Fall 2009

Realizing the Technical Advantages of Star Transformation

Karen L. Darling
Regis University

Abstract

Data warehousing and business intelligence go hand in hand, each gives the other purpose for development, maintenance and improvement. Both have evolved over a few decades and build upon initial development. Management initiatives further drive the need and complexity of business intelligence, while in turn expanding the end user community so that business change, results and strategy are affected at the business unit level. The literature, including a recent business intelligence user survey, demonstrates that query performance is the most significant issue encountered. Oracle's data warehouse 10g.2 is examined with improvements to query optimization via "best practice" through Star Transformation. Star Transformation is a star schema query rewrite and join back through a hash join, which provides extensive query performance improvement. Most data warehouses exist as normalized or in 3rd normal form (3NF), while star schemas in a denormalized warehouse are not the "norm". Changes in the database environment must be implemented, along with agreement from business leadership and alignment of business objectives with a Star Transformation project. Often, so much change, shifting priorities and lack of understanding about query optimization benefits can stifle a project. Critical to the success of gaining support and financial backing is the official plan and demonstration of return on investment documentation. Query optimization is highly complex. Both the technological and business entities should prioritize goals and consider the benefits of improved query response time, realizing the technical advantages of Star Transformation.

Acknowledgements

I would like thank Bill Padfield, my thesis advisor for spending the time to support and assist me with this project. Special thanks to my children, Carly and Zach for their patience, support and consideration of my efforts to research and write this thesis paper.

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

Business Intelligence (BI) is the process of querying data from a database or data warehouse environment for analysis and the identification of patterns and trends. Data mining, as defined by Thearling, is "the extraction of hidden, predictive information from large databases" and is geared towards solving current issues and creating new business prospects by predicting trends and uncovering information previously unknown (Thearling). BI requires data mining through querying of a database. Query response must produce expected results within a reasonably short time frame, preferably seconds. Timely and accurate BI can help a company achieve a competitive edge with intelligent decision making in a struggling economy.

Data warehousing and BI are key elements in today's business industry. Organizations are focusing on faster and better decision making in order to stay competitive. Improvements in BI initiatives pose data warehouse challenges with query response time. The "intelligent enterprise", as described by Rosenberg, is an organizational term which means that in order to achieve a competitive edge, the enterprise casts a wider net to capture more information by adding data sources and users for mining information (Rosenberg, 2006). Information that is typically mined is information like employee data, supply spend, sales transactions, and customer demographics to existing large data sets like financial transactions. The ability to analyze information efficiently and respond intelligently requires technology and optimized query response time.

The goal of this paper will focus on attaining a current best practice in query optimization unique for the Oracle RDBMS 10*g* version 2 (Oracle, 2002). The terminology used for by Oracle 10g is "Star Transformation", named for the technique which appraises queries against star schemas. Recommendations are current for using Star Transformation in Oracle (initially introduced in Oracle 7, with ongoing improvements in versions 8*i*, 9*i,* 10*g* and 11*g*) and instructs that when implemented, will optimize query response time. Supporting this optimization method as an organization can produce the technical advantages needed in gaining fast access to information that is required for successful BI efforts.

What are the business and technical requirements necessary to optimize a complex query using Star Transformation and subsequently realize all of the technical advantages? There are technical processes and business considerations. Oracle has detailed the technical requirements, but business constraints can prevent this best practice from ever being realized. Timely access to the data stored in a database can be achieved, at a price, and requires an ongoing investment of human and financial resources to keep pace with scaling BI requirements. However, funding and approval for resources to improve technical functionality is often highly scrutinized and difficult to obtain. In struggling economic times, technical projects that have been budgeted can later be revoked to improve the organization's financial bottom line (Stackowiak, 2007).

For the database administrator and the development teams, implementation of the technical requirements necessary to achieve star transformation can seem daunting, with the tasks of bitmap indexing and setting initialization parameters, as well as ensuring that the optimizer has current statistics on which to base its access path direction. The time investment, depending the number and size of the tables, and can become a lengthy, complex project. With efficient query response times, the technical group can focus on other technical optimization needs, scalability or new development projects. For the business managers, faced with approval and support of such a project, the need for improved response time and the resulting improved technical functionality may not be a significant issue for them as day to day business operations, management of people, and responsibilities to their superiors can take priority over query response time. Efficient query response time means improved access to data, and can help an operations manager achieve success both in business objectives and reporting performance measures to their superiors. The issues and priorities for both the database and the management groups clearly merit a detailed review and research project in providing the background, support of achievement and realization of Star Transformation.

A review of the literature will provide the historical perspectives and work done in the relational database as it relates to query optimization methods in Oracle. In addition, the literature review will lend insight to how BI relies upon and coincides with query optimization, but is an area of ongoing need for improvement. The connection between BI in today's business world, and data warehousing, will demonstrate the need for timely access to the information for the end users mining the data. The connection between query return, improved functionality, and its application to BI will lay the foundation for the subsequent discussion on query optimization through Star Transformation. The discussion will include steps in achieving Star Transformation, along with a case study by an interview from an industry DBA who actually implemented Star Transformation in a test environment, and the outcome. Finally, the conclusion will provide some thoughts for future research to improve and support intelligence across the enterprise.

Chapter 2 – Review of Literature and Research Section

2.1 The Impact of Business Intelligence in Conducting Queries

BI and data mining go hand in hand. The integrated database was first theorized in the early 1960's and was aimed at incorporating every aspect of management in a large corporation. With the unveiling of the Management Information System (MIS) concept at an American Management Association's conference in 1960, management consulting firm Arthur D. Little's senior representative challenged that computer technology could "address the problems of corporate administration" (Haigh, 2006). A short time later, supporters of the MIS concept proposed that management in a corporation could access every single detail of information needed to perform their work through a large automated system (Haigh, 2006).

BI is a way of turning data into information and using information to create knowledge (Flanglin, 2005). Organizations utilize BI to gain an understanding of themselves, the customers served, and the market of their business. BI applications access the data through the data repositories available for data retrieval. Typically the data repository is the database, but often disparate legacy systems store data, along with untraditional data sources. The disparate and untraditional sources of data highlight the importance of data integration, which in turn influences enterprise wide deployment of BI.

While the terminology used in BI is relatively new, application of philosophical works to business strategy tells us that knowledge and reflection are powerful tools to help us succeed. Sun Tzu probably said it best in his 6^{th} century BC writing, “The Art of War”; “compare the opposing army with your own, so that you may know where strength is superabundant and where it is deficient” (Giles, 2009). Flanglin summarizes that Tzu’s work is influential and is a foundation for BI today.

The evolution of BI coincides with the development of database systems. Since approximately 1975, deployment of some of the first software tools to conduct analysis on stored information provided assistance with financial planning. Financial information was relatively simple and the data types were consistent (Brookes, 2007). Brookes writes that by the 1980’s, use of spreadsheets were mainstream and as sources for data grew with database development, so did access to it, by use of Structured Query Language (SQL) . SQL is based on relational algebra and is used for managing data in a database. The SQL language can be used for controlling data access, creating schemas or modifying them, or querying the database. SQL can often run behind an application from a data mining user. Since the early days of data mining, application development in the area of BI has included complex data retrieval algorithms and statistical analysis.

Today, more complex methodologies are used to access data and this includes data mining. Data mining and BI today are common in the corporate world because information drives business success in a competitive market. An explosive growth in the utilization of the database for reporting transaction information drove deployment of the “data warehouse”. The data warehouse, toward the end of the 1980’s into the early 1990’s, was intended to separate decision support from transactional databases (Stackowiak, 2007). Traditional data warehouse architecture for online transactional processing (OLTP) consists of raw data, metadata and summary data. Summary data is the addition of materialized views and is a beneficial type of data for the warehouse as it creates computation in advance for extended operations. Figure 2.1 shows an architecture model of a data warehouse with materialized views (summary data) and its connection to data mining, where users access data from different sources (Oracle, 2003).

