OLAP and OLTP Data Integration for Operational Level Decision Making

By

Muhammad Azhar Chohan

Submitted to the Department of Computer Engineering
in partial fulfillment of the requirements for the Degree of
Masters of Science in Software Engineering

Thesis Supervisor

Brig Dr. Muhammad Younus Javed

College of Electrical and Mechanical Engineering

National University of Sciences & Technology

2011

OLAP and OLTP Data Integration for Operational Level Decision Making

By

Muhammad Azhar Chohan

(2007-NUST-MS PhD-CSE(E)-28)

Submitted to the Department of Computer Engineering
in partial fulfillment of the requirements for the Degree of

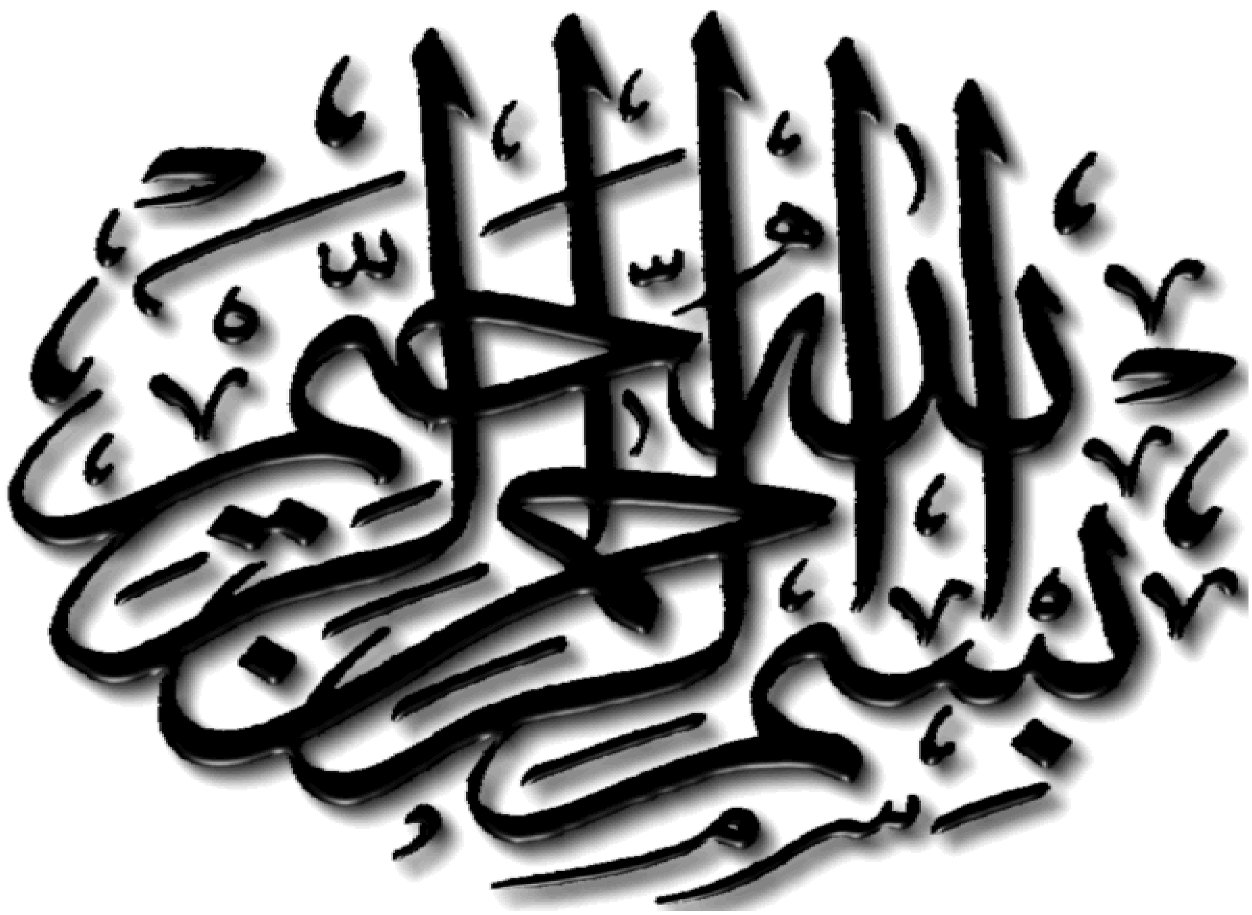
Master of Science
In
Software Engineering

Advisor

Brig. Dr. Muhammad Younus Javed

College of Electrical and Mechanical Engineering

National University of Sciences & Technology


بسم الله الرحمن الرحيم

Certificate

It is certified that the contents and form of thesis entitled "OLAP and OLTP Data Integration for Operational Level Decision Making" submitted by Mr. Muhammad Azhar Chohan have been found satisfactory for the requirement of degree.

Advisor: ___________________

(Dr. Mohammad Younus Javed)

Committee Member: __________________

(Dr. Farooq-e-Azam)

Committee Member: __________________

(Dr. Aasia Khanum)

Committee Member: __________________

(Dr. Saad Rehman)

ACKNOWLEDGEMENTS

First of all I would like to thank Almighty ALLAH who gave me the power to select, work and propose solution in a complex analytical area of research which remained unturned even after decades of focused research.

Nothing worthwhile was ever achieved in isolation. I cannot claim to have written this thesis without the significant influence of others. I would like to thank my supervisor **Brig. Dr. Muhammad Younus Javed**. It was invaluable to have the benefit of his decades of experience in the field of Information Technology standing behind his advising of me. I am most grateful that he allowed me to work with him for this thesis, and to be a part of his research group. . I would like to thank **Lt. Col Dr. Farooque Azam, Dr. Aasia Khanum** and **Dr. Saad Rehman** for serving on my committee.

I thank my parents and family for standing by me in times of crisis and for being patient with my endless years of study.

I have to thank the large circle of friends I've had at EME and outside each of whom have been special and dear. I feel particularly thankful for having had excellent friends over the years, especially Muhammad Asim Nawaz who has been a great support and a loyal friend throughout the years. Asif Nawaz and Ahmad Nawaz deserves a special mention as being close friends and a continuous source of support, always lending a nonjudgmental ear through good times and bad.

Dedicated to my parents and my wife

ABSTRACT

OLAP and OLTP Data Integration for Operational Level Decision Making

Data warehousing is being used by many organizations to make decisions at top level. These analysis-based decisions are more helpful for top management for setting their market trends to compete their market rivals. However, some decisions have to be taken at operational level in an organization such that management should be able to compare current and targeted performance values so they can take proper steps according to the situation. For these decisions to be quicker and accurate, operational data is needed. Such data cannot be gathered from the current warehouse solutions alone as the ETL process of data warehouse is generally done periodically like once every 24 hours.

This research work proposes an architecture that incorporates fresh data into DW without involving delayed ETL process. Architecture consists of three major components, Query Recognizer, Query Decomposer and Query Converter. Query Recognizer sends the query to Query Decomposer after analyzing whether query needs data from OLAP and OLTP. Query Decomposer then decomposes the query into two separate queries, one for OLAP and other for OLTP. The OLTP query is sent to Query Convertor which converts the OLTP query into OLAP query. Results are merged in Data Integrator and send back to user.

To achieve this objective, an OLTP source system has been designed and implemented using oracle database 10g. Then, client side application has been build over source OLTP system using oracle 10g forms. The data warehouse has been designed and implemented using oracle warehouse builder 10g release 2 that is used to store the captured source OLTP data which is used for analysis purpose. Comprehensive testing and evaluation of the developed system has been carried out. Performance comparison of the proposed system has also been carried out with other researcher’s work in the same area. When compared with normal and active data warehouse ETL techniques, the proposed model provides better and more accurate results. The proposed model provides most recent data with 100% accuracy and there is almost 40% decrease in average query response time as compared to active data warehouse model.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

DW	Data Warehouse
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
BI	Business Intelligence
ART	Average Response Time
QRT	Query Response Time
MRT	Median Response Time
MAP	Mean Average Precision

Chapter 1

Introduction

As a response to the changing situation on the global market, businesses are becoming more dynamic. They have changing data requirements that cannot always be anticipated. Changing data requirements is a big problem for current OLAP systems and physically integrating such data in OLAP systems is time consuming and costly [2].

As a consequence of their need to make quick decisions, businesses are also becoming more dependent on fresh data for analysis, especially on lower levels in the organization. Fresh data is important, for example, for organizations that need to make decisions based on quickly clanging data such as stock exchange data, currency rates, the current performance of different processes etc. It is however not possible for the DW to incorporate the real time data "on the fly" through the normal loading procedure called "Extract Transform Load" (ETL) because the DW has to be refreshed before changes can appear [3]. The data warehouse is loaded with new data, or refreshed, after a certain period of time, for example after 24 hours or one week and by the time the data is available for querying, it is too old to be of use for decision making. The need to incorporate quickly changing data in DW, in order to make it available for analysis and decision-making, gave birth to the idea of logical integration of such data with the OLAP system [2] [3].

