Winter 2010

Evaluation Real-Time Data Warehousing Challenges From A Theoretical And Practical Perspective

Dale Hargens
Regis University

Regis University
College for Professional Studies Graduate Programs
Final Project/Thesis

Disclaimer

EVALUATION REAL-TIME DATA WAREHOUSING CHALLENGES FROM A THEORETICAL AND PRACTICAL PERSPECTIVE

A THESIS

SUBMITTED ON 15 OF DECEMBER, 2010

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 DATABASE TECHNOLOGY

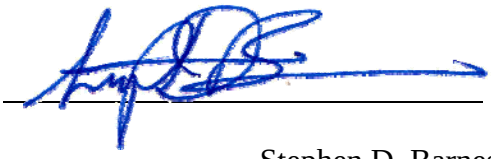
BY

DALE HARGENS

APPROVALS

Joan Lawson, Thesis Advisor

Shari Plantz-Masters



Stephen D. Barnes

Abstract

The concept of real-time data warehousing has grown in popularity in recent years as organizations demand access to critical pieces of data in real-time to produce analytics and make business decisions to gain competitive advantage. Real-time data warehousing systems differ substantially from traditional data warehousing systems, thus, presenting a unique set of organizational and operational challenges. The basis for the research was to investigate whether adequate information is available regarding the organizational and operational challenges of real-time data warehousing and whether that information is available to the database community. This exploration was done by gathering primary research, conducting a case study research design, and comparing, analyzing, and drawing conclusions from the two different types of research. The information for the primary research was gathered from scholarly, peer reviewed articles, and books on Google Scholar and ACM, computer science and database journals, various textbooks, and the Internet. The case study was conducted on an organization utilizing a real-time data warehousing system.

Acknowledgements

I would like to give thanks to my Heavenly Father for everything in my life. I would also like to give a special thanks to wife Angela, parents, father-in-law, mother-in-law, advisor Ms. Joan Lawson, and mentor and friend Mr. George De Graaf for supporting and encouraging me throughout my studies.

Table of Contents

List of Figures

Chapter 1 - Introduction

The purpose of this chapter is to acquaint the reader with an opening to the context and domain of the problem. It will provide the following background regarding the study: problem statement, statement of goals and objectives, description of the final product, and the thesis statement.

Problem Statement

A significant number of organizations in the profit and non-profit sectors rely heavily on data warehouses to bring together data from internal transactional systems and numerous external sources. The uses for the data brought into the data warehouse include the following: to grow the business, meet business objectives, support new product development, provide information to customers, and track customer feedback. These source systems often contain data from several departments within the organizations and sometimes even hold data from entirely different organizations, such as parent or child companies, competitors, and auditing firms. This central location of data is important to businesses because it enables stakeholders to gain quick accessibility for retrieval, manipulation, and analytical reporting on key business indicators that are crucial for the organization's decision makers, compliance agencies, taxing authorities, and auditing partners. Another equally important aspect of data warehousing is that it can allow stakeholders to view a particular portion of data in relationship to an entirely different segment of data.

Traditionally, data warehouses have contained a historical, point-in-time snapshot of an organization's data. The refreshing of this data has usually been completed in a batch process on a predetermined schedule, such as on a quarterly or monthly basis. Over the years, this update schedule has worked well for companies utilizing data warehouses, but recently, the need to

reduce the time interval gap has become increasingly significant, which has prompted the advent of real-time data warehousing. Real-time data warehousing allows an organization's stakeholders to use information from source systems throughout the company as transactions occur because data is instantly refreshed and made available in the data warehouse.

The demand for real-time data warehousing has gained momentum because companies need to make decisions quicker.

> Business time is increasingly moving toward real-time. As organizations look to grow their competitive advantage, they are trying to uncover opportunities to capture and respond to business events faster and more rigorously than ever. Today, the majority of competitive advantage comes from the effective use of IT. Therefore, from that standpoint, the key to achieving faster business intelligence (BI) is a robust enterprise data warehouse combined with an enterprise analytics framework.
>
> Across the enterprise, each facet of the business gathers data through an assortment of activities, and many organizations now deliver this data to a central data warehouse where the data is captured, aggregated, analyzed, and leveraged to improve decision making. The quality of these decisions depends not only on the sophistication level of the analytics applications that run on the data warehouse, but also on the underlying data. Data has to be complete, accurate, and trusted. For that reason, it has to be timely: timely data ensures better-informed decisions ("Real-Time Data Integration for Data Warehousing and Operational Business Intelligence", 2010).

This demand is supported by improved technology capabilities, including improvements in computer hardware, software, and storage and the fact that organizations are leveraging the Internet to conduct business. The improvements and declining costs associated with computer

hardware, software, and storage have allowed the amount of data that organizations can capture and make use of to grow exponentially. The widespread use of the Internet permits business to be done in a much more efficient and effective manner. With these technological advancements, companies are now demanding the tools to make informed decisions faster, to not only keep pace with, but to pass their competition.

Real-time data warehousing is a new concept that has introduced a unique set of organizational and operational challenges. From an organizational standpoint, the real-time data warehousing issues include the following: executive sponsorship, financial support, and user support. From an operational standpoint, the challenges include the following: enabling real-time extracting, transforming, cleansing, and loading (ETL) of the data, modeling real-time data, On Line Analytical Processing (OLAP) queries and changing data, scalability and query contention, and real-time alerting. Whether or not these obstacles can be contended with in a timely, appropriate, and cost-effective manner has a large bearing on the adoption, acceptance, design, implementation, and use of a real-time data warehousing solution. The problem is whether adequate information is available to data warehouse designers, database administrators, and database developers regarding optimum architecture and proper administration of a real-time data warehousing system.

Statement of Goals and Objectives

The purpose of this research is to provide an in-depth examination of the organizational and operational challenges in real-time data warehousing. The research will begin by briefly describing the demand for data warehousing as it provides a foundation for the development of real-time data needs. The research will next provide a history of real-time data warehousing, including a review of past, current, and future trends. The research will identify the benefits and

limitations that organizations may encounter when delivering real-time data warehouses. The research will explore the instrumental challenges database professionals and researchers face and how these obstacles have or have not been addressed by the industry. The objective of the research is to evaluate current real-time data warehouse organizational and operational practices from a research and practical perspective, and then determine if and how these practices influence the overall performance and satisfaction of real-time data warehousing projects.

Description of the Final Product

The study adds value to the entire information technology industry, especially the database community and data warehousing field, because it addresses and evaluates how existing research conducted by experts compares to a practical example of a fully deployed real-time data warehouse by an organization in the retail industry. The research explores the organizational and operational challenges to real-time data warehousing and what solutions have been utilized to address these issues. In addition, this examination is crucial and valuable to the industry because the introduction of real-time data warehousing is a relatively new, so the amount of exploration into the topic is small. Finally, the exploration is significant to the industry due to the increased popularity, acceptance, and use of real-time data warehousing by organizations throughout the world.

The sections of the study include the following: historical investigation of data warehousing and real-time data warehousing, the literature review of organizational and operational challenges, the case study presentation, the comparison of pertinent research and practical real-time data warehouse organizational and operational challenges and solutions, and the analysis of the findings. These sections will provide a wealth of detailed information to researchers and students and current and potential users of real-time data warehousing solutions.

The study could provide researchers and database students with a solid source for conducting their own studies and research assignments. Organizations utilizing real-time data warehousing could draw on the findings to assist them with designing a new or maintaining an existing real-time data warehousing system.

Thesis Statement

Given the recent emergence of real-time data warehousing as a viable solution to gain access to an organization's valuable information from multiple data sources for competitive analytics and timely decision-making, how do the significant organizational and operational challenges influence successful implementation of real-time data warehouses within the database community?

Chapter 2 - Literature Review

The purpose of this chapter is to provide a historical perspective of the research objective by utilizing research from previous studies. This will be done in the following sections: data warehouse overview, real-time data warehousing trends, real-time data warehousing challenges, real-time data warehousing solutions, and real-time data warehousing perceptions.

Introduction

Evidence suggests that the success of implementing traditional data warehousing systems has fostered the desire for organizations to expand upon their capabilities by introducing systems that refresh in real-time. Only recently has the concept of real-time data warehousing been viewed by the industry as a plausible solution for an organization's data needs. The pages below will provide a review of applicable literature regarding the following: data warehouse overview, real-time data warehousing trends, real-time data warehousing challenges, real-time data warehousing solutions, and real-time data warehousing perceptions.