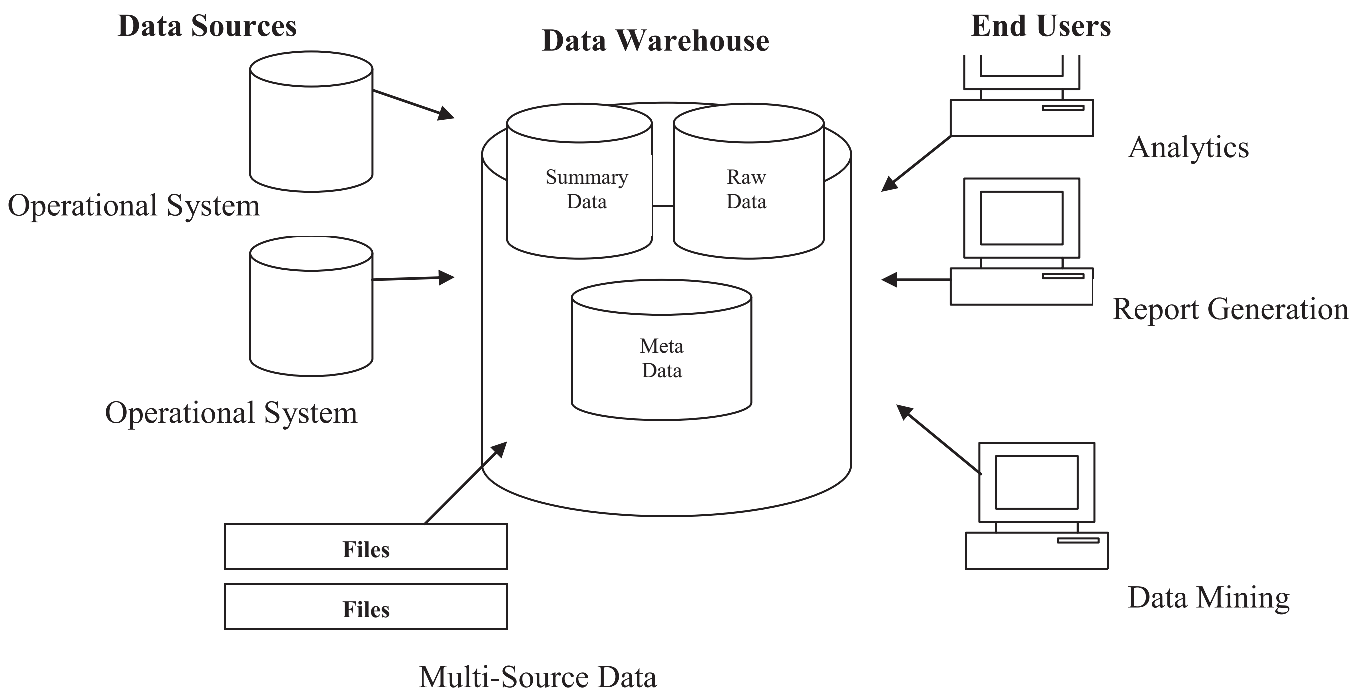

Figure 2.1 Data Warehouse Architecture adapted from Oracle. (Oracle, 2003).

Traditionally, access to data mining capabilities has been limited to the executive suite. Strategic initiatives are being put into place in an effort to leverage information for faster and better decisions. One way organizations are leveraging information is through massive deployment of BI. Employees at every level of the organization are gaining access to BI applications to more readily affect operations and make instant decisions on cost saving measures (Peters, 2006). An improvement in application development has also influenced mass deployment. Additional users running simultaneous queries against the database means that performance can be challenged. Simple queries accessing many tables can also cause a delay in return time.

BI gained through data mining and the associated reporting structures have evolved over the past couple of decades, and improvements in both the technology and methodology for measuring performance have been widespread. With multiple sources of data, data types become more obscure and with different business units using the information, data marts are now used to compartmentalize the data structures. With the multitude of data sources from legacy systems, it is important to acknowledge the need for integration of data sources. Similarly, the absence of data coding standardization and human error in data entry are issues that can place limitations on accuracy of query results. An example of a data discrepancy would be formal and short versions of a person's name at the same address, (e.g., John Smith on 1234 Main Street, Denver, CO. versus Johnny Smith at 1234 Main Street, Denver, CO) which will be seen by the database as two distinctly different people or records.

A recent development in database applications to address inconsistent data is a process for validating, error checking and reconciling data discrepancies called cleansing or scrubbing. In a large database, the quantity of data errors increase and can affect a significant amount of information, making what seems like an occasional error a significant data validation issue. Scrubbing techniques are carried out and put into place before the end user of the database query ever realizes the data poses problems for accurate retrieval of information. The quality of data is less than desired; creating a need for correcting inconsistencies in an area where data can be scrubbed or cleansed. The practice of scrubbing data is a necessity for effective BI and optimal query run time. Scrubbing data occurs in a "staging" area (Oracle, 2003). Figure 2.2 displays additional features of a data warehouse architecture that supports a "staging" area, along with data marts for different organizational groups to access departmental specific data interests.

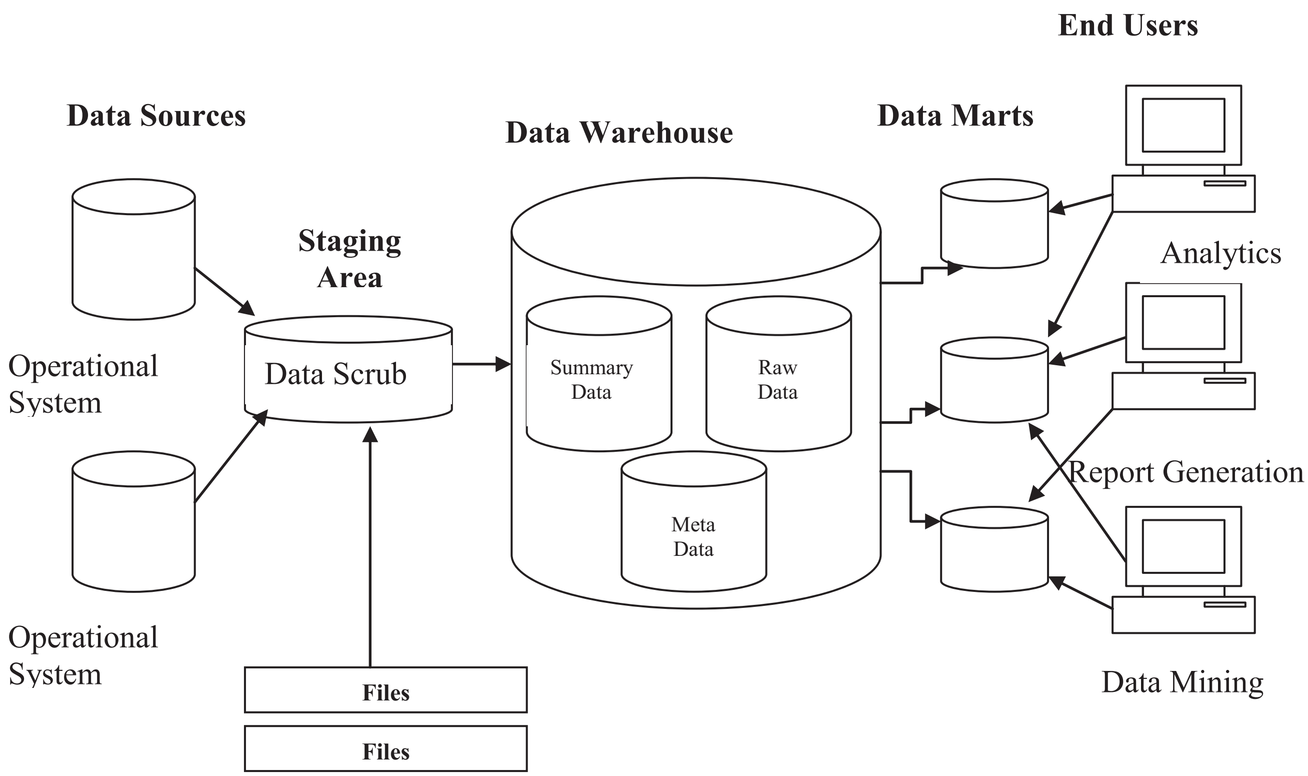

Figure 2.2 Data Warehouse Architecture with Staging and Data Marts adapted from Oracle (Oracle, 2003).

Critical to BI is the logical component of the database, and this is typically structured through an entity relationship diagram (ERD) where the data, types and connections to one another are modeled for consistency. Normalized and dimensional models both have utility for the ERD, which directly reflect the key components of the business. A well developed ERD as part of the data warehouse is critical to effective BI. There are currently applications available to develop an ERD through data modeling and it is usually incorporated into the database development to ensure that the relationships are defined appropriately. Oracle provides this type of application or it can be added in from a 3rd party vendor. Figure 2.3 shows an example of the ERD concept, adapted from a third party vendor, Quest, where the application uses the ERD to generate the SQL query (Quest, 2006).