In addition to becoming more dynamic, businesses today are also becoming more and more process-oriented. Data in the DW about the performance of different processes support the top level management in making strategic decisions for the organization. The strategic decisions can be translated into a number of lower-level goals for the performance of the different tasks that make up the processes. This is done by setting target values for the performance of these tasks. To be able to make decisions in line with the organization's strategy, the lower-level management needs to continuously monitor the performance of the tasks and compare them with the target values. To do that, fresh data about the performance of the tasks is needed. This however cannot be obtained from the DW [3].

1.1 Problem

Current data warehouse solutions are designed for supporting top level management in making strategic decisions. However, decisions are made on three levels in an organization, which are the strategic, tactical and operational levels [3]. Decisions on lower levels in the organization must be faster, and fresher data is needed for analysis than what can be obtained from the DW.

1.2 Goal

The goal of the thesis is to design a new architecture to incorporate external data (real time data, quickly changing data) which is stored in company's OLTP databases with OLAP data so that fresh data is available for querying.

1.3 Purpose

The purpose of the thesis is to design architecture for a data warehouse that provides more recent data for decision support and thereby better fulfills the needs of decision makers on all levels of organization.

1.4 Method

The overall working method of this thesis is to develop and implement an architecture based on literature studies and to evaluate the architecture by first formulating three basic DW queries and applying them on the architecture, and secondly discuss how the architecture satisfies a set of evaluation criteria.

1.5 Develop the Architecture

Based on the literature about query processing and retrieval we then developed the architecture and resolved the issues of query processing. The architecture used different optimized components from different areas and merged them to get new functionality. Mostly concepts are inspired from [2] [3] [16] and [17].

1.5.1 Develop and apply three basic DW queries on the architecture

In order to evaluate the architecture, three basic DW queries were formulated, that then were applied on the architecture. The first query requires data from an operational (OLTP) system, the second requires data from the OLAP system and the third requires data from both OLTP and OLAP systems. The queries were developed together with a researcher in the area of data warehousing.

1.6 Thesis Overview

The thesis begins with the introduction of research work in Chapter 1, and then chapter 2 contains information regarding different data warehouses models and different mechanisms to populate the data warehouse. The Chapter 3 contains the related work regarding conventional and Real Time Data warehouse. The brief and detailed proposed architecture will be discussed in chapter 4. The Chapter 5 is regarding design and implementation of proposed architecture and the chapter 6 is about experimental results and conclusions.

Chapter 2

Literature Review

This chapter entails history of data warehouse, the basic steps that are used to develop data warehouse. The data warehouse is build in order to improve the quality of information in the organization. Data in different structures and formats coming from both external and internal sources is integrated and consolidated into a single repository. Data warehouse system consists of the all components and data warehouse used for construction, accessing and maintaining the data warehouse. Data warehousing several concepts of particular importance are discussed in detail in this chapter.

2.1 Data Warehouse

The processes of an organization produce information which is an asset of the organization. This information is kept in two forms, in OLTP systems and in a data warehouse. An operational system, also called an Online Transaction Processing (OLTP) system, is used to record the daily operations and business transactions. The data warehouse, on the other hand, is used for analysis based decision making [11]. The data warehouse is a huge collection of data from different sources including the OLTP systems. Often external data, for example address lists and other kinds of customer information, is also purchased and incorporated into the DW. One major characteristic of data warehousing is that, unlike the OLTP systems, it keeps historical data. This data can be used to analyze trends and variations.

There are many definitions of data warehousing. Three definitions are given below.

"*Data warehousing is collections of decision short technologies, aimed at enabling the knowledge worker (Executive, Manager, Analyst) to make better and fast decisions*. " *[1]*

"*The conglomeration of an organization's data warehouse staging and presentation areas, where operational data is specifically structured for query and analysis performance and ease of-use. " [11, p 397]*

*A data warehouse is a **subject oriented, integrated, time-variant, and nonvolatile** collection of data that supports managerial decision making "[5,p54]*

Subject-oriented: An enterprise in general hold data that is very comprehensive to fulfill every requirement required for different departments of the company (marketing dept, human resources dept, sales dept, etc.) and optimized for transaction processing. Generally, for decision-makers this type of data is not valuable. Subject-oriented data is used by decision-makers. The Data Warehouse only contains key business information. The data in data warehouse like to be organized based on different subjects and warehouse is populated only with subject-oriented data. If all information about a specific product is required by the decision-maker, he should utilize every system similar to catalog sales system, rental sales system and order sales system, that is not the practical and the preferable way. Alternatively, data warehouse is developed by all the significant information and structured into subject areas as illustrated in Figure2-1.

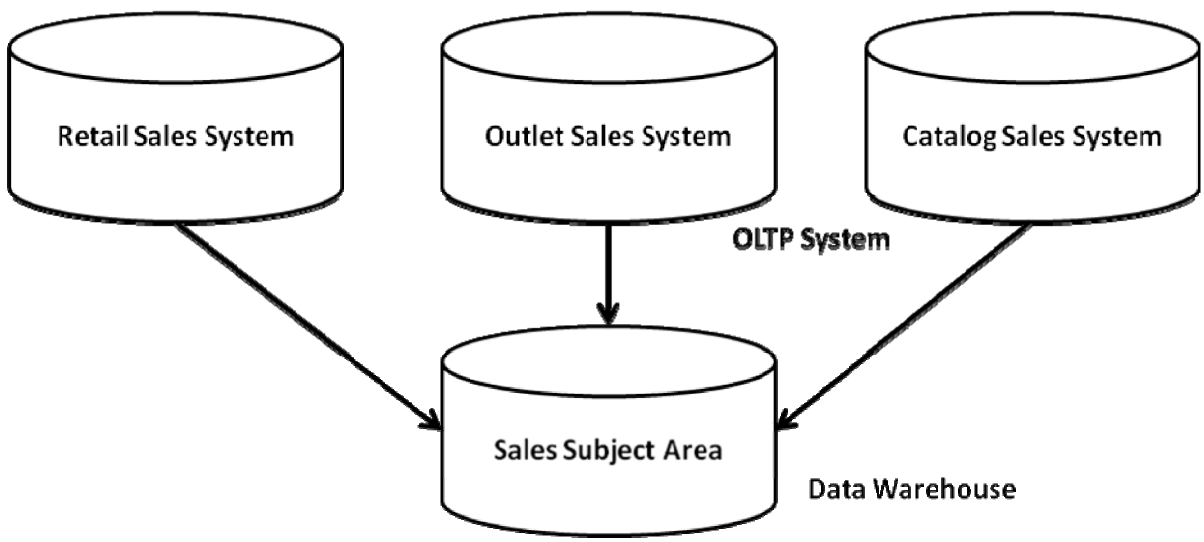


Figure 2-1 Consolidation of OLTP information

Integrated: Data warehouse is developed by integrating data from several heterogeneous source systems (like data from the legacy systems, XML data, excel sheets, flat files, relational database) in order to provide decision making, analytical reporting and structured and/or queries. Data warehouse also support methodologies for standardizing and cleaning data. Figure 2-2 describe a range of formats and uses of PRODUCT CODE attribute.

OLTP Systems		
Retail Sales System	Outlet Sales System	Catalog Sales System
Product code:	Product code:	Product code:
9999999	XXXXXXXXX	XXXX99.99

Figure 2-2 same attribute with different formats in different sources

Time-variant: Data warehouse stores historical information. In the Data Warehouse an element of time is either explicitly or implicitly included for every key structure. A data warehouse normally contains 5-10 years old data that is used for forecasting, comparisons and trends.

Nonvolatile: Data in the data warehouse are not changed or updated, so it does not need concurrency control, transaction processing and recovery and mechanisms. Figure 2-3 show this behavior of data warehouse.