Data Warehouse Overview

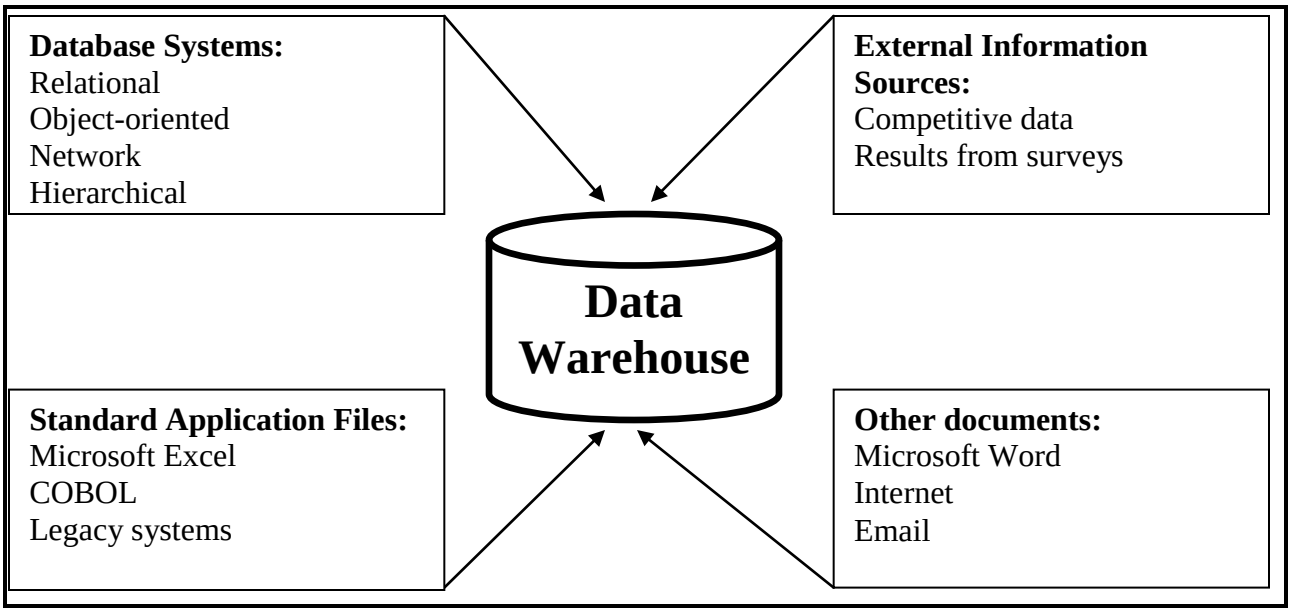
Bill Inmon defines data warehousing as a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision-making process, in his book, *Building the Data Warehouse*. In his definition, the data is:

- Subject-oriented as the warehouse is organized around the major subjects of the enterprise rather than the major application areas. This is reflected in the need to store decision-support data rather than application-oriented data.
- Integrated because of the coming together of source data from different enterprise-wide applications. The source data is often inconsistent using, for example, different formats.

The integrated data source must be made consistent to present a unified view of the data to the users.

- Time-variant because data in the warehouse is only accurate and valid at some point in time or over some time interval. The time-variance of the data warehouse is also shown in the extended time the data is help, the implicit or explicit association of time with all data, and the fact that the data represents a series of snapshots.
- Non-volatile as the data is not updated in real-time but is refreshed from operational systems on a regular basis. New data is always added as a supplement to the database, rather than a replacement. The database continually absorbs this new data, incrementally integrating it with the previous data.

According to Peter Robb and Carlos Coronel, authors of the textbook, *Database Systems: Design, Implementation, and Management*, the data warehouse is a read-only database optimized for data analysis and query processing.



Typically, data is extracted from multiple source systems throughout the organization as shown in Figure 1. These differing sources are loaded into the data warehouse after being transformed, integrated, and aggregated before being loaded into the data warehouse.

Wrappers, loaders, and mediators are programs that load data of the information sources into the data warehouse. Wrappers and loaders are responsible for loading, transforming, cleaning, and updating the data from the sources. Mediators integrate the data into the warehouse by resolving inconsistencies and conflicts between different sources. Furthermore, an extraction program can examine the source data to find reasons for conspicuous items, which may contain incorrect information. In the commercial section, these tools are classified as Extract-Transform-Load (ETL) tools and they try to automate and support the following tasks:

- Extraction: accessing different data sources
- Cleaning: finding and resolving inconsistencies in the source data
- Transformation: between various data sources and application languages
- Loading: load the data into the data warehouse
- Replication: replicating source database into the data warehouse
- Analyzing: detecting invalid/unexpected values
- High-speed data transfer: important for very large data warehouses
- Data quality: checking for correctness and completeness
- Analyzing metadata: in order to support the design of the data warehouse (Jarke, Lenzerini, Vassiliou, and Vassiliadas, 2003).

Once the information sources pass through the ETL tools, users access the data warehouse via front-end tools and/or end-user application software, normally business intelligence platforms, to mine the data in usable form.

> The concept of a data warehouse was deemed the solution to meet the requirements of a system capable of supporting decision-making by receiving data from multiple operational data sources and bringing it together in a defined relationship. The original concept of a data warehouse was devised by IBM as the 'information warehouse' and presented as a solution for accessing data held in non-related systems. The information warehouse was proposed to allow organizations to use their data archives to help them gain a business advantage. However, due to the sheer complexity and performance problems associated with the implementation of such solutions, the early attempts at creating an information warehouse were mostly rejected. Since then, the concept of data warehousing has been raised several times but it is only in recent years that the potential of data warehousing is now seen as a valuable and viable solution (Connolly and Begg, 2005).

The recent acceptance of data warehousing by organizations and the database community can be attributed to the benefits realized by successful implementations. These gains include the following: high returns on investment, competitive advantage, and increased productivity of corporate decision makers.

> A study by the International Data Corporation (IDC) in 1996 reported that average three-year returns on investment (ROI) in data warehousing reached 401%, with over 90% of the companies surveyed achieving over 40% ROI, half the companies achieving over 160% ROI, and a quarter with more than 600% ROI. The huge returns on investment for

those companies that have successfully implemented a data warehouse are evidence of the enormous competitive advantage that accompanies this technology. The competitive advantage is gained by allowing decision-makers access to data that can reveal previously unavailable, unknown, and untapped information on, for example, customers, trends, and demands. Data warehousing improves the productivity of corporate decision-makers by creating an integrated database of consistent, subject-oriented, historical data. It integrates data from multiple incompatible systems into a form that provides one consistent view of the organization. By transforming data into meaningful information, a data warehouse allows corporate decision-makers to perform substantive, accurate, and consistent analysis (Connolly and Begg, 2005).

The Demand for Real-Time Data Warehousing

In contrast to the traditional data warehouses, which provide a historical snapshot of data at a past point in time, real-time data warehouses allow decision makers to analyze current transactions and trends as they occur. "As soon as the business activity is complete and there is data about it, the completed activity data flows into the data warehouse and becomes available instantly" ("Active and Real-Time Data Warehousing", 2008). The demand for the migration from a traditional to real-time data warehousing system stems from the need for decision makers to incorporate current data from multiple data sources into the decision making. The new information gained from an event that just happened can change the direction of an action or cause a reaction depending on what the real-time information is.

Real time data warehousing is faster, allowing the business to make quicker informative decisions based on the real-time and historical data present in the real-time data warehouse. Businesses find this extremely attractive because the quicker they are able to

respond to certain trends in the market the more successful they will more than likely be ("Emanio Context: Self-servie Business Intelligence Platform", 2010).

The movement to real-time is the latest development in business intelligence (BI) and data warehousing. Real-time data warehousing provides the data that is required to implement real-time BI. By moving to real-time, firms can use BI to affect current decision-making and business processes. This capability is especially important for customer-facing applications, such as those found in call centers and check-in processes, and helps firms become more customer-centric. The purpose of real-time BI is to increase revenues, decrease costs, and improve supply chain management. Companies that successfully implement real-time BI can dramatically improve their profitability ("Real-Time Business Intelligence: Best Practices at Continental Airlines", 2006).

Real-Time Data Warehousing Challenges and Solutions Overview

Although real-time data warehousing gives organizations many benefits, the concept has initiated a new set of challenges. These challenges relate to the organizational and operational aspects of real-time data warehouses. "On the organizational side, there must be executive sponsorship and support, initial and on-going financial support, and BI and data warehousing personnel with the requisite skills" ("Real-Time Business Intelligence: Best Practices at Continental Airlines", 2006).

Operational real-time data warehousing challenges include the following: enabling real-time ETL, modeling real-time data, On Line Analytical Processing (OLAP) queries and changing data, scalability and query contention, and real-time alerting.

Real-Time ETL Challenges and Solutions

One of the most difficult parts of building any data warehouse is the process of extracting, transforming, cleansing, and loading the data from the source system. Performing ETL of data in real-time introduces additional challenges. Almost all ETL, whether accomplished with off-the-shelf products or custom-coded, operate in a batch mode. They assume that the data becomes available as an extract file on a certain schedule, usually nightly, weekly, or monthly. Then the system transforms and cleanses the data and loads it into the data warehouse. When loading data continuously in real-time, there cannot be system downtime. The heaviest periods in terms of data warehouse usage may very well coincide with the peak periods of incoming data. The requirements for continuous updates with no warehouse downtime are generally inconsistent with traditional ETL tools and systems ("Real-Time Data Warehousing: Challenges and Solutions", 2004).

Real-time data warehousing ETL solutions include the following: direct trickle feed, trickle and flip, and an external data cache.

Assuming that an application requires a true real-time data warehouse, the simplest approach is to continuously feed the data warehouse with new data from the source system. Once the data is near the warehouse, simply inserting the new data in real-time is not particularly challenging. The problem with this approach, which will probably not be readily apparent during development and initial deployment, is that it does not scale well. The same logic as to why data warehouses exist in the first place applies, that is, complex analytical queries do not mix well with continuous inserts and updates. Constantly

updating the same tables that are being queried by a reporting or OLAP tool can cause the data warehouse's query performance to degrade.

The trickle and flip approach helps avert the scalability issues associated with querying tables that are being simultaneously updated. Instead of loading the data in real-time into the actual warehouse tables, the data is continuously fed into staging tables that are in the exact same format as the target tables. Depending on the data modeling approach being used, either the staging tables contain a copy of just the data for the current day or for smaller fact tables can contain a complete copy of all the historical data.

This approach can be used with cycle times ranging from hourly to every minute. Generally, the best performance is obtained with 5-10 minute cycles, although 1-2 minute cycles (or even faster) are also possible for smaller data sets or with sufficient database hardware. It is important to test this approach under full load before it is brought into production to find the cycle time that works best for the application.

All of the solutions discussed so far involve the data warehouse's underlying database taking on additional load to process the incoming real-time data, and making it available to warehouse users. The best option in many cases is to store the real-time data in an external real-time data cache (RTDC) outside of the traditional data warehouse, completely avoiding any potential performance problems and leaving the existing warehouse largely as-is.