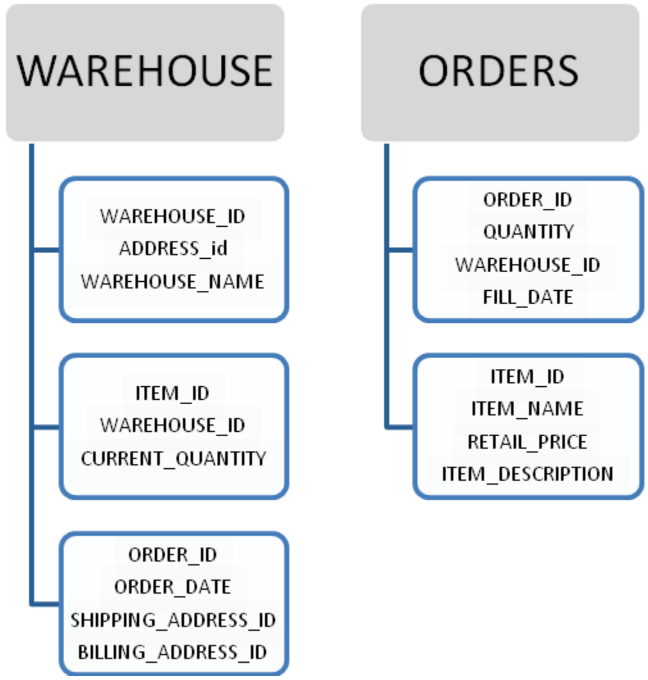

Figure 2.3 Data Modeling with ERD (Quest, 2006).

Trending information about the state of a business has been reported through spreadsheets, dashboards, and scorecards. In a 2006 TDWI Best Practices Report, writer Wayne Eckerson explains "Dashboards and scorecards are critical elements in supporting business performance management processes, which enable executives to more effectively communicate, monitor and adjust business strategy" (Eckerson, 2006). The use of dashboards is necessary for executives and business managers to understand and plan for strategy. Few defines the BI dashboard as a "display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance" (Few, 2004). Figure 2.4 is an example of a performance measurement for dashboards.

Dashboards can be created from spreadsheets, presentation software, macros, simple and small data stores, software applications, and a complete suite of sophisticated mining and statistical analysis tools. Companies typically look at dashboards weekly, monthly, quarterly and annually, or whenever a special project or problem is occurring. Teams of individuals outside of the technical team create dashboards and scorecards and can often be referred to as “decision support” personnel. Decision support systems and personnel drive the organizational decision making process by the method in which it is decided the data will be displayed graphically. Important information that is not apparent in a spreadsheet can speak volumes pictorially in a graph.

An important measure to note that is popular in the business industry today is that of “predictive analytics”. Predictive analytics uses data mining, statistical methods, and mathematical algorithms to logically predict future events. An example of predictive analytics is sales probability in retail, travel trends, and credit risk predictions. Popular in predictive analytics is the “recency” model, meaning start with the most recent to get the best response. One can apply this to customers most understandably (Siegel, 2005).

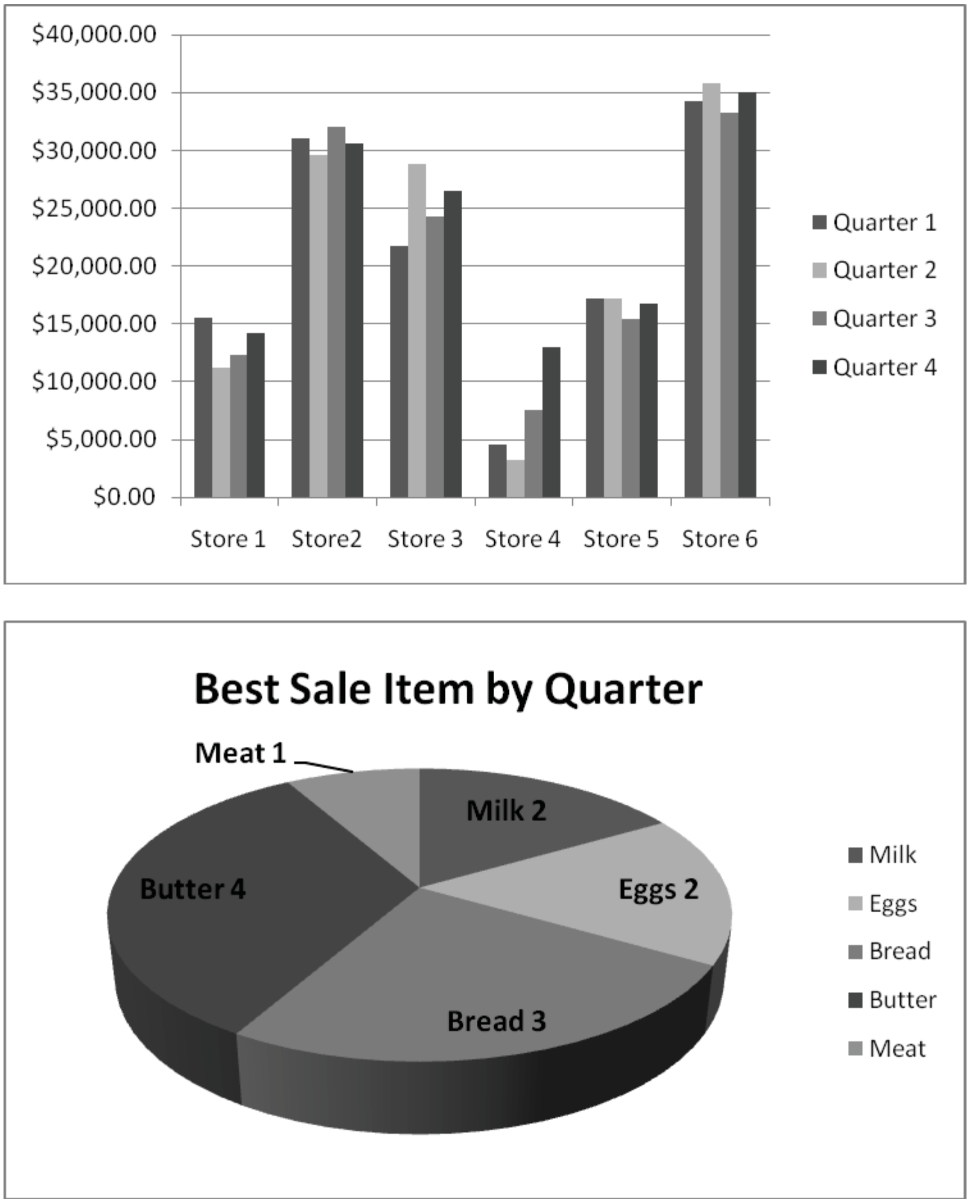

Figure 2.4 BI Dashboard Performance Measures.

Managers must be able to measure achievements and monitor their specific contributions towards customer goals (Oracle, 2008). Management contribution information in the form of regular measure of performance, in BI, is used for internal reporting mechanisms (Gonzales, 2002). Management initiatives have added benefit for an organization and they rely on BI to facilitate success. Netezza Corporation describes current BI management initiatives, popular in today's business world as; Customer Relationship Management (CRM), operations management, Supply Chain Management (SCM), Enterprise Resource Planning (ERP) and Partner Relationship Management (PRM). CRM embraces all aspects of the customer's information; the data analysis drives marketing and service activities. In operations management, trending allows organizations to conduct process improvement and achieve financial goals. SCM allows business partners to efficiently manage supplies based on demand. ERP evaluates the organization's business, financial and human resources across the enterprise, and facilitates forecasting for effective planning. PRM assists companies in partnership by effectively reviewing business-partner related activities for positive outcomes. Management initiatives utilize dynamic information from the data warehouse and other data sources to facilitate their processes (Netezza, 2004).

Other popular methodologies focused at achieving management excellence while incorporating BI in specific industries are worth mentioning because of their concentrated utilization of BI and association with large databases (Oracle, 2008). The Supply Chain Operations Reference (SCOR) model is a management methodology for supply chain operations that intends to align its process with the business strategy (Pragmatek, 2001). The Information Technology Infrastructure Library (ITIL) is a combination of best practice for service management, development, people and processes for information technology components of an organization. The European Framework for Quality Improvement (EFQM) is a quality management methodology used to organize and assess the quality processes of an organization. Used widely in European countries, EFQM, according to Nabitz, is a "self assessment, benchmarking, external review and quality awards" model and uses the concepts of "enablers" (internal people and processes), and "results" (measurements of success) in organizational excellence. All management initiatives require information that may be overlapping and subsequently in competition for database resources. Complexity and overlap of management initiatives may place additional resource requirements on the technical component of BI, the data warehouse, in order to demonstrate the desired measurement.

BI today contributes to a better knowledge base and competitive edge in organizations that maximize its utility. BI is best leveraged when an organization can integrate with currently available applications. The goal of BI is to plan for expected outcomes and make informed decisions that are beneficial to business. Businesses that embark upon management initiatives can extend the power of BI to predict future outcomes and achieve success.

Section 2.2. Database and Query Optimization

Looking back to the 1960's and 1970's, the first database systems were network or hierarchical models, and query optimization techniques were manual. New concepts on data retrieval and relational databases came to light when E.F.Codd published an IBM research article on a database titled "System R" (Greenwald, 2001). Influential work, such as System R and its join order optimization methods set the stage for later work (Sudarshan, 1999).