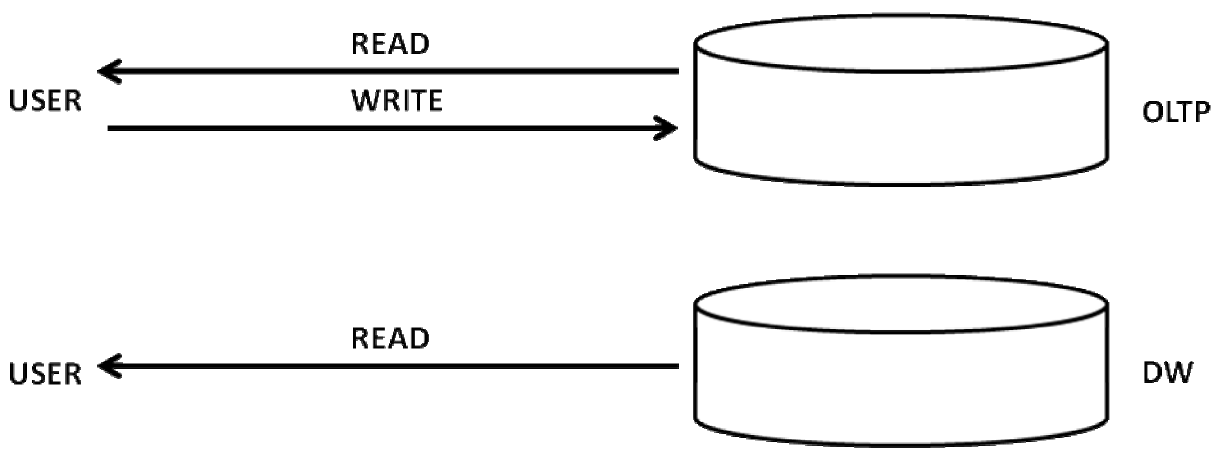


Figure 2-3 Simple Comparisons of OLTP and Data Warehouse Systems

2.2 Data Warehouse Architecture

In data warehousing, data is compiled from different sources. The sources vary depending on the needs of the organization. The sources can be operational systems which lie inside or outside the organization, other data warehouses, or data can be purchased from companies that specialize in keeping and selling up-to date data in the form of address lists etc.

A process called "Extract Transform Load" (ETL) is used to extract data from external sources, transform the extracted data to the format needed in data warehouse and finally loading it into the data warehouse. ETL is also known as the Data Staging Area. There are lots of tools available in the market for ETL. The data in the DW is presented with the help of presentation servers such as OLAP, which will be described in the next section. The data warehouse can also contain Data Marts. A data mart is a subset of a data warehouse which is targeted to a specific process and contains a low volume of data. Collections of data marts together make up an enterprise data warehouse. Different front end tools are used by end users for querying and getting information from the data warehouse.

Data warehouses are in general consist of the front stage and the back stage. For end-users who are using decision support tools, the data warehouse acts as front stage. The administrators at back stage are concerns with the task to propagate data from different OLTP source systems to the data warehouse. The data warehouse architecture consists of different number of layers and data of one layer is derived from the previous layer data (Figure 2-4).

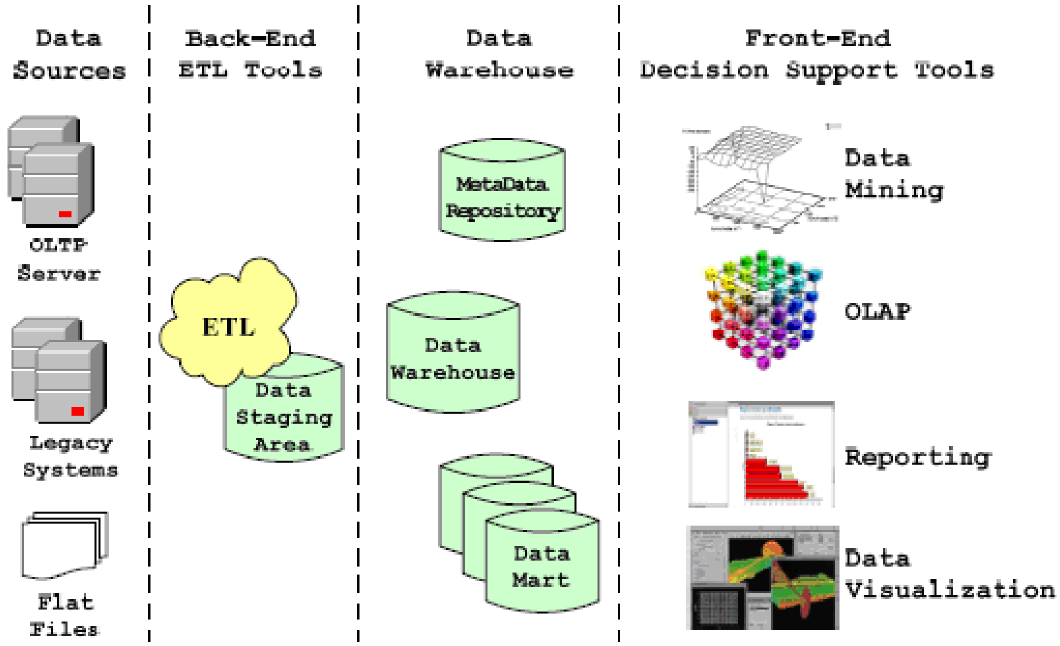


Figure 2-4 Data Warehouse Architecture

2.2.1 Components of a data warehouse

According to William Inmon five different data components can be found within a DWH:

- Metadata
- Current detail data
- Old detail data
- Lightly summarized data
- Highly summarized data

[Inm, 2000c]

2.2.1.1 Metadata

Metadata are data to describe data warehouse components [e.g., their schema] and their relationship within the data warehouse. "*In many ways metadata sits in a different dimension than other DWH data, because metadata contain no data directly taken from the operational* *environment*” [Inm, 2000c]. Nevertheless, metadata play an important role in a DWH. More on metadata can be found in chapter 2.6 ‘Data warehouse metadata’.

2.2.1.2 Current detail data

Current detail data are “*the major concern of DWH applications, because they reflect the most recent happenings*“[Inm, 2000c]. They are however voluminous due to the fact that they are data at their lowest level of granularity. Nevertheless current detail data are usually stored on disk storage to grant fast access at the price of high cost and increased complexity to manage.

2.2.1.3 Old detail data

Old detail data are current detail data that have passed the time frame of immediate importance and are therefore, and especially for the enormous data volume reason [old detail data differ from current detail data not logically, they are composed of the same level of granularity of data but from a different point in time historically speaking] migrated to a different, less costly storage space.

2.2.1.4 Lightly summarized data

Lightly summarized data are distilled the current detail data. They are an intermediary step towards highly summarized data and “*almost always stored on disk storage*” [Inm, 2000c].

2.2.1.5 Highly summarized data

Highly summarized data are the compact and easily accessible data that were aggregated by analysts and other power users to serve the information needs. “*Sometimes highly summarized data are found in the data warehouse environment and in other cases highly summarized data are even found outside the immediate walls of the technology that houses the DWH. Regardless of where the data are physically housed, highly summarized data are part of the data warehouse*” [Inm, 2000c].

2.2.2 Data warehouse Modeling

The fundamental modeling choices for a DWH were until recently a relational or a multidimensional model. Before delving into the difference of the two choices and the state-of-the-art hybrid approach, a look at the different practical levels of data modeling has to be taken.

2.2.3 The three different levels of data modeling

To support the different requirements of a DWH project three different views are taken upon the data in a DWH and consequently translated into a three level model of any DWH. The Semantic Data Model (SDM) shows a high level view of the DWH project; a model in the language of the DWH end-user. The Logical Data Model (LDM) shows entities and their relationships in a logically sound manner, to serve as model for physical implementation. And finally the Physical Data Model (PDM) shows the actual representation of the physical tables in the database as they are implemented.

2.2.3.1 The semantic level

The semantic level, sometimes also labeled conceptual level, is described by a semantic data model. An SDM aims to create an abstract representation of the situation under investigation, or more precisely, the way users think about it. Analyzing the domain of user concepts should be done independently of any software technological considerations. This guiding line is embodied in the Conceptualization Principle [International Organization for Standardization, 1987] and in the Unified Modeling Language (UML) world of software engineering the semantic level and the SDM would translate to 'Use cases' and 'Use Case Analysis'.

2.2.3.2 The logical level

The logical level is described by the logical data model. An LDM's purpose is to describe entities and their relationships to facilitate the integration of data from the disparate source systems and to enable the data analyst to normalize data elements found in those source files. It is completely process independent and only captures the objects and actions as they exist in the real world for the organization. True integration ["one *fact in one place*"] can only be accomplished with an LDM, not in a physical database because performance and other considerations demand that data is stored redundantly [AdMo, 2000].

2.2.3.3 The physical level

The physical level is described by the Physical Data Model. A PDM shows the actual representation of the tables in the database as they are implemented in the project. The PDM serves as a guide through an existing database, for programmers, administrators and analysts.

2.2.4 Relational data modeling and the Entity-Relationship model

Relational database theory is centered on the process of [data] normalization. Normalization delivers a set of storage structures that minimize the effect of data anomalies. The inputs to the normalization process are the so-called data dependencies. Much of relational database theory is about the formal manipulation of data dependencies. Relational theory assumes the existence of a universal data dependency that is decomposed in the design process in its entities [tables] and their relations. This process is called 'Normalization by Decomposition' because in progressing from one normal form [another key term in relational database theory] to the next larger tables are decomposed into smaller tables. Beyond Third Normal Form the resulting design can depend on the order of decomposition. This, though, violates the Conceptualization Principle because the choice of a designer has effect on the structure of the designed model. No conclusive guiding line has been presented so far in theory. The Entity-Relationship model, however, is the most popular technique to visualize relational data models. Classes of objects are identified and modeled as entities. Entities have properties that are either attributes that directly describe entities or can be relationships that model facts about entities. Entities, relationships and attributes are depicted visually in an Entity-Relationship diagram.