The benefits of the RTDC are increased performance, access to up-to-the-second data, and no scalability or performance risk on the existing warehouse. Also the cost of an RTDC solution is typically low compared to the cost to add sufficient hardware and

memory to the existing database server to overcome the scalability and performance issues associated with the trickle feed approaches. The negative of using a RTDC solution, with or without just-in-time data merging, is that it involves an additional database that needs to be installed and maintained. In addition, there is further work required to configure the applications that need to access the real-time data so that they point to the RTDC ("Real-Time Data Warehousing: Challenges and Solutions", 2004).

Modeling Real-Time Data Challenges and Solutions

The introduction of real-time data into an existing data warehouse, or the modeling of real-time data for a new data warehouse brings up some interesting data modeling issues. For instance, a warehouse that has all of its data aggregated at various levels based on a time dimension needs to consider the possibility that the aggregated information may be out of synch with the real-time data. The main issue regarding modeling however revolves around where the real-time data is stored, and how best to link it into the rest of the data model ("Real-Time Data Warehousing: Challenges and Solutions", 2004).

Real-time data warehousing solutions to model fact tables include partitions, views, and an external data cache.

The real-time partition usually should not be a literal table partition, in the database sense; rather, it is a separate table subject to special rules for update and query. The real-time partition should meet the following tough set of requirements:

- It must contain all the activity that has occurred since the last update of the static data warehouse.

- Link as seamlessly as possible to the grain and content of the static data warehouse fact tables.
- Be indexed so lightly that incoming data can be added continuously.
- Support highly responsive queries (“RealTime Partitions”, 2002).

From the query-tool configuration and administration perspective, the partition data modeling approach is the most complex to engineer. This approach requires either the query tool or the end user to understand where the various types of data are located and how to access them. The success of this approach depends significantly on how well the query tool insulates the end user from the extra complexities required to access real-time information.

Another real-time data modeling approach is to store the real-time data in different tables from historical data, but in the same table structure. By using database views, the historical and real-time data tables are combined together so that they look like one logical table from the query tool's perspective. This helps alleviate many of the problems associated with the separate partition approach, as the query tool or end users do not need to join two tables.

With this approach, the feed is into relatively small tables that can be easily modified or replaced when necessary by the load process. In addition, these tables will generally be small enough to sit in the database's memory cache, to alleviate some of the query contention issues involved with performing OLAP queries on changing data. Caching concerns apply and care needs to be taken to ensure that the query tool does not return old cache results to users who are requesting real-time data.

When using an external real-time data cache, no special data modeling is required in the data warehouse. The external data cache database is generally modeled identically to the data warehouse, but typically contains only the tables that are real-time. If the external data cache is accessed separately from the data warehouse, some additional tables may be required in the cache, such as lookup tables. The external data caching is most useful when the data is seamlessly integrated with historical data for query and analysis purposes. This approach has the advantage of eliminating performance problems associated with the other approaches to integrating real-time data into a data warehouse ("Real-Time Data Warehousing: Challenges and Solutions", 2004).

OLAP Queries and Changing Data Challenges and Solutions

OLAP and query tools were designed to operate on top of unchanging, static historical data. Since they assume that the underlying data is not changing, they do not take any precautions to ensure that the results they produce are not negatively influenced by data changes concurrent to query execution. In some cases, this can lead to inconsistent and confusing query ("Real-Time Data Warehousing: Challenges and Solutions", 2004).

Real-time data warehousing solutions to deal with OLAP queries and changing data include risk mitigation and an external data cache.

Risk mitigation can be accomplished by having a less-frequently updated snapshot of the real-time data in a separate partition that can be used for complex analytical queries. However, this adds a lot of setup, maintenance, and user education complexity. If an application is going to need to live with some internal report inconsistency, it is important to educate the users of the real-time information that this is a possibility. Uneducated

users who view data that does not properly add up are likely to assume that the system is malfunctioning and cannot be trusted.

The only way to solve this problem completely, without compromising report internal consistency, data latency, or the user experience, is to use an external real-time data cache. By keeping the real-time data separate from the historical data, the reports will never be internally inconsistent ("Real-Time Data Warehousing: Challenges and Solutions", 2004).

Scalability and Query Contention Challenges and Solutions

The issue of query contention and scalability is the most difficult issue facing organizations deploying real-time data warehouse solutions. Data warehouses were separated from transactional systems because high rates of transactional processing are contentious with the retrieval of data for analytical processing.

Usually the scalability of data warehouse and OLAP solutions is a direct function of the amount of data being queried and the number of users simultaneously running queries. Given a fixed amount of data, the number of users on the system is proportional to query response time. A large amount of concurrent usages causes reports to take longer to execute.

While this is still true in a real-time system, the additional burden of continuously loading and updating data further strains system resources. Unfortunately, the additional burden of a continuous data load is not just equivalent to one or two additional simultaneously querying users due to the contention between data inserts and typical OLAP select statements. While it depends on the database, the contention between

complex selects and continuous inserts tends to severely limit scalability ("Real-Time Data Warehousing: Challenges and Solutions", 2004).

Real-time data warehousing solutions to mitigate scalability and query contention include the following: simplify and limit reporting, apply more database horsepower, and an external data cache.

Many real-time warehousing applications are relatively simple. Users that want to see up-to-the-second data may have relatively simple reporting requirements. If reports based on real-time data can be limited to simple and quick single-pass queries, many relational database systems will be able to handle the contention that is introduced. Frequently the most complex queries in a data warehouse will be accessing data across a large amount of time. If these queries can be based only on the non-changing historical data, contention with the real-time load is eliminated.

Another important consideration is to examine who really needs to be able to access the real-time information. While real-time data may be interesting to a large group of users within an organization, the needs of many users may be adequately met with non-real-time data, or with near-real-time solutions. In addition, many users who may be interested in real-time data may be better served by an alert notification application that sends them an email or wireless message alerting them to real-time data conditions that meet their pre-defined thresholds. Designed properly, these types of systems can be scaled to 100 or 1000 times more users than could possibly run their own concurrent real-time warehouse queries.

It is always an option to add more hardware to deal with scalability problems. More nodes can be added to a high-end database system, or a stand-alone warehouse

server can be upgraded with faster processors and more memory. While this approach may overcome short-term scalability problems, it is likely to only represent a short-term resolution. Real-time query contention often has more to do with the fundamental design of the database management system than with the system resources available.

Using an external real-time data cache solves the scalability and query contention problem by routing all real-time data loading and query activity to an independent database that can be tuned for real-time access. With all the real-time activity on the separate cache database, the data warehouse does not bear any additional load. However, this approach, on its own, is not an entirely satisfactory solution. With the real-time data external to the warehouse, it is not possible for a single report or analysis to join or co-display real-time and historical information. Further, if complex analytical reports are run on the real-time cache, it is possible for the cache to begin to exhibit the same internal report inconsistency, database contention, and scalability problems that a warehouse would exhibit. This approach does work well for some stand-alone real-time reporting, analysis, and alerting applications, particularly those with few concurrent users and those with limited or no need for historical data.

An approach exists where the real-time information sits in an external data cache and the historical information sits in a warehouse, and the two are efficiently linked together as needed. This can be accomplished by just-in-time information merging (JIM). In a JIM system, queries that require real-time data are pre-processed by the a component known as the JIM Request Analyzer (JIM-RA). The JIM-RA analyzes the query to determine exactly what real-time data components are required, then the JIM Data Imager (JIM-DI) component takes a snapshot image of the required parts of the real-

time data cache. The real-time data from the snapshot is then loaded into temporary tables in the data warehouse by the JIM-DI component. Once the real-time data is present, the JIM-RA modifies the original query to include the temporary tables containing the snapshot data.

This approach allows existing business intelligence and OLAP tools to access real-time information seamlessly within the framework of existing data warehouse applications. There is no possibility of introducing scalability problems, as the required real-time data is only brought into the warehouse when needed, and on a temporary independent one-off basis for each query. Query contention is not a problem, as the data does not change while it is being queried. In addition, there is no risk of report inconsistency, as the real-time information is held constant in the snapshot between the multiple passes of SQL.

A variant of JIM is Reverse Just-in-time Data Merging (RJIM). Reverse JIM is useful for queries that are mainly based on real-time data, but that contain limited historical information as well. In Reverse JIM a similar process takes place, except the needed historical information is loaded from the data warehouse into the external data cache on a temporary basis, and then the query is run in the data cache. This only works when the data cache is located in a RDBMS system with full SQL support, and will not work with some IMDB systems that do not support many SQL functions.

An intelligent RJIM system needs to process as much of the query as possible on the warehouse before transferring only the data required at the highest level of summarization required for each query, otherwise an RJIM system could easily overflow an external data cache with large amounts of historical information. The best systems are

able to use both JIM and RJIM, and decide which process to use for each query based on the amount of data that is likely to go in either direction, and choose the path of least resistance and likely best performance ("Real-Time Data Warehousing: Challenges and Solutions", 2004).

Real-Time Data Quality and Alerting Challenges and Solutions

Another real-time data warehouse issue relates to data quality. Since the real-time data is commonly used for business monitoring, and subsequent creation of alerts, it is critical to ensure the quality of the data. For example, if an order was inadvertently placed for $1,000,000 rather than $1,000, then the order will trigger an alert. The incorrect order will take a short period of time to be investigated and corrected, which will cause real-time analytics to be out of sync.

Most alerting applications associated with data warehouses to date have been used to distribute email versions of reports after the nightly data warehouse load. The availability of real-time data in a data warehouse makes alerting applications much more appealing, as users can be alerted to real-time conditions as they occur in the warehouse, not just on a nightly basis.

The availability of real-time data makes products such as MicroStrategy's NarrowCaster and similar products from Cognos and Business Objects very valuable. These products operate on a schedule or event basis, so they can either trigger an alert every few minutes or hours, or need to be triggered by an external system. There is also the issue of threshold management. When alerts are triggered frequently (as opposed to once a night upon warehouse load), there needs to be a mechanism in place to make sure that once an alert is sent due to a condition in the warehouse that the alert is not

continuously sent over and over again during each alerting cycle ("Real-Time Data Warehousing: Challenges and Solutions", 2004).