Early development of database technologies proposed that the programmer would navigate the database. The algebraic methods used to query were replaced by a more simple syntax, such as "UPDATE", "INSERT", or "DELETE" and would encourage management to do their own queries (Haigh, 2006). Today a simple and standard language for running queries is SQL.

Initial use of SQL in System R (released as a research project by IBM in late 1974) relied on a model of cost based optimization algorithms (Blasgen et al., 1979). In 1979, Relational Software, Inc. or "RSI", (known today as Oracle Corporation, Inc.) released the first commercially available relational database system "Oracle V.2", and by the next year implemented SQL, ad-hoc query methods, and rule based optimization (Oracle, 2009). Since then, Oracle uses SQL to manipulate data, control transactions and access data through queries. Remarkably, IBM did not release a relational database product until 1983 (Greenwald, 2001).

Rule base optimization for the Oracle RDBMS was based on a set of predefined rules that made optimization decisions in early versions of the database. The rules were very simple and sometimes underestimated the complexity of orders and conditions in a large database with many tables and indexes to access. There were approximately 20 rules by which to base a query optimization decision on and with a complex query, the rule based optimizer did not always come up with the best option and performance suffered. Utilization of the rules included application of a "score" to each of the possible execution methods. The method with the best score was used to optimize the query. In the event of a tie in scores, a syntax statement was used to break the tie by choosing the first named table in the query. This had huge implications when the size of the tables varied widely and optimization was not always the end result. Critical to the system, effects on resources used to access the information was sometimes unpredictable (O'reilly, 2008). Rule based optimization methods in Oracle prevailed until the release of Oracle database version 7. Today, the rule based optimization method is no longer supported as of database version 10g, version 2. Oracle 10g, version 1 was developed with backward compatibility to help in transition from the rule based optimizer of earlier versions (Oracle, 2005).

The cost based optimizer set forth to improve the execution path, as well as minimize the number of logical I/O operations needed to return a result. The new approach by the cost based optimizer was a significant improvement since it was able to determine which table to use to initiate the query while economizing on resources. In order to employ the cost based optimizer, statistics about the data structures must be used, and include information about the database details. Statistics are summarized in table 2. 1. Historically, either setting up some procedures to demonstrate an automated statistical costing or using an SQL statement such as the ANALYZE command would invoke a statistics estimate for the cost based optimizer to use in evaluating a query. Statistics are stored in the data dictionary. Since Oracle version 10g, statistics are automatically gathered by the Automatic Workload Repository (AWR) (O'Reilly, 2008).

Statistics used by the cost based optimizer are categorized by two general types, the first being the CPU and I/O system performance statistics. With CPU and I/O system performance statistics, the best quality of access approach will be determined. The second type of statistics used by the optimizer is object statistics, which provides a description about the data and how it is physically stored. Object statistics identify the number of rows in a table, helping the optimizer determine if indexes are appropriate for use in the query (Foot, 2006).

Generation of statistics can take some resources as well, and may not be necessary. As every database administrator hopes, a time may come when the database and queries are in a well optimized or "steady" state (Oracle, 2005). As of Oracle 10g, the system has the capability of providing a "test" environment to employ newer statistics to see if improvements are relevant.

Table 2.1 Statistics Used by the Cost Based Optimizer,

Adapted from O'Reilly (O'Reilly, 2008).

Overall System Statistics	Data Structure	Type of Statistics
I/O	Tables▶	Numbers of Rows
CPU Performance	Number of blocks ▶	Number of unused blocks
CPU Utilization		Average available free space per block
		Number of chained rows
		Average row length
	Columns ▶	Number of distinct values per column
		Second lowest column value
		Second highest column value
		Column density factor
	Index ▶	Depth of index B*tree structure
		Number of leaf blocks
		Number of distinct values
		Average number of leaf blocks per key
		Clustering factors

Oracle's automated gathering of statistics is enabled with the later versions, beginning with version 10*g*. This automated function includes the ability to check for outdated or "stale" statistics by reviewing the tables for updates through monitoring.

If a table's updates constitute a greater than ten percent change, then the statistics are automatically updated. Automated statistics gathering ensures that statistics are updated appropriately, allowing the optimizer to utilize the information in determining the best query execution options. Figure 2.5 illustrates the effect of optimization choices as they relate to joins, table size and I/O's.

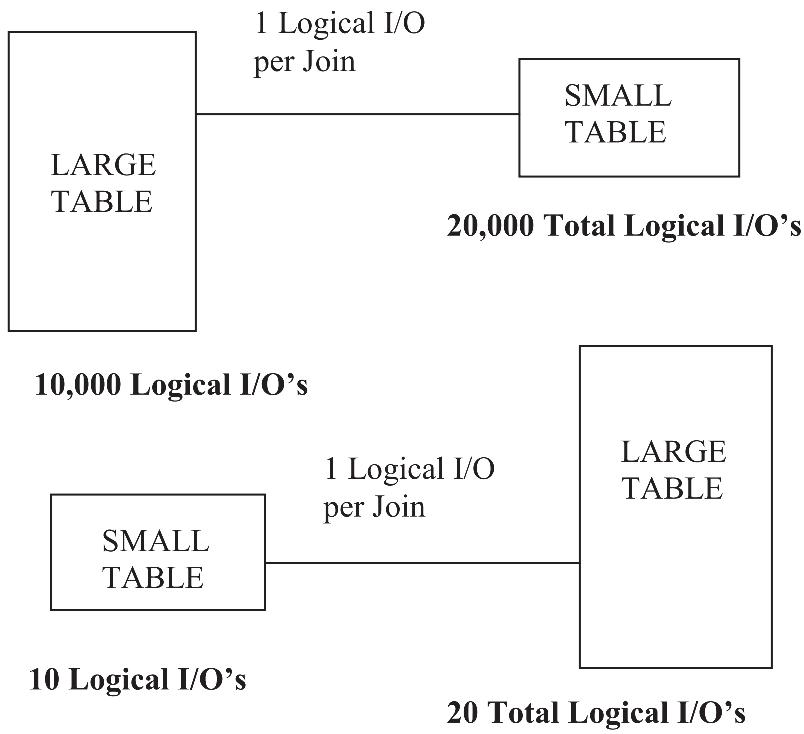

Figure 2.5. Effect of Optimization Choices (O'Reilly, 2008)

While the cost based optimizer chooses the best approach in optimizing a query using the lowest cost, its utility is hidden from the end user. To date, there are four high level elements of query optimization. First, there is the method of taking an SQL statement and improving it for efficiency. Second, there is defining an execution plan through the EXPLAIN PLAN view which shows the step by step execution of the SQL statement that was executed. Third, there is provision of the cost estimates (which have been discussed in detail). Lastly, there is dynamically optimizing runtime based on current state of the database (using system statistics).

Star Transformation is a cost estimated type of query optimization method and utilizes bitmap indexes and join-back elimination to achieve efficient access to fact tables (Oracle, 2005).

Specific issues related to query optimization include data type issues, unnecessary sorting, poorly defined indexes, incorrect join order, lack of parallelism and non-existent partitioning for query selectivity. A review of query optimization issues and their resolution by description follows. Data type issues (characters versus digits) can trigger full table scans instead of the preferred indexing method. This is guaranteed to cause significant delays in query response. Unnecessary sorting occurs when an SQL statement ORDER BY is used to sort data when a GROUP BY statement would have returned a more efficient result with less rows and desired information as an aggregate. Indexes must be well defined and can be helpful with query response time. If the index is built in the order most likely to avoid sorting, the GROUP BY and ORDER BY clauses will not use resources to sort. The optimally refined index will eliminate the need for a table look-up because the query is satisfied with the index. B-tree and bitmap indexes are types of indexes that can improve query response by finding data in a table quickly where there are many rows and value types. Join order initially was limited to five tables or less for the optimizer to correctly handle joins. Later versions of Oracle (7.3 and later), introduced join methodologies that improved join ordering (e.g., nested loops, sort merging, and hash joining). Parallel execution of SQL statements can improve performance dramatically as long as single threading (multiple processing scan on a single table) does not occur. Partitioning of tables by specific keys allows a query selectivity, which returns a faster query response (Holdsworth, 1997).

To bring the concepts of BI and query optimization together, it's important to emphasize the interrelationship that exists. Measurement of management methodologies and use of BI relies upon query optimization. BI is a significant area of interest and surveillance. BI directly correlates with data utilization and effectiveness. BI attributes are surveyed on a regular basis through the BI Survey, a market research report which seeks to identify why organizations choose BI, how they use it and if it is effective for their organization.

The 2007 BI Survey, conducted by Nigel Pendse through the Business Application Research Center, provides important quantitative information on a sample of 1,900 users globally. The survey covered information about applications, utilization, users, and practice patterns. An astonishing 87% of respondents recommended wide deployment of applications and users, while only 38% reported that they were adding licenses for additional users. The deployment and license report supports the trend of additional users. The most frequently reported technical problem was slow query performance (21%). Respondents reported slow queries as the most serious problem they encountered in BI development. Company politics was the most frequently reported people related problem, but is a less frequently reported problem from previous surveys (Pendse, 2008).