2.2.5 Multidimensional data modeling

The multidimensional data model is based on the key concepts cube, dimension and hierarchy. The cube is the concept that describes the whole of the data which are presented along labeled edges of that cube, the dimensions. Whereas hierarchies define the way in which dimensions are grouped. This dimensional modeling approach results in a database design that is consistent with the paths by which users wish to enter and navigate a DWH. Frequently requested aggregates, or calculated measures, are stored in the database, creating useful data redundancies that make it possible to avoid performance-inhibiting repetitive calculations every time a report is prepared.

2.2.5.1 Cube

A [data] cube is a representation of multidimensional data defined by a set of k different dimensions. Data can be organized into a data cube by calculating all its possible aggregations [SQL statement GROUP-BY]. The set of aggregations form a k-dimensional data cube.

2.2.5.2 Dimension

The dimensions of a cube depend on the data to be modeled in the database. Typically, each dimension is an independent entry point or mechanism for selecting data. For example, a cube for a classic DWH application has the following three dimensions: product, market and time. A cube in a medical context on the other hand could have time, diagnosis and treatment as dimensions.

2.2.5.3 Hierarchy

A Hierarchy defines the way in which a dimension can be grouped in a DWH. For example, a possible hierarchy for the time dimension could consist of year, quarter, month, week and day.

2.2.6 Multidimensional modeling techniques

The two prominent data modeling techniques for the multidimensional model are the Snowflake Schema and the Star Schema.

2.2.6.1 The Snowflake Schema

A snowflake schema is the traditional multidimensional model implemented on a relational database management system (DBMS). It consists of a central fact table containing the measures or some other content of interest, and is surrounded by dimension entities containing conformed context for the analysis of the fact. The dimensions usually relate to the facts in one to many relationships and the snowflake schema exposes them as fully normalized structures usually consisting of many entities with often complex intra dimensional relationships. An advantage of the Snowflake Schema is the clear exposure of the dimensional hierarchy which often directly relates to the aggregation levels that can be applied to the fact. For example, the time dimension may consist of a hierarchy representing year, quarter, month, week and day. This hierarchy is directly related to the temporal aggregations that can be applied to the fact, that is, view the facts consolidated by year, quarter, month, week or day. This is easy for the end user and supports an ad hoc analysis environment. The main disadvantage of the Snowflake Schema is the large number of tables that have to be joined to support even the most basic queries.

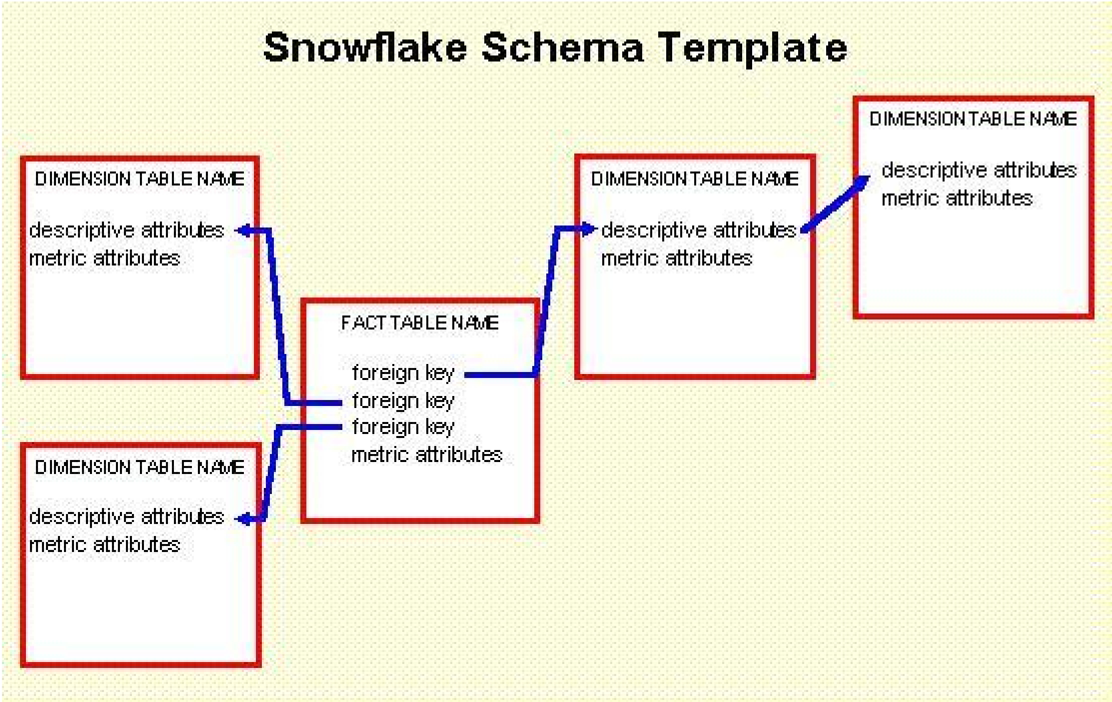


Figure 2-5 Snowflake Schema

2.2.6.2 The Star Schema

Star Schemas were a direct reaction to that disadvantage of Snowflake Schemas. They were designed to produce models that boast a minimum join distance. Therefore the Star Schema simply consists of the fact surrounded by a single level of collapsed or consolidated dimension entities. For the example, the information representing year, quarter, month, week and day would all have to be effectively represented in a single entity. It is obvious that this will usually lead to better query performance as queries will now join fewer tables, a benefit to the RDBMS that does not perform multi-table joins efficiently.

The disadvantage of the Star Schema, like any denormalized model, is the hiding of relationships like the dimensional hierarchy or consolidation path so evident in the Snowflake Schema. The Star Schema will also introduce redundancy and processing anomalies that require a more skilful handling of queries. Which modeling technique is better suited therefore depends on the project at hand. A star schema id represented in fig 2-6.