Real-time data warehousing solutions to combat the alerting challenge include the following: n-minute cycle schedule, true data monitoring and triggering, and alert threshold management.

Currently all data warehouse alerting technology works on a schedule basis or on an event basis. This means to perform true real-time alerting, a process needs to continuously monitor the incoming data and trigger the events when appropriate. One way to approximate real-time alerting, without the added complexity of a real-time data stream monitoring solution, is to utilize a data warehouse alerting package on a scheduled basis, with the schedule typically set to every 1, 5, 15, or 30 minutes. This approach works reasonably well and provides near-real-time alerting. For a warehouse that is loaded on a near-real-time basis, all that needs to be done is to set the alerting schedule to trigger right after the data is refreshed.

For a warehouse that is being updated on a truly real-time basis, then using a n-minute cycle schedule will introduce a certain amount of latency, as an alert can't be triggered until the next cycle window comes around after the threshold condition is met. If the cycle time is low enough, and the alert is to be sent via email anyway, a 1 or 5-minute latency may be acceptable for many applications.

It is appealing to set the cycle threshold as low as possible, but this can introduce certain complications. Frequent alert cycles will introduce more loads on the data warehouse, which can impact performance for other users. In addition, if a cycle begins before the previous cycle has completed, it is possible for users to receive duplicate

alerts. Some tools will overwrite temporary worktables if another batch is triggered before the previous one completes, which can cause system errors or missing alerts. Until more sophisticated alert cycle pipelining technology is added to current alerting tools, it is important to choose a cycle window that will provide enough time for the previous batch of alerts to be completed before the next begins.

For true real-time data alerting, a triggering system needs to be in place, as existing data warehouse alerting systems are not capable of monitoring real-time data streams looking for exception conditions. General-purpose versions of such technologies are currently under development by companies such as Apama, but in the short-term custom solutions optimized for the task and data stream at hand are the best solutions.

These systems are complex, and depending on the number of users and alert conditions, and on the amount of incoming data, can require large amounts of hardware and particularly memory. Imagine a system providing real-time stock market alerts for 100,000 users. Portfolios for that many users and alert thresholds need to be stored in memory and then compared against every incoming tick from the various stock markets, which can be hundreds or thousands per second during market hours. This is a very difficult task, and there currently are not general-purpose technologies that can meet these needs.

Regardless of whether alerts are generated using near-real-time batch cycles or by a real-time triggering system, it is critically important that the users' alert thresholds are properly managed. Imagine a user who asks to be alerted by email when the inventory level of any item in his store drops to 5% of the normal level during the course of a day.

The user assumes that he or she will be alerted once when the level drops to 5%, but with a system running on a 5-minute cycle, a new alert will be send every 5 minutes.

The problem occurs when static threshold definitions, which are fine for systems that load on a nightly or weekly basis, are applied to systems that update more frequently. It is unlikely that the user desires to be reminded every 5 minutes. It might be better to alert the user once levels reach 5%, then maybe again at 2%, and one final time when stock runs out completely. On the other hand, a brokerage customer may want to know when a certain stock exceeds $20, and then be alerted again every increment of $2 it goes higher.

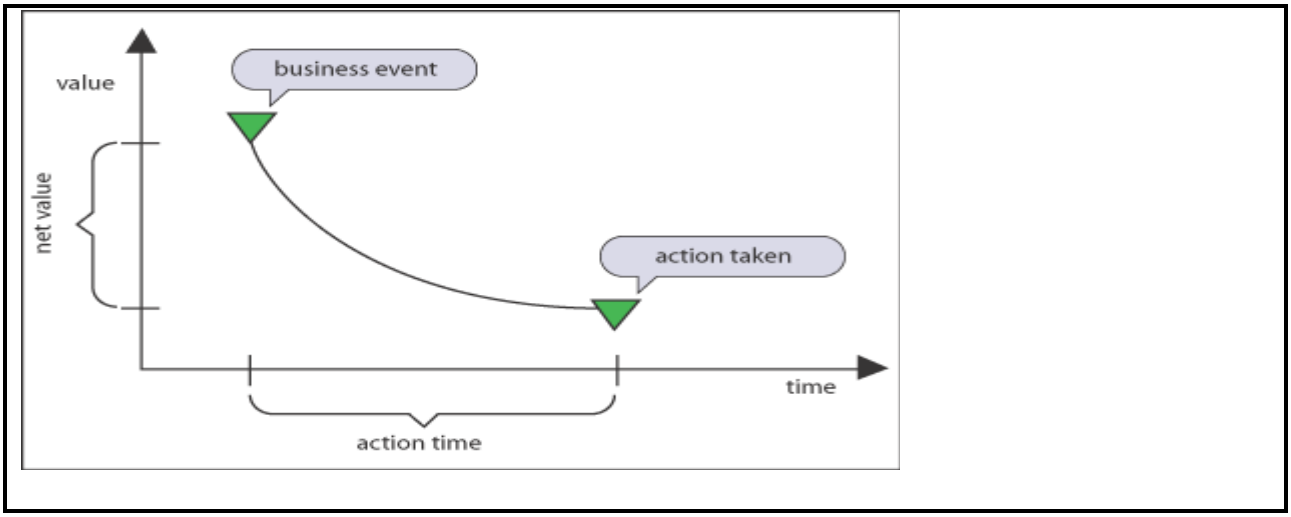
This type of threshold management is necessary for real-time alert systems to be accepted by users. Unfortunately, current warehouse alerting tools do not natively support it. Until this support is added, the best approach is to use the tools' post-service plug-in abilities to run custom SQL or procedures to directly update the user's thresholds based on the current data conditions. It is this post-execution job that make it critical that cycles don't overlap, as if the next batch begins before the thresholds are updated, it is likely that duplicate alerts will be sent (“Real-Time Data Warehousing: Challenges and Solutions”, 2004).

Real-Time Data Warehousing Perceptions

Controversy has struck the database community regarding the “real-time” terminology of the real-time data warehouse. Some researchers deem the expression misleading, that it is a scheme to sell a product, and that it detracts from the fundamental issues pertaining to the business need for quicker information.

Researchers state that “real-time” conveys a partial truth while obscuring the important issues. The usage of real-time as an adjective has unfortunately become marketing fluff. Some have suggested the term right time as a substitute; however, that begs the question about what is right. The key concept behind real-time is that the artificial representation must be coordinated with the real world so that we can respond to events in an effective manner. In today's technology, the data warehouse has become that artificial representation of real business world. In this regard, the primary purpose of the data warehouse is to maintain a unified and consistent view of business reality. Doing so in a timely manner is only one aspect of fulfilling this purpose.

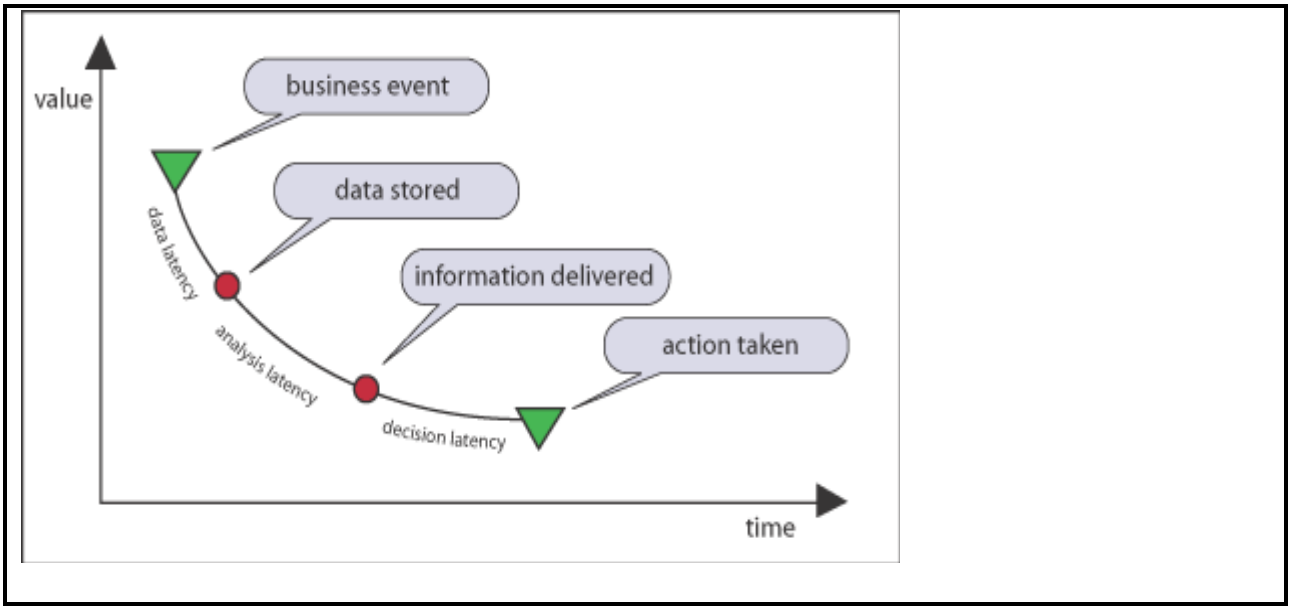
Figure 2. The Value-Time Curve



To clarify the relationship between time and business value, consider the value-time curve in Figure 2. At an initial time, a business event requiring a response occurs within some business process. Later, action is taken to respond to that event. For example, a customer requests information on a specific product, and then the company provides the information within a few seconds, minutes, hours, days, weeks, or months.

The assumption behind this decay curve is that the longer the delay or latency of the response, the less business value accrues to the company. In this example, the value is the probability that the customer will purchase the product. The action time is the duration between the event and the action, while the net value is the business value lost (or gained) over this duration.