Database development and the technology, along with the data that is utilized in business, drives the improvement efforts that have evolved over the last few decades. Table 2.2, adapted from Thearling's timeline, provides an overview of how technology, data mining and modeling techniques have been developed, coinciding the application to business information needs. An important note about database development is that most OLTP systems will not allow system "read" (ad hoc queries typically done in a data warehouse) while system "writes" (transaction updates in OLTP) are occurring. Oracle provides a unique feature that allows simultaneous read and write functionality of the OLTP, eliminating the need for a separate data warehouse. Unfortunately, many system vendors would prefer that a separate (and expensive) investment be made to separate the OLTP system and the data warehouse. Maximizing revenue by requiring separate systems may contribute to the lack of concurrency.

The direct connection between the database, query response and the need for timely access to information in the business setting, requires sophisticated coordination of IT and business objectives. Overlapping business units within the organization and management initiatives can help to drive collaboration. Working in silos will not help in the effort to improve and succeed in business today. Alignment is necessary and critical for using information to drive strategy (Nugent, 2004).

Table 2.2 Overview of technology, data mining, modeling and business application timeline, adapted from Thearling's timeline (Thearling).

Database Development	Business Application	Supporting Data Technology	Data Approach
Late 1940's - 1960's: Data Collection, Storage	Totals, e.g., amount of financial information for given time period	Mainframe Computing, tapes, disks	Historical data, with simple, consistent data types
1979 & 1980's: Data Access, updating and retrieving with SQL	Totals, e.g., amount of financial information for given time period and its relationship to a specific entity	Relational Database Management Systems (RDBMS) & Structured Query Language (SQL)	Historical data, with capabilities in retrieval of record level information
Late 1980's & 1990's: Data	Totals, e.g., amount of financial information for given	Online Analytical Processing (OLTP),	Historical data, complex algorithms, multi-level drill down

Warehousing, Dynamic Reporting & Decision Support Systems	time period and its relationship to a specific entity along with trending and support for decision making, dashboards & scorecards	multidimensional data, data warehousing	and modeling techniques introduced
2000 and Today Business Intelligence &	Operations, Key Performance Indicators (KPI's), real time data availability for quick decisions & strategy changes	Massive (terabyte) sized data warehouses, Online Transactional Processing (OLAP), Integration of data sources from disparate systems	Rapid cycle data analysis, real time data analysis, advanced modeling techniques and strategic use of data (e.g., just in time supply chain management)

The overall intent of the database, supporting systems, and efforts to improve them is to store data, update it when necessary, organize it logically, ensure accuracy of content, and provide the ability to access the data for analysis with regularity. There must be some assurance and predictability about how and when a user can expect to retrieve the information. Likewise, a query return must be prompt and consistencies in run times should provide assurance for the end user that the information will be accessible. The predictability of information retrieval will help to accurately anticipate report delivery. Oracle's advanced methodologies for query optimization can help achieve optimal query performance. The review of the literature demonstrated that BI and associated query performance issues supports a thorough discussion on current best practice for query optimization in Oracle, Star Transformation (Oracle, 2003).

Chapter 3 – Methodology

A complete review of the data warehouse as it relates to the logical operations to be undertaken in developing Star Transformation was conducted. A discussion of implementation requirements is detailed. In addition, an interview with a subject matter expert in Oracle database administration was done to evaluate the results and outcomes, along with the implications of achieving Star Transformation in a test environment. Provision of example input and output of the SQL query is included.

The methodology assumed that a traditional OLTP system is in place and a separate data warehouse is available for running ad hoc queries by the mass deployment of business users to run queries against it. The separate data warehouse environment is necessary, even with the Oracle RDBMS' unique functionality which can maintain concurrency in its OLTP system, allowing for transaction updates (write functions) and queries (read functions) without one waiting for the other (Greenwald, 2001). Note that experienced programmers generally are running the complex queries against the OLTP system and these are known reports run with regularity. The Star Transformation query

optimization method requires a denormalized schema, versus a normalized or third normal form schema (which reduces duplication through normalization and contains more tables). The database schema for this methodology focuses on a partially denormalized schema as well. Essential to the development of Star Transformation is the use of the Oracle's Cost Based Optimizer (Scalzo, 2003, p. 108). The decision to focus the methodology on a partially denormalized schema was based on the notable differences between the traditional OLTP system and the data warehouse. Table 3.1 summarizes the differences in order to provide rationale for the methodology, adapted from Lane's Oracle 10g, release 2 Data Warehousing Guide (Lane, 2005).

There are experts who suggest that the normalized data warehouse is better, with partially or fully de-normalized data marts. The rationale is that business requirements are best understood at a fine level of detail, in a normalized layer. Without the detail, there will ultimately be errors and inconsistencies in the data. Problems with updates and purges can occur if the read/write functions cannot occur simultaneously.

The star schema in a partially denormalized data warehouse was reviewed in preparation for the construction of Star Transformation. In review of the star schema design, the star schema actually resembles a star, is a simple and logical view of the business to which the end user can best relate. Now this appears to be too good to be true, but it is true that all of the benefits of the Star Schema outweigh any risks that could be identified. The star schema is by far the simplest type of schema for a data warehouse.

Table 3.1 Summary of Differences between OLTP and a Data Warehouse System adapted from Oracle Data Warehousing Guide (Lane, 2005)

Requirements	OLTP	Data Warehouse
Workload	Usually supports on pre-defined queries	Uncertain workload, supports ad hoc queries, more variety
Data Modifications	Routine access by end user, consistently updated, reflective of current business transactions	Standardized bulk updates, scheduled on a regular basis and no end user access
Schema Design	Normalized schema (3rd Normal Form, “3NF”) for transaction performance and consistent data	Schema design is de-normalized or partially de-normalized to support Star Schemas
Operation Type	Built to access small number of records for the purpose of checking a record	Built to access many rows, to support the ad hoc query
Historical Data	Regular purging and short term data is kept with limited historical information	Long term storage for later retrieval that supports an end user & ad hoc queries

The ERD actually looks like a star configuration where the fact table resides in the center and is surrounded by dimension tables that are denormalized. The fact table is a large table with foreign keys and the dimension tables are considerably smaller, and include information about the details in the fact table, including descriptive, hierarchal, and metric attributes (Oracle, 2005).

Figure 3.1 demonstrates the star schema, its fact table and surrounding dimension tables in a diagram. Star schemas can be used for large data warehouses as well as small data marts.

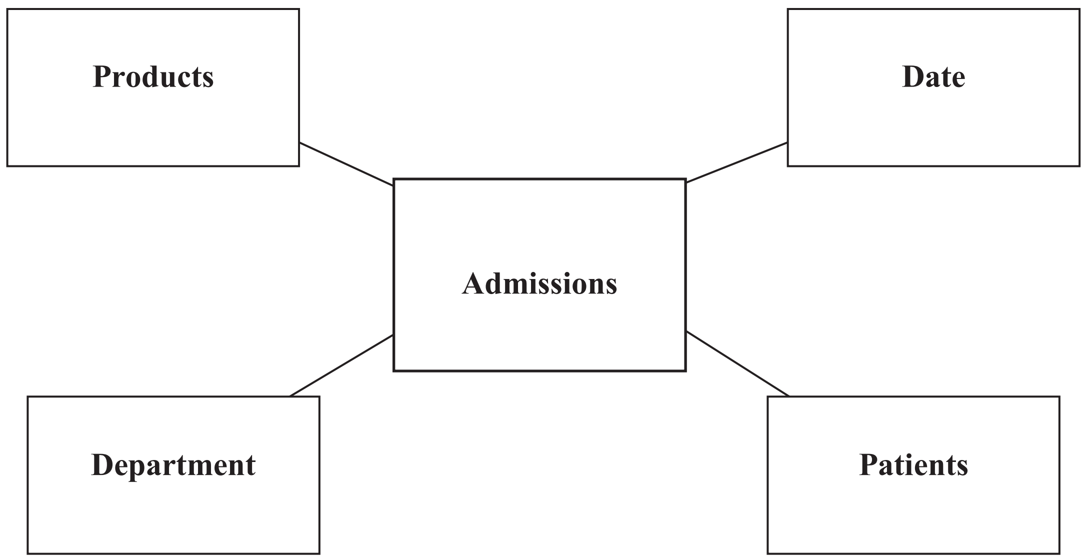

Figure 3.1 Star Schema Diagram adapted from Oracle, (Oracle, 2005)

There are many advantages to the star schema, as this dimensional modeling technique provides for efficient query optimization. The star schema provides easy, insightful mapping of business units through the schema design for the end user. Star schemas are the best way to optimize queries because the dimension tables are a necessity for BI applications to work effectively (Borysowich, 2007). Star schemas facilitate the star query, where the center of the star, the fact table, joins with the outer points of the star, the dimension tables. The star join in a star query occurs by means of the primary keys of the dimension tables to the foreign keys of the fact table (Oracle, 2003).