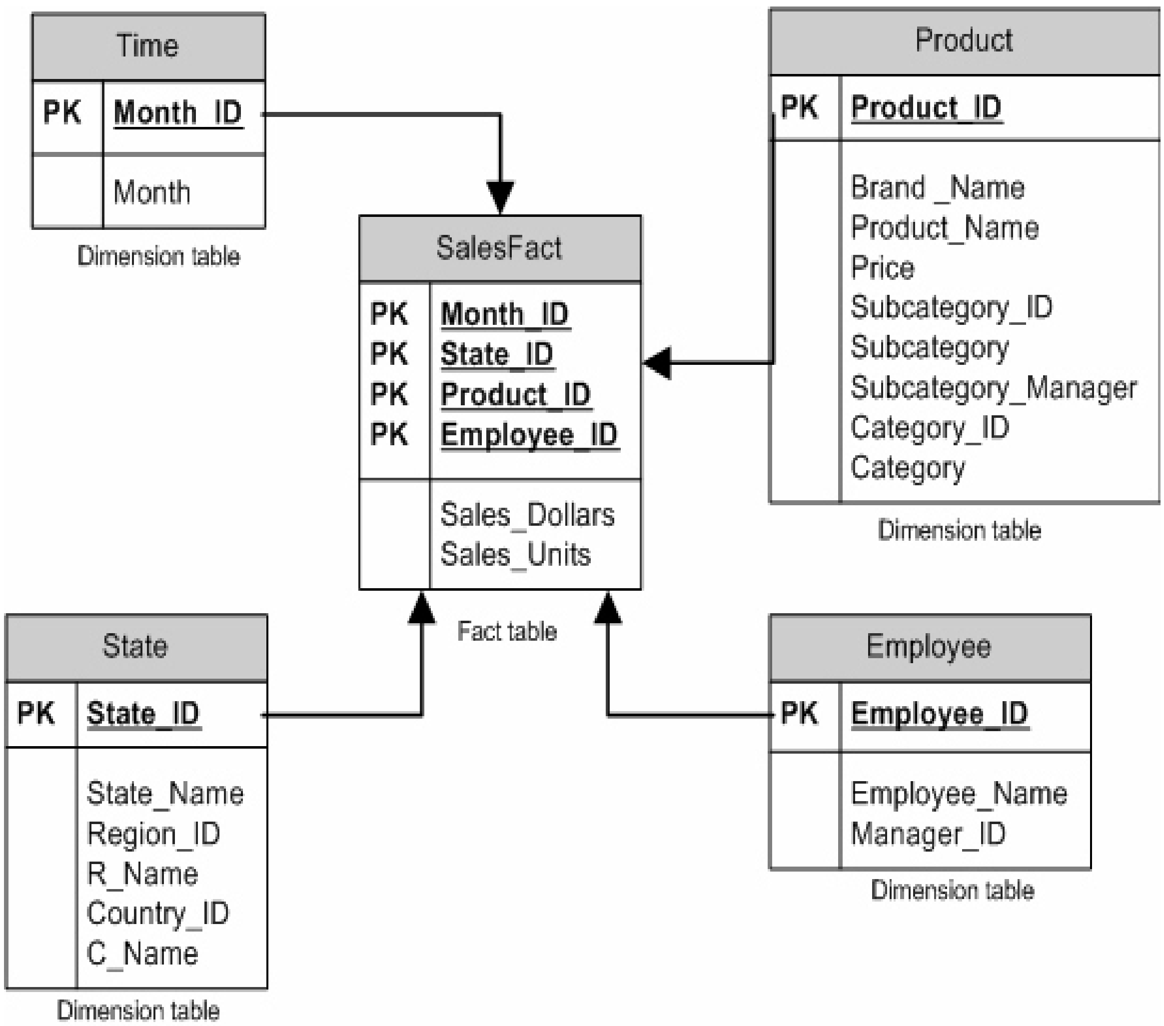


Figure 2-6 Star Schema model

2.3 Business Intelligence

Golfarelli et al [3] define Business Intelligence (BI) as "the process of turning data into information and then into knowledge". The knowledge of interest to organizations often concerns customers and their needs, the competition, and different trends, among other things.

The technology of BI is based on customer and profit oriented models. This means that BI targets to improve relationships with customers. It is used to reduce operational costs and provide increased profit by improving sales, productivity, and customer services. the concept of 131 came in the early 90's when managers of different organizations felt the need for a technology which could help them to analyze organizational data to better understand the situation of their businesses and to make better decisions at right time [14].

BI soon became popular among business organizations and these days, it is known as a combination of different technologies such as data warehousing Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP) etc [10].

Chapter 3

Related Work

3.1 Introduction

In this chapter some terminological issues are discussed by writing few lines about it. The term semi real-time data warehouse has been selected although for this name diverse variants already exist. This chapter also includes the related work of different researchers in the field of both conventional and active data warehouse.

3.2 Online Analytical Processing (OLAP)

Online Analytical processing (OLAP) is a technique for structuring and presenting the data in the DW. It supports ad-hoc querying. Users can manipulate and visualize data through multidimensional views. Multidimensional views means that data can be viewed from different perspectives. For example sales of a store can be viewed from the perspective of products, date, region or a combination of all of them. In OLAP data is stored in multidimensional cubes, see figure 3-1 below.

Figure 3-1 shows an OLAP cube which consists of different dimensions like product, time and market. It also has a sales fact. The sales fact can be calculated from different perspectives like product and time or product and market etc.

- Dimension

  Can be defined as a business perspective from where data is looked upon. A dimension is a collection of attributes that are highly related with each other [11].

- Measures/Facts

  Are variables which are related to the aspects (or dimensions) of a business. It consists of numeric facts and measures.

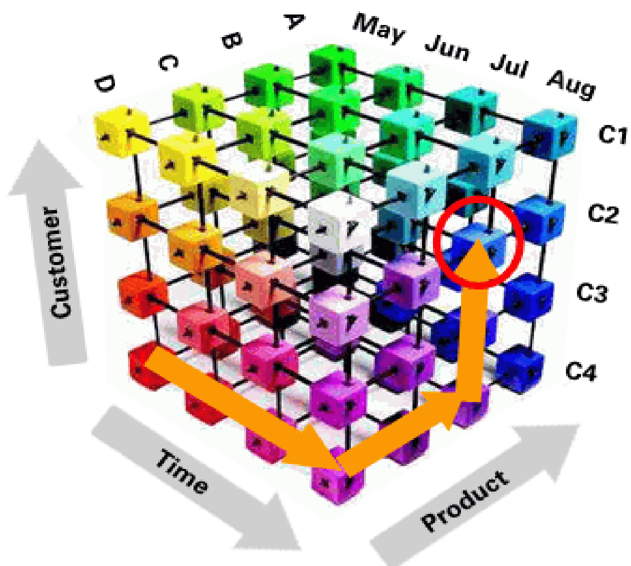


Figure 3-1 OLAP Cube

3.3 External Data

External data can be integrated in the DW in two ways. One way is to load it into the DW through ETL which means physical integration with the DW. There are certain reasons why physically storing data in current SLAP systems is not feasible or not possible. One reason is that it is not always possible to anticipate all data requirements when designing a database schema. Physically integrating unanticipated data can be time consuming and costly, as it would involve changes in the database schema and the ETL process. If data requirements change frequently this might not be a feasible solution. A system also becomes too complicated if someone tries to anticipate many future data requirements [2]. Another reason why physical integration may not be a possible solution is that sometimes the data needs to have a certain freshness, which is lost by the time the data has gone through the ETL process and is available in the DW [3].

The other way to integrate external data into the DW is to make a logical integration of external data with OLAP [2]. The concept is to extract data on the fly from the external sources when there is a request from the client system. The query is divided into two parts, one part of the query is sent to the OLAP server and other part is sent to the external data system. The results from the query to the OLAP server are then merged with the result from the query to the external data system and the final result is passed to the user. These external data sources can be, for example, operational systems [3] or data that is fetched from the web [2].

3.4 OLAP-XML Federation

The concept of "OLAP-XML federations" is presented in research by Pedersen et al [2] as a solution to the problem of physically integrating unexpected or quickly changing data into the DW. They introduce the idea of logically integrating such data with data from the DW. External data is available on the web and will be extracted on the fly.

These days many business to business (B2B) applications use Extended Markup Language (XML). The increased use of XML data and its popularity indicate that it would be the format of fixture data available on web [2]. The theme is to make OLAP more flexible to deal with quickly changing data; work is being done to logically integrate external data with OLAP databases. Since almost all data sources can be effectively wrapped in xml, flexibility is ensured using XML as the external data format [6].

Figure 3-2 below shows architecture of an OLAP-XML federation. A federation manager is placed between the user interface and the OLAP component. The federation manager receives the request in SQL_{XM} form (a special extension of Structured Query Language, SQL) and decides where to find the data required for the query. It divides the query into two parts, and passes one query to the OLAP component. The other part of the query is transformed into XPath (a query language which is used to extract data from XML documents). The XPath query is sent to the external XML data source available on the web. Finally extracted data from both sources is merged in a temporary database, and the result is sent back to the user.

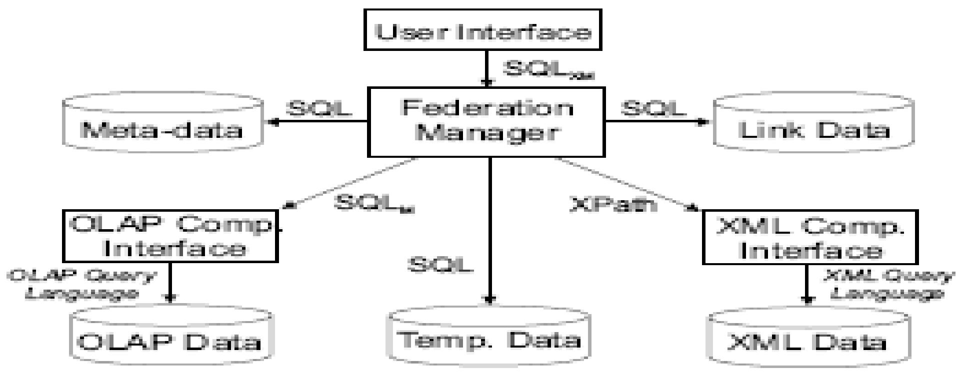


Figure 3-2 Architecture of the federation system

3.5 Query Decomposition

The process of querying multiple databases or parts of database is a complex task. In order to use single query for retrieval of data from multiple data sources, the concept of breaking a query into smaller parts is used. In distributed databases this process of dividing a transparent query into fragmented queries is called localization [16, p 200].

Pederson et al [2] has implemented the concept of dividing (decomposing) a query into sub queries for querying the OLAP and XML sources as shown in Figure 3-2.

3.6 Query Mapping

It is really a challenge to integrate different database schemas. Global schema proposed in [17] presents a complete picture of component databases when schemas of component databases are integrated. This global schema contains all information about the structure of local databases. So, different transformation rules could be defined to map the query according to its respective database. The concept of integrating metadata of heterogeneous databases into one global metadata has been implemented by [17].

Pederson et al [2] also implemented the concept of mapping queries and converting extended SQL queries into OLAP and XPath queries.