Figure 3. Components of Action Time



Within the context of data warehousing, there are three components to this action time, as shown in Figure 3. The data latency is the time required to capture transform, cleanse, and store this data in the warehouse, where it is then ready for analysis. The analysis latency is the time required to analyze and disseminate the results to the appropriate persons. The decision latency is the time required for a person to understand the situation, decide on a course of action, and initiate it ("The BI Watch: Real-Time to Real-Value", 2004).

Chapter 3 – Methodology

The purpose of the methodology chapter is to provide a detailed discussion of the steps taken, the people involved, and where the study was conducted. This will be done in the following sections: research approach, research design, participants, data collection, and data analysis.

Research Approach Background

The objective of this thesis is to compare the research and practical perspective of the significant real-time data warehousing organizational and operational challenges. The study will start with a brief overview of data warehousing and a thorough review of real-time data warehousing including the history, current trends, future trends, and organizational and operational challenges as compiled by the various researchers in the database community. The information for the literature review will be explored throughout scholarly, peer reviewed articles, and books on Google Scholar and ACM. Older research will be utilized to provide a historical account of data warehousing. The practical perspective of the study will include an investigation into the most common research methodologies and a selection of the approach that supports the goals of the study.

Research Approach

For the purposes of gathering data from the practical perspective of this thesis, the qualitative research approach will be utilized.

> Qualitative researchers often start with general research questions rather than specific hypotheses, collect an extensive amount of verbal data from a small number of participants, organize the data into some for that gives them coherence, and use verbal descriptions to portray the situation they have studied (Leedy and Ormrod, 2005).

Furthermore, the qualitative methodology allows the focus of the study to be conducted in natural settings, involving an in-depth examination of the research problem in its entire complexity to construct a complete picture of the situation to be observed.

Research Design

Based on the objectives of the thesis, the case study qualitative research design will be used to gather information about the practical perspective of real-time data warehousing organizational and operational challenges. The purpose of a case study is to understand a person or situation in great depth by conducting interviews or observations in a natural setting. In addition, according to Paul Leedy and Jeanne Ormrod, authors of the textbook, *Practical Research: Planning and Designing*, the researcher in a case study records details about the context surrounding the case including information about the physical environment and any historical, economic, and social factors that have bearing on the situation. By identifying the context of the case, the researcher helps others who read the case study draw conclusions about the extent to which its findings might be utilized to other situations.

Data collection

Data will be primarily collected for the case study research design by conducting an informational interview with a real-time data warehouse professional employed by one of the world's largets online retail organizations. The questions will be asked in a teleconference to an identified employee currently working at the company in a mid-level management role. This is the most effective way to conduct the research because of the company's distant location. In addition, it allows the subject an adequate amount of time to answer each question fully and the capabilities to obtain answers from others within the organization, if necessary. Follow up sessions will be used if necessary to gather additional data that may not have been collected

during the initial interview. These sessions will either be via teleconference or email correspondence. The interview questions will explore the following topics (refer to the Appendix Interview Questions for a complete listing of the interview questions):

- Business and Support: understand what products and services are offered, what inspired them to do real time updates, what the business case for real time is, how long data warehousing and real-time data warehousing has been utilized, what type of data is stored, and how the organization is structured.
- Challenges and solutions: discuss the issues and investigate solutions to these issues regarding real-time ETL, modeling real-time data, OLAP queries and changing data, scalability and query contention, and real-time data quality and alerting.

In addition, articles, books, and company websites will be used to collect data about various topics that will come up during the interview process that have not been discussed previously. For example, if the contact describes a specific vendor's software packages, then supplementary data about the vendor and the software might be gathered and discussed in order to add value to the analysis.

Due to confidentiality requirements, the organization and participant in the case study will not be disclosed; rather, they will be described at a high level. The organization is a top five online retail company based in the western United States and will be referred to as Company ABC. They have doing business since 1999 and started off as a very small organization with only a few employees. Company ABC utilized real-time databases from inception to provide users with real-time reporting capabilities, but decided to implement a enterprise-level real-time data warehousing, reporting, and analytics system after years of exponential growth. The contact that was used for the interview process was Company ABC's director of data warehousing.

Data analysis

The findings from the previous research in the literature review section will be compared to the information gathered from Company ABC in the case study. The case study information will be used to either validate or exploit gaps in the previous research. The results of both findings will be discussed in the discussion chapter and conclusions chapter. Sections in the conclusions chapter will present the research limitations, conclusions, and ideas for future research that may affect the successful implementation of a real-time data warehousing system.

Chapter 4 - Results

The purpose of the results chapter is to provide a detailed discussion of the case study with Company ABC. This will be done in the following sections: business and support and challenges and solutions.

Business and Support Overview

Company ABC is a leading outlet shopping site and a top 5 e-tail business offering discount brand name merchandise including sporting goods, bed and bath items, electronics, jewelry, travel, and luxury cruises. It recently began offering its own auction website that allows individuals to bid against each other on various merchandise items. Company ABC has been conducting business in the retail industry for the last 10 years. It started out as an extremely small company in relation to sales, website hits, and employees. All systems were developed in-house with the expectation of allowing users the ability to obtain real-time data. This worked well while the company size remained small, but after the company began to grow exponentially in 2002, at upwards of 100% per year, their existing systems proved inadequate. This growth, which caused stress to their databases and reporting capabilities, coupled with the fact that Company ABC's management wanted to adopt a centralized and standardized, enterprise-level data warehouse system and reporting and analytics system, caused them to decide to implement a real-time data warehousing system in 2005. They chose Teradata for the relational database management system. Teradata has a data warehouse migration product specifically for converting Oracle data warehousing systems to Teradata.

Company ABC also eventually selected to use GoldenGate for real-time data integration. Oracle GoldenGate delivers low-impact, real-time data acquisition, distribution, and delivery across heterogeneous systems. Using this technology, it enables cost-effective

and low-impact real-time data integration and continuous availability solutions. Oracle GoldenGate 11g offers tighter integration with Oracle technologies and applications, support for additional heterogeneous systems, and improved performance ("Oracle GoldenGate 11g", 2010).

