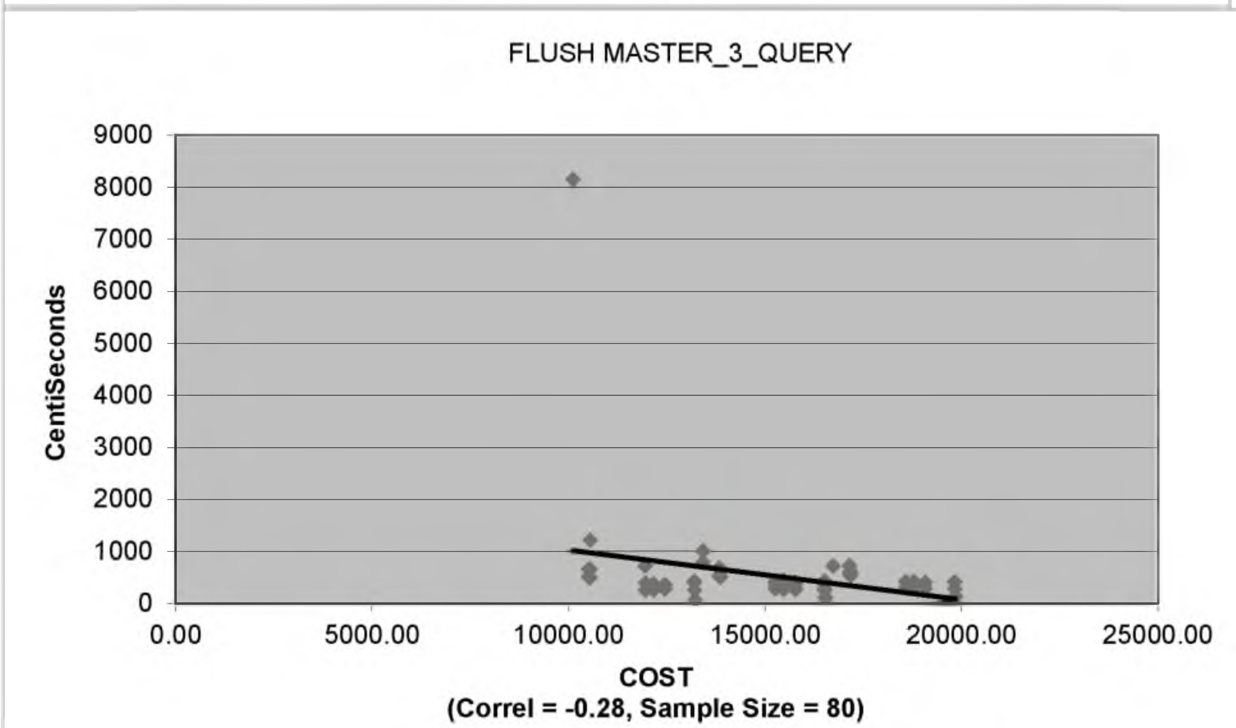

FLUSH MASTER_1_JOIN_1
CentiSeconds
450
400
350
300
250
200
150
100
50
0
0.00
5000.00
10000.00
15000.00
20000.00
25000.00
COST
(Correl = -0.24, Sample Size = 58)

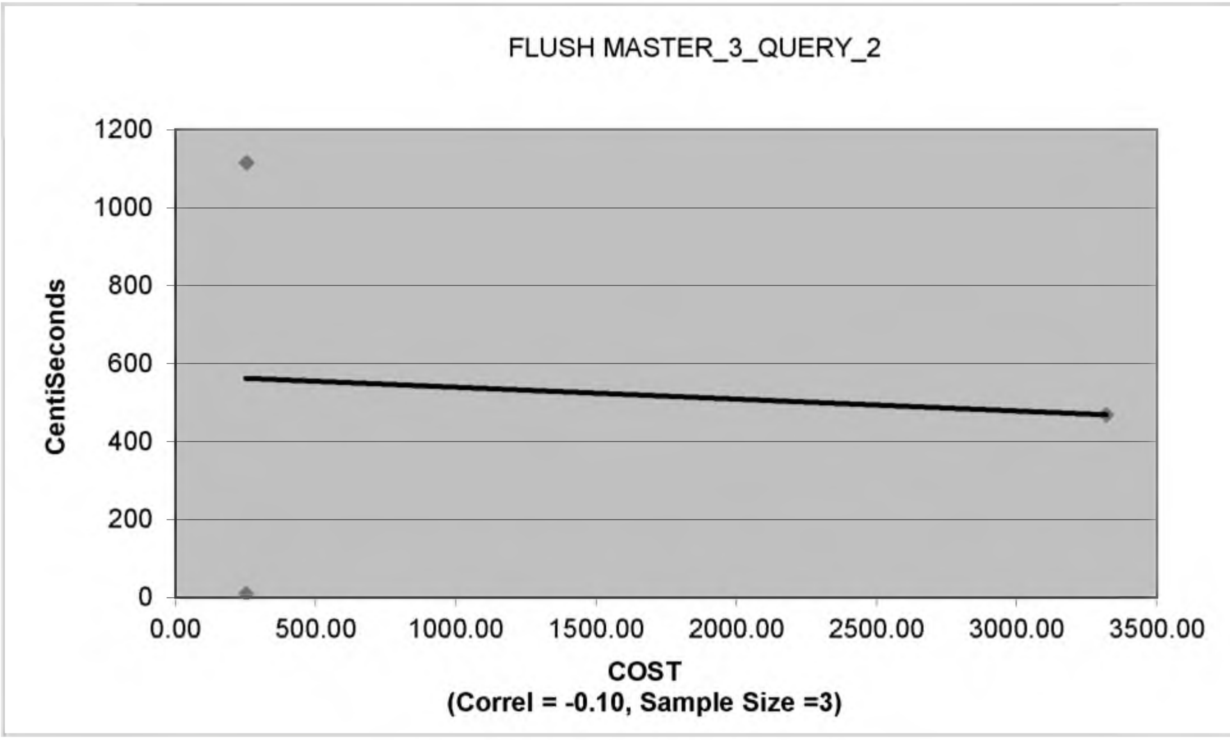

FLUSH MASTER_1_JOIN_2
CentiSeconds
900
800
700
600
500
400
300
200
100
0
0.00
2000.00
4000.00
6000.00
8000.00
10000.00
12000.00
COST
(Correl = -0.35, Sample Size = 30)


FLUSH MASTER_1_JOIN_3
CentiSeconds
500
450
400
350
300
250
200
150
100
50
0
0.00
2000.00
4000.00
6000.00
8000.00
10000.00
12000.00
14000.00
16000.00
COST
(Correl = -0.50, Sample Size = 119)


FLUSH MASTER_1_QUERY
CentiSeconds
500
450
400
350
300
250
200
150
100
50
0
0.00
5000.00
10000.00
15000.00
20000.00
25000.00
30000.00
35000.00
COST
(Correl = 0.06, Sample Size = 311)


FLUSH MASTER_2_QUERY
CentiSeconds
1400
1200
1000
800
600
400
200
0
0.00
5000.00
10000.00
15000.00
20000.00
25000.00
COST
(Correl = -0.21, Sample Size = 174)


FLUSH MASTER_3_QUERY
CentiSeconds
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0.00
5000.00
10000.00
15000.00
20000.00
25000.00
COST
(Correl = -0.28, Sample Size = 80)


FLUSH MASTER_3_QUERY_2
1200
1000
800
600
400
200
0
CentiSeconds
0.00
500.00
1000.00
1500.00
2000.00
2500.00
3000.00
3500.00
COST
(Correl = -0.10, Sample Size =3)