3.7 Business Process Management (BPM)

BI has recently given birth to a new concept known as Business Performance Management (BPM). The concept of BPM not only includes a Data Warehouse but it also requires an additional component which can monitor different operational processes which are vital for an organization. The BPM scenario introduced by Golfarelli et al [3] describes that, based on information from the DW, the top level management in an organization sets target values for the performance of operational processes. These are then translated into a number of target values for the different tasks that make up the processes. The tactical and operational level decision makers responsible for the time-critical operational tasks need fresh data at the right time to monitor the performance of the tasks. Based on this fresh data, the tactical and operational level management are able to compare current and targeted performance values so they can take appropriate steps according to the situation [3].

Figure 3-3 explains a scenario where data warehouse is used to extract data from information systems which keep transactional data of an organization. The data warehouse keeps historical data. The strategic level of organization use data warehouse to analyze data. Based on this analysis, they set target values for different tasks of each business process. Managers on different levels then compare the target values with current performance values and make decisions based on this information [3].

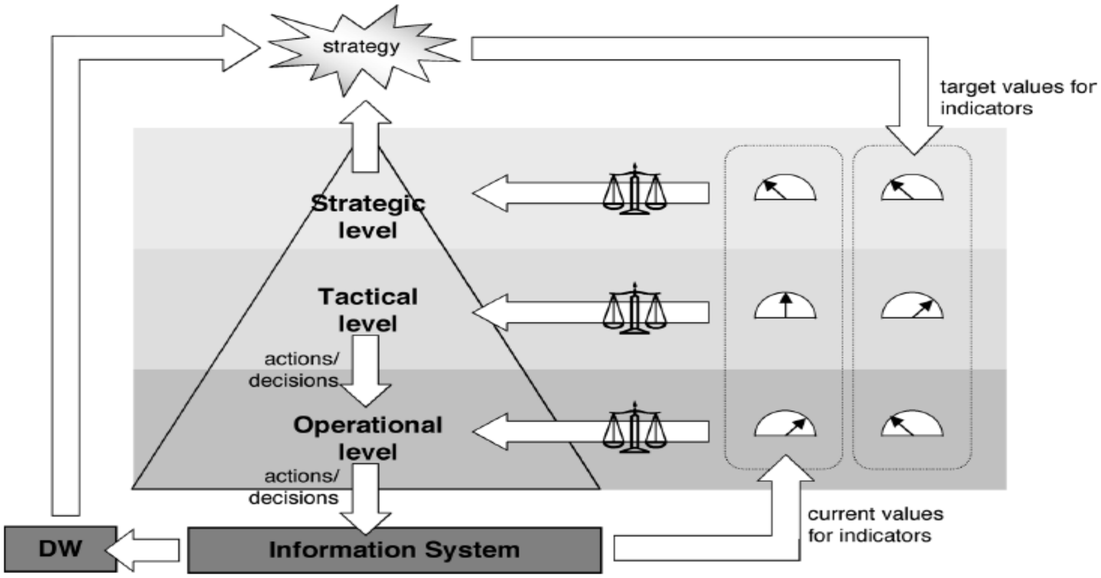


Figure 3-3 the closed-loop in BPM approach [3]

Chapter 4

Proposed Approach

The proposed architecture is different from traditional data warehouse architectures. Architecture assume a situation in an organization where historical data is very important to analyze for better decision making on the strategic level, but fresh data is also needed "on the fly" to support better decision making on all levels of management in an organization. Our proposed architecture extracts historical data from OLAP cubes as well as fresh data from OLTP systems "on the fly" if required. The proposed architecture thereby offers more flexibility than current data warehouse architectures.

The proposed architecture is presented below. It is explained on two levels for complete understanding, first on an overview level (fig 4-1) and then on a more detailed level (fig 4-2).

4.1 Level-1

Figure 4-1 shows an overview of the proposed architecture. This architecture is an extension of conventional data warehouse architecture. The ordinary components of data warehouse architecture are End User Tools, OLAP, Data Warehouse, ETL and OLTP. A description of the additional components of the DW architecture is given below.

4.1.1 Controller

The controller is the core component of this new DW architecture. It is situated as a layer between the end user tools and the OLAP cube. It acts as a manager which facilitates and coordinates different processes during the execution of the query sent by the user. One of the main activities of the Controller is to recognize that either fresh data from OLTP systems is required for this query or historical data is needed from the OLAP cube. It can decompose the query into two parts and send each part to the right system for processing. Its other main function is to convert the query from an OLAP to an OLTP query. It also has a temporary database where it temporally stores results received from the OLAP and OLTP systems. The final results are compiled there and sent to the end user.

4.1.2 Data Integrator (DI)

The Data Integrator or DI, manages all the external data stored in the OLTP system. This component assures that all data is available at the right time to facilitate the analysis. The DI component can also be extended so that it can integrate data from different sources like other DW, file systems etcetera and make the llama available for analysis.

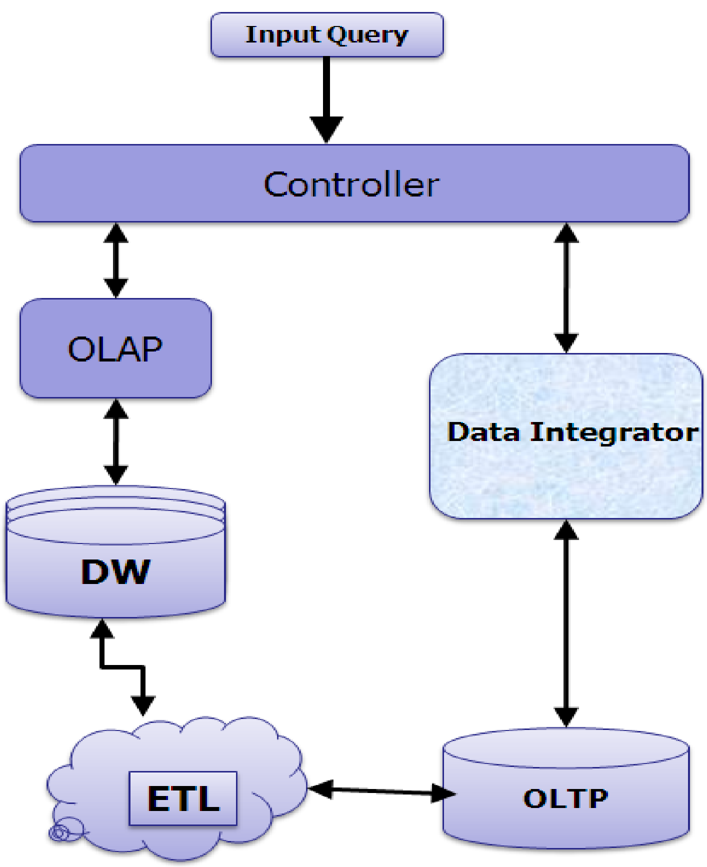


Figure 4-1 Overview of purposed architecture

4.2 Level-2

The proposed architecture in figure 4-1 is explained in more detail in figure 4-2. Figure 4-2 gives the full picture of all components in the architecture. A details explanation of the additional components in our architecture is given below.