Another type of star schema exists and this is referred to as a snowflake schema. The snowflake schema design is more complicated and its physical design is similar to that of a snowflake. Snowflake schemas require that the dimension information is stored in many dimension tables instead of a single large dimension table that the star schema references. Figure 3.2 diagrams the snowflake schema where an additional dimension table is appended. The additional dimension table gives the look of a snowflake. The dimension tables of the snowflake schema are normalized to eliminate redundancy. This may seem to make the snowflake schema more efficient without the redundancy; however more dimension tables require more foreign key joins, which in the long run reduces efficiency and can contribute to delays in query run times (Oracle, 2005).

In a star schema, the dimension tables do not have parent tables but in a snowflake schema the dimension tables can have one or many parent tables. Because there are dimension tables that have additional hierarchies in a snowflake schema, there are more tables to join and a subsequent delay in query response will occur. Oracle recommends that star schemas be used (Oracle, 2005).

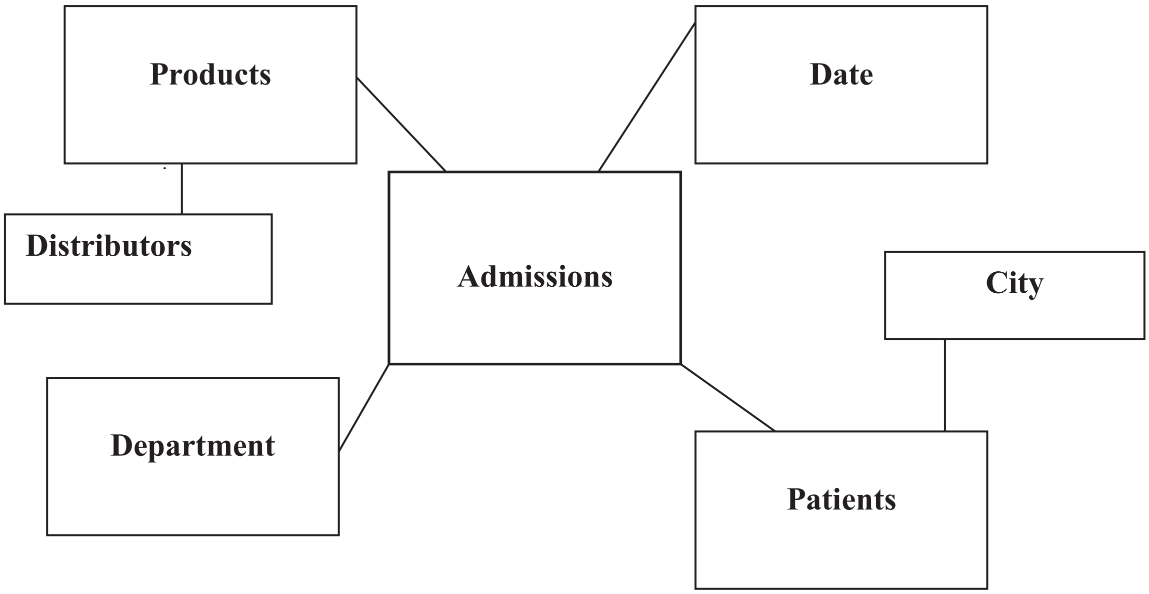

Figure 3.2 Snowflake Diagram (Oracle, 2005)

For the end user accessing the data, a star schema provides straightforward access because it simply reflects the relationships of the business information they are trying to retrieve. Although the star schema is simple for the end user, it can be complex to design and maintain for the DBA or data warehouse developer (Borysowich, 2007).

Optimization and tuning techniques for star queries have some basic requirements. In order to best tune a star query and enable the feature for star query execution in the optimizer, a bitmap index needs to be built on each column of the fact table. Next, setting the initialization parameter STAR _TRANSFORMATION_ ENABLED to TRUE provides the elements necessary to achieve the transformed query "Star Transformation". Whenever Star Transformation is needed and the conditions have been met, the optimizer will automatically choose it. Star Transformation essentially re-writes the original query from the end user. The query is processed by accessing only the necessary rows of the fact table and then joins these rows to the dimension tables for the result. The re-write change will go unnoticed by the end user, but query optimization will be robust (Oracle, 2005).

While the optimization technique seems simple and effective, it is important to understand how the bitmap indexing component of Star Transformation works. When a bitmap index is used, it directly points to the rows in the queried table for a key value. For each key's list of rowid's, the bitmap is substituted. An automated mapping process can convert the bitmap back to the rowid. Each bit corresponds to a row and its key value when it is set. Bitmap indexes are condensed in comparison to regular indexes and thus require much less storage, processing and memory, resulting in improved query performance (Lane, 2005).

Star Transformation works by using the fact table's join column of foreign keys that include a single column bitmap index. As an element of a two part procedure, in the initial phase, the foreign key column's bitmap indexes bring in only those rows that are needed from the fact table. In the next phase of Star Transformation, the query is essentially re-written in more simplified terms via the bitmap indexes. The re-writing of the query is where the "transformation" takes place. Figure 3.3 demonstrates an example of the two part transformation of a star query for patients admitted to the general hospital by department for quarters three and four, adapted from the Oracle 10.2 Warehousing Guide (Lane, 2005).

There are benefits to both the database team and the end user by optimizing a query with Star Transformation. The end user benefits by writing a complex SQL statement and is able to retrieve the requested information in optimal time. Although the end user does not realize that their query was optimized, they can realize the benefit of asking a complex question and receiving a quick answer. The database team can also realize the benefits because minimal resources were used in the ad hoc query.

Star Query from end user requesting information about patients admitted to a hospital by department 3rd and 4th quarters.

Initial Star Query (1st step):

```
SELECT de.deparmtnet_class, p.pat_city, d.calendar_quarter_desc,
   SUM(s.quantity_admited) admission_quantity
FROM admission a, date d, patients p, departments de
WHERE s.date_id = d.date_id
AND   s.pat_id = p.pat_id
AND   s.department_id = de.department_id
AND   p.pat_city = ‘DENVER’
AND   de.department_desc in (‘General’,’Inpatient’)
AND   d.calendar_quarter_desc IN (‘2009-Q3’,’2009-Q4’)
GROUP BY de.deparmtnet_class, p.patient_city, d.calendar_quarter_desc;
```

Transformation of the original query utilizing bitmap indexes to rewrite a simpler query that is optimized.

Star Transformation (2nd step):

```
SELECT … FROM patients
WHERE date_id IN
  (SELECT date_id FROM dates
   WHERE calendar_quarter_desc IN(‘2009-Q3’,’2009-Q4’))
   AND patient_id IN
  (SELECT patient_id FROM department
WHERE patient_city =’DENVER’)
   AND department_id IN
  (SELECT department_id FROM department
WHERE department_desc IN(‘General’,’Inpatient’));
```

Figure 3.3 “Transformation” adapted from Oracle’s Data Warehousing Guide 10.2 (Lane, 2005)

At a glance, it can be determined that the example of transformation in figure 3.1 shows a more simplified query in the second part of the transformation process. Note that the WHERE clause is lengthy in the first part and the bitmap index function is leveraged on this part of the transformed, re-written query. The optimizer will produce this transformation when it has been determined to be the optimal plan. Star Transformation can be verified through the EXPLAIN PLAN. In order to request an EXPLAIN PLAN, the script provided by Oracle is executed to create the PLAN_TABLE, which will contain the EXPLAIN PLAN result, detailing how the query will be run. The details provided in the table result include the steps that will be used in executing the query (Oracle, 2005).

Some issues with queries and tables will prevent Star Transformation from being executed by the optimizer. It's important to include a few of the specifics about the issues preventing Star Transformation in order to gain an understanding of the limitations. Small tables evaluated by the optimizer do not select the option of Star Transformation because of their lower cost. Similarly, tables that have a direct access path will not utilize Star Transformation. Bind variables within queries will not be transformed, as well as a query containing a hint that is not compatible with the bitmap index. Fact tables without enough bitmap indexes will be excluded as well as those that reside remotely (in contrast to dimension tables that are remote).

The optimizer will only execute a transformed query if it has determined that it is the best option with the lowest cost. The optimizer utilizes the best plan possible. The specific join method is also chosen by the optimizer, and as illustrated in the re-written query example, the hash join is efficient when there are joins needed for dimension tables and it's the most likely join method to be chosen (Oracle, 2005).

A case study was developed for the methodology by interview and personal communication. The case study describes an actual project to implement Star Transformation in a test environment for a large corporation with a terabyte sized database.