Figure 4. CEO Organizational Chart

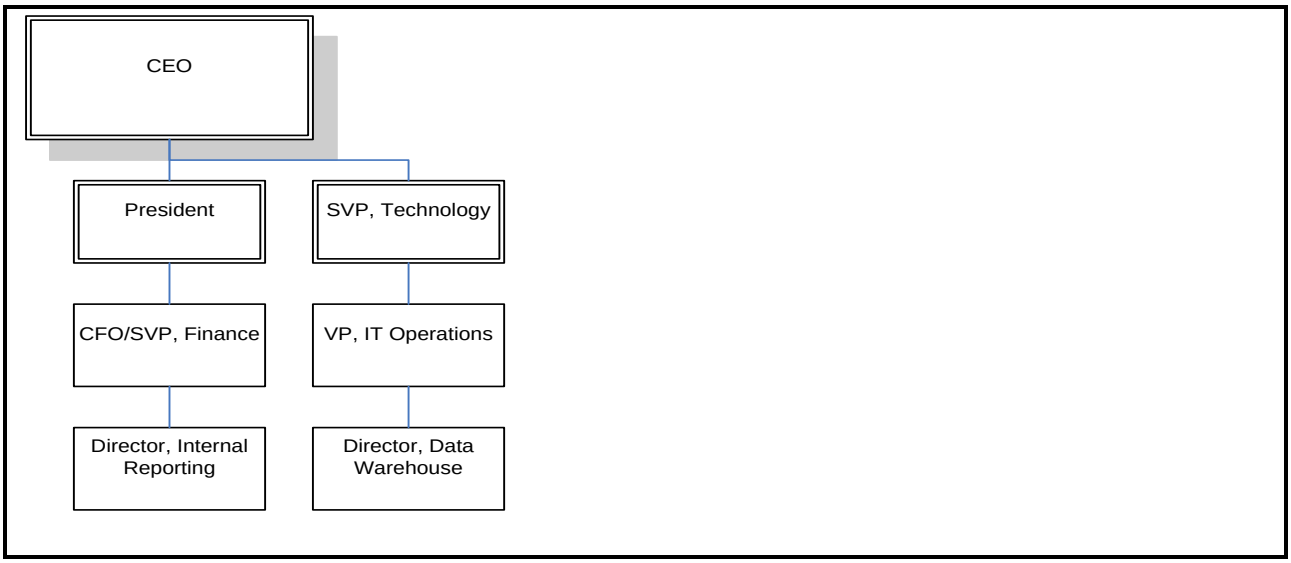


Figure 5. Internal Reporting Director Organizational Chart

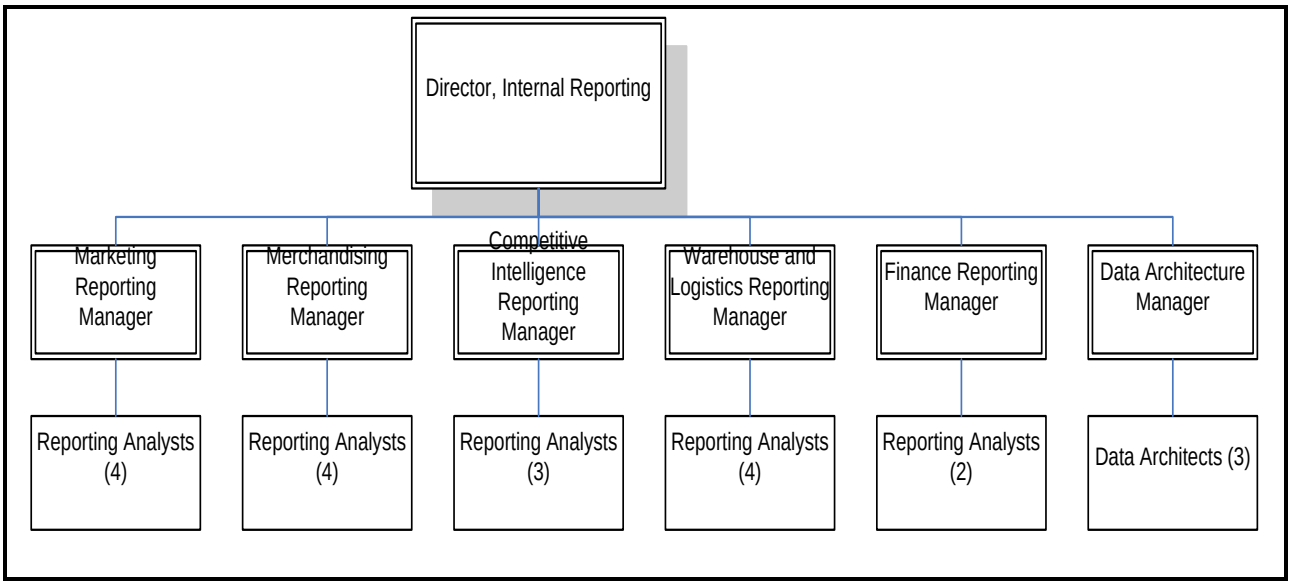
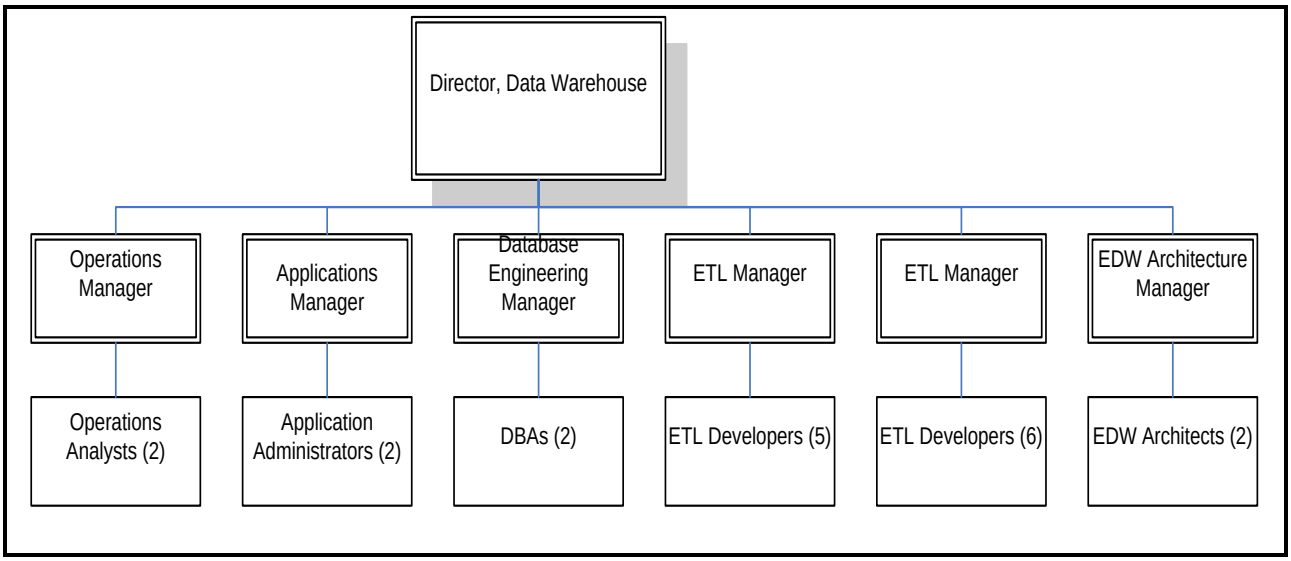


Figure 6. Data Warehouse Organizational Chart



Figures 4, 5, and 6 present the distinct organizational charts. As depicted in Figures 5 and 6, the team responsible for real-time data warehousing at Company ABC is split up into two areas: 1) Development and Support and 2) Reporting and Business Analysis. The Development and Support department is led by the Director of Data Warehousing and the Reporting and Business Analysis department is led by the Director of Reporting and Business Analysis. Company ABC has a support team to handle various aspects of their real-time enterprise data warehouse system. This team handles a wide variety of items including the following: incident tracking, troubleshooting, and resolving, service level agreements and service level guarantees, and performance tuning measurement.

Incident support at Company ABC is handled in-house during regular business hours and by offshore support during non-peak hours. All issues are tracked using an internal tracking system. Incident support is centered around the enterprise scheduler, GoldenGate data integration, ETL, and configuration. During peak hours, the data warehouse team conducts first level support. This team is responsible for troubleshooting and fixing or gathering all

appropriate information. If the data warehouse team is unable to fix the problem, then they distribute all the information they collected during their initial troubleshooting process to the second level support team, which is comprised of application developers.

Company ABC has approximately three hundred and eighty total business users of their real-time enterprise data warehouse. Three hundred of the regular business users have access to the canned reporting capabilities and the various views against the data warehouse tables. Eighty individuals have been deemed power users and are able to write and run SQL statements against the real-time data warehouse tables in addition to the privileges of the regular users. As previously stated, all users are encouraged to leverage the real-time reporting system that has been built to run again real-time views instead of ad-hoc querying and reporting mechanisms that are run against the actual real-time tables because using the tables introduces a high potential for decrease performance of the real-time system.

Company ABC must measure real-time data warehouse performance to be able to monitor the overall health of the system. In order to do this for the day-to-day performance, the database administrators setup separate IDs for all jobs in order to track the top jobs that cause system stress. This approach helps make it easier for the support team to dig into these issues and resolve them. For system capacity, forecast, and planning, Company ABC looks regularly at CPU usage by each workload to see how they are trending. If an uptick is encountered, they try to investigate, which includes finding out if something changed or if it was an operation issue. This approach allows support team members to catch problems early on and fix them.

Challenges and Solutions

Company ABC encountered numerous challenges throughout the various stages of the real-time data warehouse life cycle including the following: real-time ETL, modeling real-time

data, OLAP queries and changing data, scalability and query contention, and real-time data quality and alerting. The challenges will be discussed, followed by a summary of the solutions put into action by Company ABC.

Real-Time ETL Challenges and Solutions

Real-time data integration vendor selection was an obstacle because of the iterative nature of the process. Company ABC considered three vendor options initially, but quickly eliminated one because of negative experiences reported by employees that had previously used it. After a short deliberation, which Company ABC deemed necessary because of the need for rapid implementation of their real-time data warehouse, they decided to begin the process of implementing the second vendor option. Soon after the implementation process of the second vendor option, it was abandoned because Company ABC found out that the product they purchased did not have real-time data integration capabilities; rather, it was a facility for real-time data replication.

After abandoning the second vendor option, the third vendor option, Oracle GoldenGate was selected and the implementation process began quickly. Oracle GoldenGate was able to deliver the right real-time solutions for the extremely tight period requested by Company ABC. In one week, the Oracle GoldenGate data integration product was able to have Company ABC's four main data source systems incorporated into the Teradata enterprise data warehouse.

Although the rapid success of the Oracle Golden Gate data integration solution made the decision for Company ABC to stay with the product simple, it did come with a few challenges that needed to be worked through. Early on in the implementation, the Oracle GoldenGate process would become overloaded, which caused transactions to not be processed in real-time. The system would try to load all the backed up data all at once to catch up, but sometimes this

would be up to three hours worth of transactions. This catch up process saturated the system and caused extreme bottlenecks on the system.

To reduce the problem, Company ABC had to build tuning around the Oracle GoldenGate real-time data integration process. This tuning continuously monitors the real-time data integration flows. If a problem occurs, then the real-time data warehouse support team is alerted of the issue. In addition, if the issues cause an obstruction of transaction processing in the system, the tuning procedure Company ABC built will move transactions that were unable to process gradually once the system is running properly instead of trying to move all of the transactions at one time. If the real-time data integration is unable to load any transactions during normal processing throughout the day, then Company ABC has a nightly batch load that moves all of this data into the real-time enterprise data warehouse. This batch load is done at 6:00 AM, which is before normal business hours. In addition, Company ABC has separate servers that are dedicated to data integration activities, so it can be moved from one server to another at anytime.

Company ABC had to overcome an obstacle with their real-time ETL that they discovered before going live. They realized that with the way their system worked different sections of the customer invoices would not always be updated at the same time. Given any set of tables with referential integrity, such as invoice headers and lines, purchase order headers and lines, and customer households and accounts, there had to be a mechanism in place to prevent the processing of incomplete records. For example, if the real-time ETL processing falls behind on the invoice header table, they needed to prevent invoice lines without a matching header to be written to the real-time enterprise data warehouse. Company ABC eliminated this issue by

thoroughly researching how all the tables relate to each other and building a tuning process around the checks and constraints.

Modeling Real-Time Data Challenges and Solutions

Company ABC had to overcome performance and tuning issues related to modeling real-time data. For instance, they created partitions on some of the larger tables within their real-time data warehouse system with the intent to improve query performance. An unforeseen consequence of this tuning activity was that rows updated in real-time often crossing multiple partitions, which caused the real-time updating process to encounter performance degradation. This inadvertent performance hit was unacceptable to the data warehouse team and various users throughout the organization.

The performance hits on the loading of the real-time enterprise data warehouse for partitions too significant to ignore, so Company ABC had to conduct a more thorough investigation on what partitioning strategy to utilize. With this and many other specific tuning obstacles, it took Company ABC multiple passes at the problems to get them corrected.

OLAP Queries and Changing Data Challenges and Solutions

Company ABC had to overcome the challenge of changing data in their real-time enterprise data warehouse and the effect on OLAP queries. To contend with this issue before implementation, they adopted the concepts of mini-batch updates and denormalized and pre-aggregated data structures. Company ABC decided to append real-time records in a mini-batch mode once the source systems are updated. In addition, they built denormalized and pre-aggregated data structures for the mini-batch loads in order to allow for no data latency.