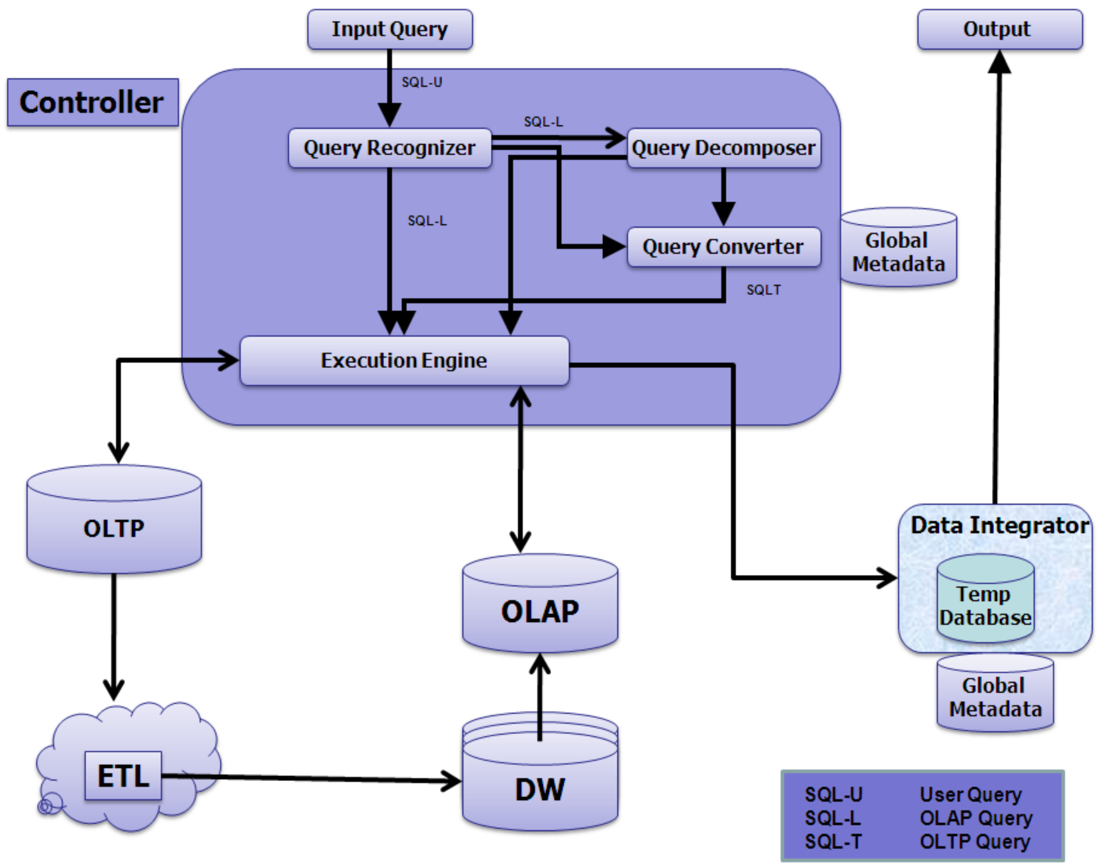


Figure 4-2 Complete Architecture

4.2.1 Controller

- ***The query Recognizer*** is used to check whether the query from the user needs data from OLAP or OLTP systems or from both. If only historical data is required then the Query Recognizer passes the query directly to the execution engine which sends it to the OLAP system. On the other hand if the query needs fresh or real time data from the OLTP system as well as historical data from the OLAP system, then the query recognizer invokes the query decomposer to divide the query into two parts. If only fresh data is needed then the Query Recognizer sends it directly to the Query Converter. No transformations are done in the query recognizer.

- ***The Query Decomposer*** divides the OLAP query into two OLAP queries. One is sent to the execution engine to extract required data from the OLAP system while the other is sent to the Query Converter to convert it into an OLTP query.
- ***The Query Converter*** receives the OLAP query from the Query Decomposer and converts it into an OLTP query using Metadata which has information about OLTP systems. The query is then sent to the execution engine.
- ***The Execution Engine*** accepts queries from upper components and executes OLAP queries on the OLAP cube and OLTP queries on fresh data present in DI which comes from OLTP systems. When the execution engine get results from OLAP and DI, it stores these results in a temporary database and the final result is compiled there and is then sent back to the end user tools for presentation to the user.
- ***The Temp Database*** is used to store results received from both OLAP and OLTP sources. The final result is compiled here based on the initial results and sent back to the user.
- ***The Global Metadata*** contains information about the structure of data in the data warehouse, the OLAP and OLTP systems.

4.2.2 Data Integrator

- *The Data Integrator (DI)* integrates at 'the right time data from operational databases. "Right time" means that data is extracted from the OLTP system when required in order to always get fresh and up to date data [3].

4.3 Origin of Idea

All the internal functionality of the components in level-l was further explained in level-2. We did not develop any new components of our own but organized and modified different existing and optimized components described in the literature to achieve our objectives. The syntactic and semantic analyzer is described by Ozsu and Valduriez [16, p 198]. The query recognizer was inspired by Pedersen et al [2]. Query decomposition is a process which is mostly used in distributed databases [16, p 200] and Pederson et al [2] [6] also use a decomposition process. The idea of a query converter comes from [2] [17]. The temporary database component is a core idea obtained from [2] [6]. Right time data (RTI) Originates from [3] and the concept metadata come from [17].

4.4 Limitations

There are some limitations in the architecture in figure 4-2. The RTI can be extended to integrate data from different sources, for example different operational sources, other data warehouses and file systems etc. In this thesis we however only consider integrating data from one operational source OLTP system.

4.5 Query Processing

A process is something which produces an outcome or result. So a process can be said to be a designed sequence of operations which take time, resources, space and produce a result [13]. In this section we will describe different components of the architecture proposed in section 4.3 and how these components interact with each other through algorithms, and describe the general process behind each component during query processing. We give an example of a scenario and will explain process details keeping this example in mind. In this section we try to give the reader a deeper understanding of the architecture presented in the previous sections. The example given below is one of many scenarios that exist in large organizations.

4.5.1 Example Scenario

Assume a big retail store company ABC having stores throughout Europe. ABC has regional warehouses to supply goods to these stores. ABC maintains a data warehouse and an OLAP cube showing goods in stock with dimensions Store, Product, Quantity, Date and the measure quantity. Assume that ABC data warehouse gets loaded after 24 hours during night.

Some high selling, products are: sold very quickly in stores and there are complaints that often these products become unavailable in stores.-The top level management decides that lower level management should order fresh stock from main warehouse when current stock of high selling products comes down to 30% of normal stock levels Now lower level management needs to keep an eye on current stock of high selling products. For that reason they need fresh data at the right time to analyze the situation and to fulfill the goals according to the guidelines of top level management. So, fresh data on the fly is needed from external sources that are the OLTP systems.

In the following sections we will describe query processing through algorithms. When the user sends a query it is received by the Coordinator (see Figure 6). How different components of the Coordinator process the query is explained below.

4.5.2 Global Metadata

The Global metadata component contains all the information about the OLAP system (GM_A) and the OLAP system (GM_T). The metadata contains business, technical and transformational metadata. Business metadata is mainly needed by end users, for example end user specific documentation, dictionaries, details about predefined queries, user reports and thesauri etc. Technical metadata contains schema definition, configuration specifications, physical storage information and access rights etc. Transformational metadata contains information like data processing, information regarding the loading and refreshment process, the analysis process and inherent administration of the data warehouse system [7].

The Global metadata also contains the transformational rules and mapping of data between the DW and the OLTP systems. This information will be needed by the query converter component to convert the OLAP query into an OLTP query.

4.5.3 Query Recognizer

This is used to check whether data needed is from OLAP or OLTP or from both. This decision is based on the date on which last ETL process was done and data was loaded in DW. If the load date is greater than all of the given dates in the input query then it means historical data is needed, and then query is send directly to execution engine. If load date is smaller than all of the given dates in input query then it means only fresher data is needed. The query is sent to Query Converter. If the load date lies within given dates of input query then query is sent to Query Decomposer.

We have written an algorithm in relational algebra to define the process of Query Recognizer.

Where

SQL_E User query

Tname variable which stores all the table names from the query

GM_A Global Metadata Information about OLAP (regarding tables and attributes)

TableN variable of link list that stores the table names of table existing in the same system (OLAP or OLTP)

Attname variable which stores all attributes from the query

AttN variable of link list that stores Attributes from the query associated with the table link

GM_T Global Metadata Information about External Source

Algorithm 1 Recognizer ();

Result: Identify the query and direct query towards targeted component.

fetchQueryInfo (); //fetches all table names, attributes and conditions within the SQL_E

```
if ((date_att ∃ in SQL_E)   Λ  date_att   ε  SQL_condition) then
{
   getDateGM_L();        // gets date and time for last load to warehouse from transformational metadata
       if (GM_L.date  ≥  SQL_E .date) then
            if (flag = = 0)
                 func executionEngine();    // query based on historical facts
            else
                 dFlag ←—— 0    func decomposition();    // query based on both sources
       else
            func queryConverter ();             //query based only on fresh data
       if ((SQL_E .date ∃ in (GM_L.date  ε   (GM_T.date  Λ  GM_A.date) )) then
            if (flag  = = 0)
                  dFlag ←—— 1 ; func decomposition ();
            else
                  dFlag ←—— 2 ; func decomposition (); // query based on both sources
            }
            else
                 if (flag = = 0)
                 func executionEngine ();
```

else

dFlag ⟵ 3; func decomposition ();

4.5.4 Query Decomposer

Depending on the decomposeFlag in Recognizer() function the input query is divided into two OLAP queries. First OLAP query will require data up to load date and will send this query to execution engine. The second query requires data after load date, so this query will be sent to Query Converter to convert it into OLTP query.

4.5.5 Query Converter

After consulting Global Metadata query is converted to OLTP query. Query is converted into OLTP query after the mapping of table and columns by consulting Global Metadata. An example of mapping is depicted in figure 4-3. The OLTP query is then sent to the execution engine. Global metadata contains transformational metadata where all mapping is stored. It means that it stores information about the sources, tables and attributes of OLTP, which the warehouse attribute originates from. When the Query Converter receives a query, it applies table and attribute mapping on the query and converts the query into an OLTP query.