The data warehouse team included a staff DBA, a database developer and a data modeler for the payroll department. The environment contained several billing fact tables and multiple dimension tables with thousands of indexes. Performance problems drove the decision to implement Star Transformation. The data warehouse was Oracle version 9. The fact tables were partitioned and the dimension tables were not. Thousands of indexes, many of which were not being used, seemed to be the cause of the performance problem. The data warehouse team met to brainstorm and discuss the problem in an effort to identify if potential improvements or changes could be made and they prioritized their options.

The database environment to be optimized was actually a data mart for a subset of financial information. Built in a star schema, the data mart was de-normalized and ready for optimization. Each member of the team had expertise in the necessary areas; the DBA knew what database changes needed to be made, the developer knew the content of the data, the modeler knew the constraints for the indexes. Decisions were made about setting the initialization parameters, table modifications that were necessary, the limitations on the scope of work, and how many fact tables would be affected.

Due to all of the partitions and thousands of indexes on fact tables the team needed to keep the scope limited to five fact tables. The team decided to fine-tune what they had, since the data mart was denormalized and the primary keys were in front of them. They made sure that there were unique primary keys on dimension tables. Initial modifications were made by setting the initialization parameters. The Star_Transformation_enabled=TRUE was set and WHERE clauses were used in the queries. Bitmap indexes were built on the fact tables. There were some existing B-tree indexes on the fact tables, which were left in place. Star Transformation did not work. This was due to the “mix” of b-tree and bit map indexes. They went back to the fact tables and eliminated the existing B-tree indexes and replaced them with bit map indexes which then enabled Star Transformation. The project, from start to completion, took approximately six weeks.

Challenges reported for the project included the rebuild of the bitmap indexes after trying to partially use what was available (B-tree indexes) and gaining support from the business unit leadership to fully implement Star Transformation in production. A recommendation was made by the project DBA to apply Star Transformation to data with low cardinality. Low cardinality data is data that incurs infrequent updates. In addition, development and testing was done in a test environment and not in production because the end users rely on production availability. It was important to ensure that the data was clean by validating it during loading. The optimizer had current statistics to avoid significant access path changes. After the test project, no further implementation in production was carried out, due to lack of support from the business executives (Personal communication with interview subject, see Appendix A for Institutional Review Board (IRB) exemption).

The work to be completed when implementing Star Transformation requires a coordinated effort of individuals with knowledge, experience and appropriate access to the data warehouse environment. Some considerable time is needed for modeling, creating the bitmap indexes, and implementing the transformation against queries. For this reason, the database administrator and the IT unit for the data warehouse must collaborate with the business leadership to ensure that objectives and time frames for the project are agreed upon. Regular workloads for the data warehouse team may be offset by the project, creating a need for supporting roles to complete necessary production tasks.

Considering that the end user may have little concern over query response time until the moment there is a delay, the project may not be a priority if planned in advance to be proactive. In addition, since the Star Transformation is happening in the background, a user may not readily notice an improvement if no apparent problem was perceived initially. It may be that during a period of poor query performance, the issue is apparent to the end user and project support might be obtainable. This reactionary approach may not provide for the time needed for modeling modifications, indexing work or testing the transformation process outside of the production environment. This is where the business objectives must be carefully weighed against a query optimization project.

Approvals and support from leadership for a modeling and transformation project can be difficult to obtain. Communication about the project, presented to the approving leadership, should include the risks, benefits and return on investment. Technology initiatives must provide a comprehensive cost analysis, showing return on investment by way of financial justification. This is a critical step to ensure that resources (hardware, software, and people) are allocated for the project. A clear association between the project and resulting business success must be evident and defined within a formal proposal. All aspects of the project should be formally presented, and confidence in leveraging and maintaining the expected business results long term must be demonstrated. The investment of human and financial resources in a struggling economy can prevent the data warehouse team from embarking upon an optimization project. (Stackowiak, 2007).

Chapter 4 – Project Analysis and Results

A de-normalized database, along with a simple star schema, appears to be an easy start for a Star Transformation project. The benefits of improved query response time by reducing the number of rows needing to be scanned are evident. It's clear that the re-write of the query during the second part of the transformation simplifies the execution plan. It's been shown that star schemas are simple and are preferred for ad hoc queries. Ad hoc queries are intended to best support management initiatives and BI for strategy and gaining the competitive edge. From the literature review and methodology, Star Transformation can be realized. There is a high level of guarantee for success in significantly reducing query response time and satisfying the end user. Star Transformation closely aligns with business objectives intended for information analysis. Oracle consistently improves on its optimization strategy and has patented the Star Transformation join process.

The implementation of Star Transformation has been demonstrated to be fairly simple and straightforward and can be done successfully. The simplicity and success of implementation is based on the concise requirements detailed in the literature and simple methodology, along with the case study of an actual implementation in a test environment. The most difficult challenge to overcome is administrative buy in and allocation of resources to complete the project.

Political issues can be another challenge. Financial justification and improved business results are important. Gaining support from the political side for a resource intensive project can be overwhelming. Sometimes, having key information on personal individual or group agendas for prioritization of business objectives can help focus the strategy for gaining political support. Alignment of business objectives, clear communication and financial justification can be helpful, but competing priorities and timing can limit the ability to allocate resources for technical improvements such as Star Transformation.

Actual implementation of Star Transformation in an Oracle RDBMS was not done for this project. Implementation would have included using SQL to create the bitmap indexes, setting the “star_transformation_enabled=TRUE”, and actually querying a database to request the information about the admitted patients to a department in the time period given. Actual implementation was not possible due to the project being conducted in a theoretical way instead of hands-on practicum experience. The disadvantage of this is lacking the input and output experience with the query.

Trial and error experience was not available; however this is demonstrated in the case study when the DBA describes partial use of B-tree indexes, and subsequently creating all bitmap indexes on every column of the fact table. Some literature reassures that it is easy to remove all the indexes and rebuild them, which had not been discussed in depth of the case study or the supporting information in the methodology.

Theoretical work, along with a hands-on practicum may have better emphasized the methodology and case study to compare and contrast the experiences. Although this was a limitation of the methodology, a thorough understanding was achieved through the personal communication and case study.

Chapter 5 – Conclusions

Improvements in the technology and processes used to build, process, store and retrieve information have grown by leaps and bounds over the past four decades. Database technologies continue to improve and are influenced by the foundations from which they emerged. The majority of businesses will continue to base decisions on historical and new information contained in the database. The end users who query and conduct analysis will likely remain oblivious to their impact on the systems they access but expectations of timely access will be constant.

Efforts to maintain efficiency through optimization methods will continue to evolve. The ability to run complex queries for BI will continue to provide many companies with a competitive edge. Efforts to facilitate ongoing competitive success, especially in a customer driven environment, necessitate the efficient utilization of technology, people and best practices.

In the world of databases and data warehousing, there must be a constant review of architectures, schema design and operational issues. Database priorities should be determined with careful thought and consideration towards throughput of the query while conserving resources. This can be accomplished by application of best practices, those that have been tested and implemented with a minimum of unintended consequences. Collaboration between experts and business units for database development and modeling can ensure that a true picture of the business and its intended objectives are incorporated into the data warehouse design. Alignment of priorities should drive initiatives in a collaborative effort between technology and business units.

Sponsorship from executive leadership is critical in getting project approval and backing. All aspects of the project should be discussed and confidence in leveraging and maintaining the expected business results long term must be demonstrated up front (Stackowiak, 2007).

Lessons learned from the project include identifying differences with database versions by fully understanding the previous version as the Oracle improvements seem to significantly change (for the better) with each new release. Knowledge of changes between versions can help in understanding improvements such as automation of old features and introduction of new functionality. Lessons learned from the case study, an informal project, may provide for the most important aspect of embarking on this type of optimization project. While the improved performance reached success in a test environment, it was stifled by the lack of support from the business leadership based on prioritization of production needs. Perhaps a presentation showing factors of success in support of a formal proposal would shift priorities.

One of the key lessons for me as the writer was that in order to fully understand the vast amounts of complexity in just one method of query optimization, let alone all of them, one must have a very good understanding of development milestones and each version of the Oracle database to be able to fully understand, interpret and explain the techniques to others.

Thoughts for future research include strategies to better align IT with business units and attaining improved visibility and integration among the end user community, so that each entity may gain a better understanding of the other. Long term research should include development on the provision of "built in" optimization strategies that do not require an implementation project to realize the advantages of best practice.

References

Blasgen, M., Astrahan, M., Chamberlain, D., Gray, J., King, W., Lindsay, B., et al. (1979). *System R: An Architectural Update* (Computer Science IBM Research Report RJ2581). San Jose, California.