Once the real-time enterprise data warehouse was implemented a new problem with the influences of changing data on OLAP queries shifted to physical and conceptual data

warehousing tuning issues. Company ABC encountered a problem with their indexing approach because the indexes they were using caused delays in the real-time updates. They used join indexes on the Teradata database management system, which they found to be similar to materialized views in the Oracle database management system. When a data set is defined from one or more base tables, and as the base tables are updated, the join index updates automatically. The Teradata optimizer determines at query run time whether to utilize the join index or the base table to satisfy a query. Company ABC encountered a lot of difficulty with the Teradata join indexes, which was centered on two fundamental problems. Once the base tables had been updated, it took too long for the join indexes to update and while the join indexes were updating, they were locked so any query that used a join index as part of its plan had to wait for the join index to finish updating.

Scalability and Query Contention Data Challenges and Solutions

Company ABC has to deal with query contention and scalability issues for their real-time enterprise data warehouse. To handle these challenges, they decided to adopt a business philosophy that end users can only query views that are created from the data warehouse table, instead of querying the tables directly. This approach prevents end users from being able to lock the tables while they are being updating in real-time. In addition, end users are encouraged to leverage the canned reporting system that are built instead of ad-hoc querying and reporting mechanisms that may take much longer to build and run.

Real-Time Data Quality and Alerting Challenges and Solutions

Company ABC has data quality and alerting issues that exist with their real-time data warehousing system. To work through these challenges, they have a data quality team comprised of people in several functional areas across the business. This team works diligently

to get data quality issues corrected, but the process to do so can be extremely difficult and time consuming. Many of the data quality issues are a challenge to find and once they are found, the solution typically involves changes in the operational system, ETL process, real-time data warehouse management system, reporting system, and business rules.

In addition to the data quality functional team, alerts are set up to monitor each step of the process to move data in real-time from the source systems to the Teradata data warehouse. Alerts are distributed to users through an enterprise scheduler, which is an automated monitoring system. The enterprise schedule sends email notices when issues are encountered in the real-time data integration and batch processes that are run each morning.

Chapter 5 - Discussion

The purpose of the discussion chapter is to compare the findings of the case study to the previous research. The case study will either validate or provide additional information that was not contemplated in the previous research. This comparison will be done in the following sections: organizational support, real-time ETL, OLAP queries and changing data, scalability and query contention, and real-time data quality and altering.

Organizational Support

Organizational support is a critical element to the success, acceptance, and support of a real-time data warehousing system. This support includes the following: executive sponsorship and financial support and help desk support.

The case study of Company ABC confirms the previous research of the importance of executive sponsorship and financial support and user acceptance. The previous research states, "on the organizational side, there must be executive sponsorship and support, initial and on-going financial support, and acceptance of use of real-time data by organizational personnel" ("Real-Time Business Intelligence: Best Practices at Continental Airlines", 2006). The director of data warehousing described the senior-level management team as having buy-in and financial support from the onset due to their desired quick implementation, the necessity to maintain real-time data, and the importance of converting to an enterprise-level data and reporting system. Both types of research provided detailed accounts concerning how user acceptance was a slowly evolving process because the users were used to their existing data and reporting systems.

The case study of Company ABC provided a wealth of detailed information regarding help desk support as compared to the previous research. Company ABC also provides training materials, such as diagrams that illustrates the location of various data and reporting items they

might utilize. The director of data warehousing also provided a detailed description of their real-time data warehouse support team, which requires much different support than a traditional data warehousing system from the standpoint of troubleshooting, resolving, service level agreements and service level guarantees, and performance tuning measurement.

Real-Time ETL

The case study identified a significant research gap when compared to the previous research for real-time data integration. The vendor selection for Company ABC was a major obstacle in the real-time data warehouse implementation. They spent lot of time, effort, and money on two separate vendor product adoptions. The first product chosen ultimately failed to meet their needs and the second product, Oracle GoldenGate, satisfied their demands for quick incorporation. Previous research aimed at real-time data warehousing challenges and solutions did not provide information regarding real-time ETL vendor selection and practical world examples; rather, it discussed specific types of ETL methodologies and tuning approaches that could be used for real-time data integration.

Modeling Real-Time Data

The case study supported the previous research regarding physical and logical database tuning required to model real-time data. Company ABC had challenges ensuring that indexes and partitions did not lock real-time updates and adversely affect performance. They took several passes at tuning these database objects to improve the system's performance. The previous research discussed the use of indexes and partitions, and views for physical and logical database tuning. The author of the article, *RealTime Partitioning,* affirmed that indexes on the real-time data warehouse tables need to be light to not adversely affect the performance on tables that need to be updated continuously. According to the author of the article, *Real-Time Data*

Warehousing: Challenges and Solutions, the partition data modeling approach is the most complex to engineer and the success of this approach significantly depends on how well the query tool insulates the end user from the extra complexities required to access real-time information.

The case study of Company ABC exploited some deficiencies in research concerning modeling fact tables. The director of data warehousing discussed Company ABC's strategy for developing their real-time fact tables. The previous research did not discuss the development strategy of fact tables; rather, it discussed tuning options on the overall database structure, including fact tables. According to the author of the article, *Real-Time Data Warehousing: Challenges and Solutions,* the main issue regarding modeling however revolves around where the real-time data is stored, and how best to link it into the rest of the data model. Real-time data warehousing solutions to model fact tables include the following: partitions, views, and an external data cache. The success of the partition approach significantly depends on how well the query tool insulates the end user from the extra complexities required to access real-time information. With database views, the historical and real-time data tables are combined together so that they look like one logical table from the query tool's perspective. An external data cache approach focuses on feeding the real-time data into relatively small tables that can be easily modified or replaced when necessary by the load process.

OLAP Queries and Changing Data

The case study provided a gap in research for OLAP queries and changing data in real-time data warehousing. Company ABC adopted the following two concepts to contend with the challenge that changing data poses to OLAP queries: mini-batch updates and denormalized and pre-aggregated data structures. They implemented a strategy to append real-time records in a

mini-batch mode once the source systems are updated. In addition, they built denormalized and pre-aggregated data structures for the mini-batch loads in order to allow for no data latency. The existing research discussed leaving the historical data in a separate table or partition than the real-time data so that OLAP queries using historical data only will not be hindered by changing data and using views for querying.

> By using database views, the historical and real-time data tables are combined together so that they look like one logical table from the query tool's perspective. This helps alleviate many of the problems associated with the separate partition approach, as the query tool or end users do not need to join two tables ("Real-Time Data Warehousing: Challenges and Solutions", 2004).

Query Contention and Scalability

The case study confirmed the previous research in relation to query contention and scalability. Company ABC implemented a strategy to utilize views to enable users to run queries and reports against data updated in real-time. This strategy prevents business users from locking tables that have real-time data feeds. In addition, Company ABC does not allow most users the ability to write queries against the tables and all users are encouraged to use the canned reporting systems.

The previous research, from the author of the article, *Real-Time Data Warehousing: Challenges and Solutions*, provides the concept of using views to increase the system performance and prevent table locks. The author also discusses that companies should examine who really needs to be able to access the real-time information because although, the real-time data may be interesting to a large group of users within an organization, the needs of many users may be adequately met with non-real-time data, or with near-real-time solutions.

Real-Time Data Quality and Alerting

The case study validated the previous research for alerting. Both stressed the importance of continuously monitoring data quality by using trigger events. Company ABC uses their in-house developed enterprise scheduler for real-time alerts, which coincides with the previous research. "For true real-time data alerting, a triggering system needs to be in place, as existing data warehouse alerting systems are not capable of monitoring real-time data streams looking for exception conditions" ("Real-Time Data Warehousing: Challenges and Solutions", 2004).

The case study presented additional information as compared to the previous research regarding data quality. Company ABC put together a data quality team, comprised of individuals from the various business units, to work through data quality issues stemming from the legacy systems. This process can be difficult and extremely time consuming.

Chapter 6 - Conclusions

The purpose of the conclusions chapter is to evaluate the findings of the case study and previous research from the viewpoint of how they affect the successful implementation of a real-time data warehouse. This will be accomplished in the following sections: research limitations, conclusions, and future research.

Research Limitations

The case study methodology utilized to provide research of Company ABC's real-time enterprise data warehousing system introduced important limitations that must be discussed. The fact that only one organization was selected to conduct the study makes it difficult to generalize the discoveries to other experiences and situations of a wide variety of companies.

An additional limitation of the case study is that all the responses to the questions being asked were from the viewpoint of only one individual, the director of data warehousing at Company ABC. Many individuals may go through the exact same experiences, but may have an entirely different outlook than someone else. For example, when discussing the necessity for Company ABC to purchase all new hardware and software for their real-time enterprise data warehouse, the subject described it as not being a major obstacle at all because it just needed to be done. This person's perspective may be a completely different perspective than the individuals that had to install, configure, support, and fix problems related to the various pieces of new hardware and software. These individuals might say that the new hardware and software was an extremely burdensome task.

Another inadequacy of the case study is that the implementation of the real-time enterprise data warehousing system at Company ABC was carried out five years ago. The subject's involvement in the process of implementation could have limited or non-existent, thus,

making their responses to the questions a collection of second-hand information. This can introduce limitations to the study because it is possible that the responses and descriptions provided by the director of data warehousing will be less accurate than if the time gap was much smaller and the original employees that were present during the implementation provided information.

A shortfall of the research in general is that many of the articles and company profiles are written from the viewpoints of various vendors that have helped companies implement a section of their real-time data warehousing system. These articles are trying to help prospective customers buy their products and in multiple cases, the articles exaggerated the scope of the real-time systems. This created a challenge for the previous research section because there is only a small amount of research in the area and many of the sources for real-time data warehousing were unusable. In addition, when the articles written by vendors were used to contact companies to participate in the case study, many contacts within these companies stated that the vendors may have overstated their use of real-time data warehousing and they could be of no help to the case study.

Conclusions

As stated in the introduction section, the purpose of this research is to address and evaluate how existing research conducted by experts in the database field compares to a practical example of a fully deployed real-time data warehouse by an organization in retail industry. This was done by exploring the organization and operational challenges and solutions related to real-time data warehousing and how they influence a successful implementation.

The way a company handles the organizational challenges related to implementing a real-time data warehouse is the foundation as to whether it is a successful implementation or not. The

organizational challenges begin with executive sponsorship and financial support and without either an implementation of a real-time data warehouse would not be possible. Although both the case study and previous research illustrated the importance of these items, these organizational issues are not specific to a real-time data warehousing system; rather, they are fundamental to the success of any IT project.

Another organizational support topic that is important to the successful implementation of a real-time data warehouse is help desk support. A real-time data warehouse support team designed to handle incident tracking, troubleshooting, problem resolving, service level agreements, service level guarantees, and performance tuning measurement is vital part of a successful implementation. This team must be experts on all aspects of the system in order to provide high availability, zero data latency, and recommendations for system performance improvement.

The manner in which a company handles the various operational issues surrounding a real-time data warehouse has a drastic effect on the success level. The operational issues include the following: real-time ETL, modeling real-time data, OLAP queries and changing data, scalability and query contention and scalability, and alerting and data quality.

Real-time data integration proved to be an instrumental area for the success of a real-time data warehouse system. The issues that Company ABC encountered in this realm could have been detrimental to the implementation of the real-time system because one product they purchased ended up not being able to fulfill their need for real-time ETL after they adopted it. They had to abandon it, and then start fresh with a new product. This experience could have caused the entire project to fail for Company ABC and must be considered for any organization choosing among the various products being offered by third-party vendors. The case study

brought the forefront a gap in research for real-time data integration products that will be discussed in more depth as part of the future research section. The existing research provided different strategies for developing real-time data integration solutions in-house and the advantages and disadvantages of each. This research would be beneficial to a company choosing to develop their own ETL processes.

The matter of modeling real-time data from the standpoint of physical and logical tuning can have a drastic effect on the success of a real-time data warehouse adoption. Traditional performance enhancing database objects such as indexes and partitions sometimes decrease performance if not utilized properly. The case study and existing research suggests that the use of these objects can help performance, but that must not adversely affect the real-time updates or the user's ability to query the data or use canned reports. Views can be used to feed queries and reports in order to prevent users from locking the tables updated in real-time.

The theme of modeling fact tables is directly related to the achievement of a real-time data warehousing system. The case study and existing research took differing, but viable, approaches to the discussion of real-time fact tables. The participant from Company ABC discussed the strategy of developing real-time fact tables. The previous research did not discuss the development strategy of fact tables, but it discussed the storage of fact tables and tuning options. It is crucial for organizations to understand how their fact tables should be modeled in relation to utilizing vendor products, developing fact tables in-house, storage, and tuning options.

The affect of changing data on OLAP queries in real-time data warehousing must be reflected regarding successful adaptation. Company ABC initiated the concepts of mini-batch updates and denormalized and pre-aggregated data structures. They implemented a strategy to append real-time records in a mini-batch mode once the source systems are updated. In addition,

they built denormalized and pre-aggregated data structures for the mini-batch loads in order to allow for no data latency. The existing research discussed leaving the historical data in a separate table or partition than the real-time data so that OLAP queries using historical data only will not be hindered by changing data.

Scalability and query contention is a fundamental issue that influences the successful implementation of a real-time data warehouse. The ability of users to lock real-time tables must be minimized or eliminated to the extent possible. Companies can investigate what business unit users need the real-time ability and limit accessibility. When doing this they must be careful to not adversely affect user acceptance, as some users may become discouraged if they do not have access to the system. As discussed earlier in the physical and logical data warehouse tuning section, views can be used to prevent users from being able to lock tables that are updated in real-time and still allow them to access the data.

The topic of alerting and data quality is considered imperative to the successful utilization of a real-time data warehouse. The research emphasized the significance of having a constant monitoring system that alerts users and the data warehouse support team of data quality issues. In addition, companies adopting a real-time data warehouse should strongly consider a data quality team of individuals from the various business units and the real-time data warehouse support team to identify, research, resolve, and document real-time data issues with all of the data sources that feed the warehouse. This team should work both proactively and reactively to resolve data issues.

Future Research

The process of comparing the previous research to the case study introduced many areas for additional exploration that would be valuable to the successful implementation of a real-time

data warehousing system including real-time ETL and skills needed for support team. The research, in either the form of the case study or existing research, provided viable information for many of these topics, but a few were not adequately covered, leading to areas for future research considerations.

Real-time ETL was a topic thoroughly discussed by both the case study and previous research. Company ABC's experience and issues with their vendor selection process uncovered an area for further research. The time and money spent implementing two separate products from third party vendors provides the evidence that research comparing different real-time integration vendor products would assist the database community.

The subject pertaining to the new skills needed for real-time data warehousing support personnel was not covered in either the case study or previous research. The development or acquisition of the skills necessary to support a real-time data warehouse system is extremely significant to successful deployment. The fact that this support is drastically different than that of a traditional data warehousing system from the standpoint of database administration, design, and development, suggests that additional research detailing this would benefit the database industry.

References

Active and Real-Time Data Warehousing [Electronic (2008). Version] Retrieved May 5, 2010 from http://domino.research.ibm.com/comm/research_people.nsf/pages/ubnambiar.pubs.html/$FILE/MNSV-eds09.pdf

Connolly, Thomas, and Begg, Carolyn. 2002. Database Systems: A Practical Approach to Design, Implementation, and Management. 4th ed. Essex, England. Pearson Education Limited.

Emanio Context: Self-service Business Intelligence Platform [Electronic 2010. Version] Retrieved December 8, 2010 from http://www.emanio.com/Real-time-data-warehousing.html

Hackathorn, Richard. (2004). The BI Watch: Real-Time to Real-Value. The Thomson Corporation and DM Review.

Inmon, William. 2002. Building the Data Warehouse. 3rd ed. New York, New York. John Wiley & Sons.

Jarke, Matthias, and, Lenzerini, Maurizio, and, Vassiliou, Yannis, and, Vassiliadis, Panos. Fundamentals of Data Warehouses. 2nd ed. New York, New York. Springer-Verlag Berlin Heidelberg New York.

Langseth J. (2004). Real-Time Data Warehousing: Challenges and Solutions. DSSResources.COM.

Leedy, Paul, and, Ormrod, Jeanne. 2005. Practical Research: Planning and Design. 8th ed. Upper Saddle River, New Jersey. Pearson Education, Inc.

Oracle GoldenGate 11g [Electronic (2010). Version] Retrieved October 18, 2010 from

http://www.oracle.com/us/products/middleware/data-integration/goldengate/index.html

Real-Time Data Integration for Data Warehousing and Operational Business Intelligence [Electronic (2010). Version] Retrieved December 11, 2010 from http://www.oracle.com/us/products/middleware/data-integration/goldengate11g-realtimedw-wp-168215.pdf

RealTime Partitions [Electronic (2002). Version] Retrieved July 3, 2010 from http://intelligent-enterprise.informationweek.com/020201/503warehouse1_1.jhtml

Rizzi, S., Abello, A., and Lechtenborger, J., and Trujillo, J. (2006). Research in Data Warehouse Modeling and Design: Dead or Alive?. Proceedings of the 9th ACM International Workshop on Data Warehousing and OLAP, 3-10.

Rob, Peter, and Coronel, Carlos. 2007. Database Systems: Design, Implementation, and Management. 7th ed. Boston, MA. Thomson Course Technology.

Watson, H.J., and Wixom, B.H., and Hoffer, J.A., and Anderson-Lehman, R., and Reynolds, A.M. (2006). Real-Time Business Intelligence: Best Practices at Continental Airlines. Information Systems Management, 7-18

Appendix A

Case Study Questions

1. What type of business does your organization conduct?
2. How is your organization structured, including organizational titles and responsibilities?
3. How long have you been using data warehouses?
4. How long have you been using real-time data warehousing?
5. Why did you choose to go to a real-time data warehouse?
6. What do you use a real-time data warehouse for?
7. What types of operational data are stored in your real-time data warehouse?
8. How much of your data warehouse is updated in real-time?
9. Do you have queries being performed against both the data structures that were updated in real-time and those updated in batch?
10. What have been your major real-time data warehouse structure obstacles?
11. What organizational obstacles did you encounter when adopting real-time data warehousing, such as executive buy-in, financial commitment, developing personnel expertise, and end user acceptance?
12. What technological issues did you encounter related to new hardware and software and establishing processes and procedures for supporting and managing real-time data feeds from source systems?

13. How has the real-time data warehouse affected your network?

14. What ETL problems have you faced?

15. How did you overcome the organizational obstacles?

16. How are your real-time fact tables modeled?

17. What is your strategy for OLAP queries and changing data in the real-time data warehouse?

18. How do you handle query contention and scalability?

19. How are hardware and software problems resolved?

20. How do you handle data quality challenges and alerting capabilities?

21. Who is responsible for ETL?

22. How is ETL carried out?

23. How have you resolved ETL problems?

24. If the support team is different than your team, then how is your real-time data warehouse support team structured, including organization titles?

25. What are the collective and individual responsibilities of this team?

26. How are real-time data warehouse incidents tracked?

27. What are you service level agreements related to data availability and data quality?

28. How many users utilize your real-time data warehouse?

29. How have business users been trained to use the real-time data warehouse?

30. How have you bridged the communication between the users and developers and designers?

31. Do you have a team responsible for managing real-time data warehousing problems? If so, who solves the problems?

32. How is performance of your real-time data warehouse measured?