Borysowich, C. (2007, November). *Star schema modeling (data warehouse)*. Retrieved On December 6, 2009 from http://it.toolbox.com/blogs/enterprise-solutions/star schema-modelling-data-warehouse-20803

Brookes, C. (2007). *A structured method for specifying BI reporting systems.* Retrieved On March 24, 2009 from http://www.information-anagement.com/white_papers/ 2220457 -1.html

Few, S. & Ross, M. (2004, March). *Dashboard confusion.* Retrieved on September 17, 2009 from http://intelligent-enterprise.information week.com/showArticle.jhtml;jsessionid=5F1YE14ONPPNXQE1GHOSKH4ATMY32JV N?articleID=18300136

Flanglin, R. (2005, June). *BI 101*. Retrieved on October 17, 2009 from http://www.powerhomebiz.com/062005/intelligence.htm

Foot, C. (2006, February). *10g optimizer statistics gathering.* DBAzine. Retrieve on October 13, 2009 from http://www.dbazine.com/blogs/blog-cf/chrisfoot /blogentry.2006-02 11.0939245208

Giles, L. (2009) Sun Tzu's the art of war, translation. Retrieved on October 17, 2009 from http://en.wikipedia.org/wiki/The_Art_of_War direct link to electronic media.

Gonzales, M. (2007). *BI: bringing analytics into operations.*Database Magazine, (vol. 12, no. 2). Retrieved on November 14, 2009 fromhttp://www.Ibmdatabasemag.com/shared/printableArticle.jhtml?articleID=201200513

Greenwald, R., Stackowiak, R., & Stern, J. (2001). *Oracle essentials: Oracle 9i, Oracle 8i and Oracle 8 (3rd ed.)*. Sebastopol, CA: Oreilly Media.

Haigh, T. (2006, June). *A vertical bucket of facts, origins of the data base management system.*

SIGMOID Record, (Vol. 35, No.2), 33-44. Retrieved on November 30, 2009 from http://www.tomandmaria.com/tom/Writing/VeritableBucketOfFactsSIGMOD.pdf

Hall, T. (2005) *Performance Tuning Enhancements in Oracle Database 10g.* Retrieved on November 24, 2009 from http://www.oracle-base.com/articles/8i/CostBasedOptimizer AndDatabaseStatistics.php

Holdsworth, A. (1997, January). *Data warehouse performance management:* Oracle services advanced technologies data warehouse practice. Retrieved on October 10, 2009 from https://www.indiana.edu/~dbateam /databases/oracle/dwperf.pdf

Lane, P. (2005, December). Oracle data warehousing guide 10*g* release 2 (10.2). Retrieved on October 31, 2009 from http://download.oracle.com/ docs/cd/B19306_01/server102/b14223.pdf

Nabitz, U., Klazinga, N., & Walburg, J. (2000). *The EFQM excellence model: European And Dutch experiences with the EFQM approach in health care.* The International Journal for Quality in Health Care 2000, (vol. 12, no. 3, pp. 191-201).

Netezza (2004, September). *Business intelligence in a real-time world.* Netezza Corporation white paper. Retrieved on March 24, 2009 from http://download.101com.com/pub/tdwi/files/BI_in_a_Real-Time_World_Sept04.pdf

Nugent, M. (2004, December). *The four phases of business/it alignment.* CIO Update. Retrieved on December 12, 2009 from http://www.cioupdate.com/insights/ article.php/3446591/The-Four-Phases-of-ITBusiness-Alignment.htm

Oracle Corporation, Inc. (2009). Oracle Timeline. Oracle Corporation, Inc. Web site. Retrieved on September 20, 2009 fromhttp://www.oracle.com/timeline/index.html

Oracle (2008, September). "*Management excellence: how tomorrow's leaders will get ahead", an Oracle thought leadership white paper.* Retrieved on March 24, 2009 from http://www.oracle.com/solutions/business_intelligence /index.html

Oracle (2003). Oracle Database data warehousing guide, 10*g,* release 1. Retrieved on September 16, 2009 from http://download.oracle.com/docs/cd/ B14117_01/server.101/b10736.pdf

Oracle (2005, June)**.** *Query optimization in Oracle Database 10g release 2.* An Oracle White Paper. Retrieved on October 17, 2009 from http://www.oracle.com technology/products/bi/db/10g/pdf/twp_general_query_optimization_10gr2_0605.pdf

Oracle (2005), Schema modeling techniques, Oracle data warehousing guide 10*g* release 2 (10.2), Retrieved on November 27, 2009 from ttp://download.oracle.com/docs/ cd/B19306_01/server.102/b14223/schemas.htm

O'reilly Media, database articles (2008, February). *Query optimization in Oracle.*

Retrieved on November 30, 2009 from http://www.codewalkers.com/c/a/Database-Articles/Query-Optimization-in-Oracle/2/

Pendse, N. (2008, June). Results from the BI 7 survey: excerpt from DM Review Special Report,June 3, 2008. Retrieved on 12/16/2008 from http://www.information management .com /specialreports/ 2008_81/10001298-1.html fromhttp://www.thearling.com/index.htm#wps

Peters, B. (2009, October). *BI for the front lines.* Information Management. Retrieved on November 10, 2009 from http://license.icopyright.net/user /view FREE Use.act?fuid=NTgzNQ%3D%3D

Pragmatek (2001, June). *How to SCOR with six sigma.* Pragmatek Consulting Group White Paper. Retrieved on March 24 from http://jobfunctions.bnet. com/abstract.aspx?docid=60326&tag=content;col1

Quest (2006). Toad Data Modeler, Quest Software, Inc. Retrieved on September 17, 2009 from http://modeling.inside.quest.com/entry.jspa?externalID=2054

Raden, N. (2008, May). *Fast analytics and BI for everyone: best practices for deploying collaborative bi*. Retrieved on March 24, 2009 from http://www.hiredbrains.com/Tableau.pdf

Raymond, E. (2000). *The cathedral and the bazaar.* Thyrus Enterprises. Retrieved on November 29 from http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/

Rittman, M. *Building BI dashboards with Oracle 10g*. Retrieved on September 17, 2009 from http://www.oracle.com/technology/pub/articles/rittman _dash.html

Rosenberg, A. (2006). *Improving Query Performance in Data warehouses.* TDWI Business Intelligence Journal. Retrieved on June 8, 2009 from http://www.tdwi.org/Publications/BIJournal/display.aspx?ID=7891

Stackowiak, R., Rayman, J., & Greenwald, R.(2007). *Oracle data warehousing and business intelligence solutions.* Justifying projects and claiming success (p. 330). Indianapolis, Indiana: Wiley Publishing, Inc.

Sudarshan, S. (1999, October). *Brief Overview on Query Optimization.* Presentation given at the workshop on query optimization, IIT Bombay, CSE Department. Retrieved on September 20, 2009 from http://www.cse.iitb.ac.in /dbms/Data/Conferences /IITBDB Workshop/index.html

Surajit, C**.** (1998). *An overview of query optimization in relational systems.* Microsoft Research. Retrieved on November 27, 2009 from ftp://ftp.research.microsoft.com/users/surajitc/pods98-tutorial.pdf

Thearling, K. *An introduction to data mining.* Retrieved on March 24, 2009 from http://www.thearling.com/

Appendix A: Survey Interview Questions

Interview Questions to be asked, the majority are open ended to elicit as much information as possible, specific to the topic, from the subject matter expert in their own terminology. Response will be written down and not recorded.

1. Name
2. Title
3. Department
4. Company
5. Role in Business Intelligence (BI) or Star Transformation (as applicable to subject matter expert)
6. Describe the type of information you work with.
7. Describe in your experience, performance issues with queries.
8. How are decisions made to improve performance issues and can you provide a specific example of the decision making process as applicable to efforts in optimizing queries or Star Transformation?
9. Describe a team that you have worked with in evaluating a query run time problem or BI problem
10. Describe the timeline for any development efforts.
11. What type of database do you work with, and what version?
12. Is the database normalized?
13. If Star Transformation was used, describe the project in terms of:
    a. Bit map indexing
    b. Initialization Parameters
    c. Hints (SQL)
    d. Other important aspects
14. Describe major challenges.
15. Describe successes.
16. Describe outcomes.
17. Are there any "key" take-away information you would like to share?
18. Do you know of any future efforts or challenges?

Thank you for your time.

4. Risk/Benefit assessment. Assess the risks and potential benefits of the investigation.

Risks of participation in the interview are minimal and included confidentiality of answers, specifically in naming the participant as interviewee. Only the researcher and the faculty advisor for the project, and the Regis IRB will have access to the names of the interview participants. There are no tangible benefits to participation, however the information may be beneficial to the research as well as students and faculty, and whoever reviews the published research.

5. Provision for informed consent. Provide details of informed consent procedures to be used, including samples of project descriptions to be given to subjects and consent forms to be used.

6. Additional ethical considerations. Describe provisions for anonymity or confidentiality and any additional measures not previously addressed taken to protect the rights and safety of subjects.