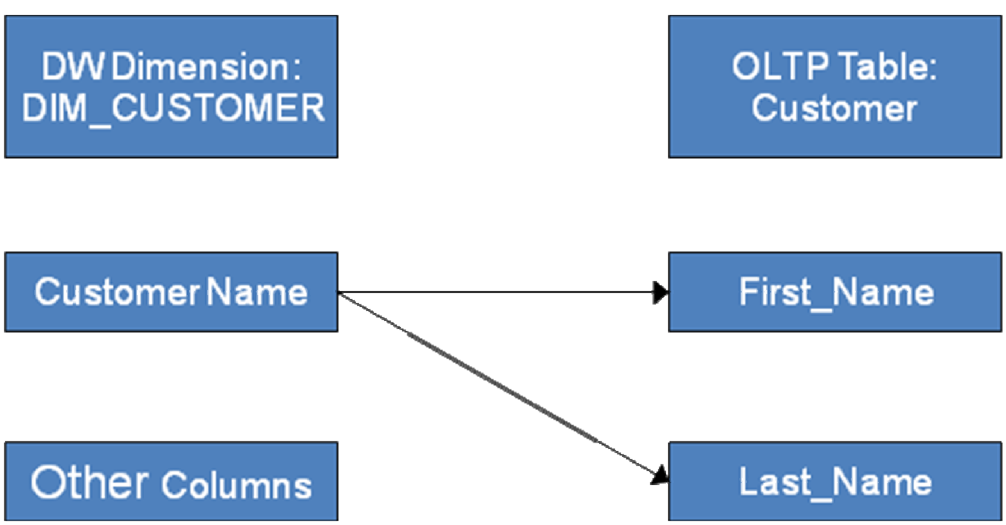


Figure 4-3 Column Mapping

Chapter 5

Design and Implementation

In the previous chapters, we discussed the working of the proposed architecture with the help of diagrams and algorithms. This chapter discusses detailed implementation of the proposed architecture. We start with the implementation of OLTP source system and its application. Then we implement DW using Oracle Warehouse Builder 10g realease2 and construct logical cubes of sales and purchase. We then develop a front end application in Oracle IDS 10g release2 for the implementation of the architecture. The description will cover the challenges faced during the implementation of these modules.

5.1 Design and Implement OLTP Source System

In our thesis work we have to first design and implement source OLTP database. Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

5.1.1 Design Process

The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Not all of these steps will be necessary in all cases. Usually, the designer must:

- Determine the relationships between the different data elements
- Superimpose a logical structure upon the data on the basis of these relationships.

Within the relational model the final step can generally be broken down into two further steps that of determining the grouping of information within the system, generally determining what are the basic objects about which information is being stored, and then determining the relationships between these groups of information, or objects. This step is not necessary with an Object database.

The tree structure of data may enforce a hierarchical model organization, with a parent-child relationship table. An Object database will simply use a one-to-many relationship between instances of an object class. It also introduces the concept of a hierarchical relationship between object classes, termed inheritance

5.1.2 Determining data to be stored

In a majority of cases, the person who is doing the design of a database is a person with expertise in the area of database design, rather than expertise in the domain from which the data to be stored is drawn e.g. financial information, biological information etc. Therefore the data to be stored in the database must be determined in cooperation with a person who does have expertise in that domain, and who is aware of what data must be stored within the system.

This process is one which is generally considered part of requirements analysis, and requires skill on the part of the database designer to elicit the needed information from those with the domain knowledge. This is because those with the necessary domain knowledge frequently cannot express clearly what their system requirements for the database are as they are unaccustomed to thinking in terms of the discrete data elements which must be stored. Data to be stored can be determined by Requirement Specification.

5.1.3 Conceptual schema

Once a database designer is aware of the data which is to be stored within the database, they must then determine where dependency is within the data. Sometimes when data is changed you can be changing other data that is not visible. For example, in a list of names and addresses, assuming a situation where multiple people can have the same address, but one person cannot have two addresses; the name is dependent upon the address, because if the address is different than the associated name is different too. However, the other way around is different. One attribute can change and not another.

5.1.4 Logically structuring data

Once the relationships and dependencies amongst the various pieces of information have been determined, it is possible to arrange the data into a logical structure which can then be mapped into the storage objects supported by the database management system. In the case of relational databases the storage objects are tables which store data in rows and columns.

Each table may represent an implementation of either a logical object or a relationship joining one or more instances of one or more logical objects. Relationships between tables may then be stored as links connecting child tables with parents. Since complex logical relationships are themselves tables they will probably have links to more than one parent.

In an Object database the storage objects correspond directly to the objects used by the Object-oriented programming language used to write the applications that will manage and access the data. The relationships may be defined as attributes of the object classes involved or as methods that operate on the object classes.

5.1.5 Physical database design

The physical design of the database specifies the physical configuration of the database on the storage media. This includes detailed specification of data elements, data types, indexing options and other parameters residing in the DBMS data dictionary. It is the detailed design of a system that includes modules & the database's hardware & software specifications of the system.

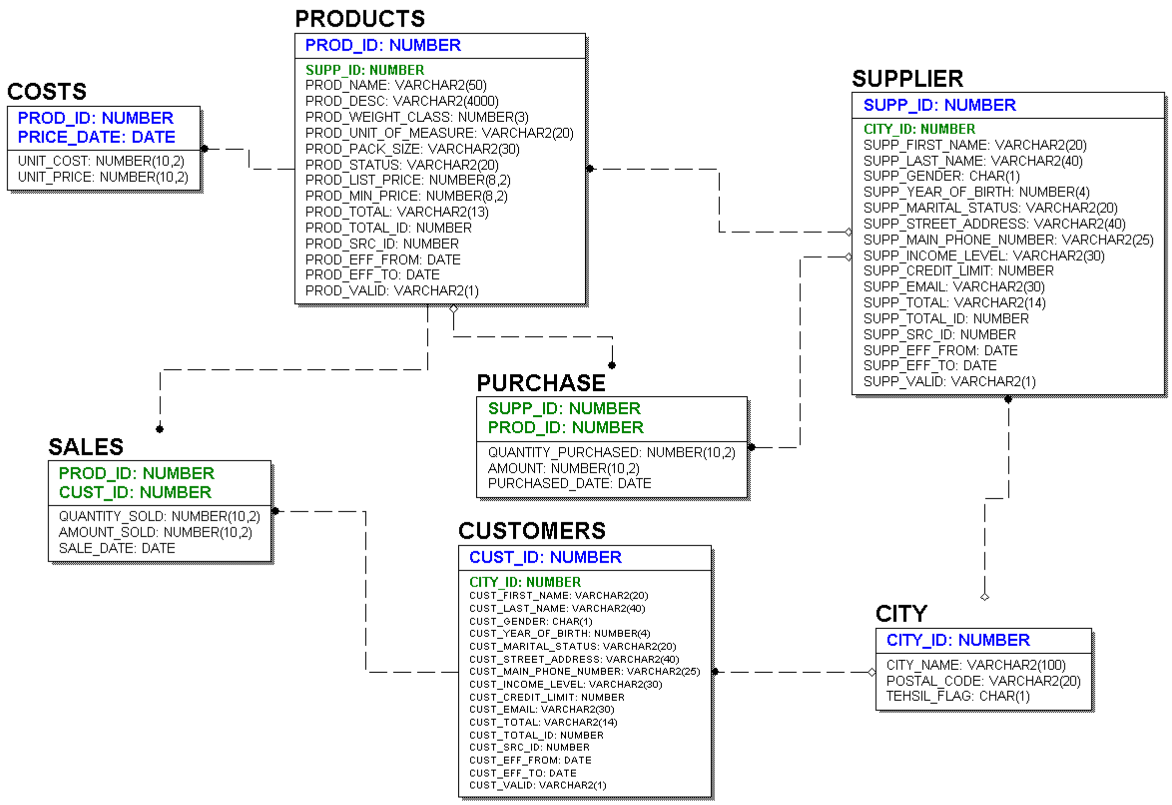


Figure 5-1 Database Design

5.2 Implementing Client Side Application

We develop a client side application on source OLTP database. Users use this application in order to insert data in OLTP source system. We develop this application using Oracle Form Developer 10g.

5.2.1 Design

The design model is the refinement of requirement model with respect to the implementation environment. The most important part of design model is to define all the objects and classes in your system. Class diagrams are very helpful in this respect. Along with the class diagram developer can also refine the interaction diagrams in the designing phase. As this project involves management of distributed database so a database design is also crucial in the design model so that it can easily be deployed in the implementation phase.

In design, different classes are created. These classes represent the functions and attributes to be used in implementation. The objects in object oriented analysis will be the classes for our system. In structure analysis these are the entities of our system. We shell discuss these classes and their functionality in detail here. The class diagram will show the variable and operation of that class and relationship among different classes.

5.2.1.1 Class Diagram

Class diagram is used to represent the different relationship among different classes as shown bellow.

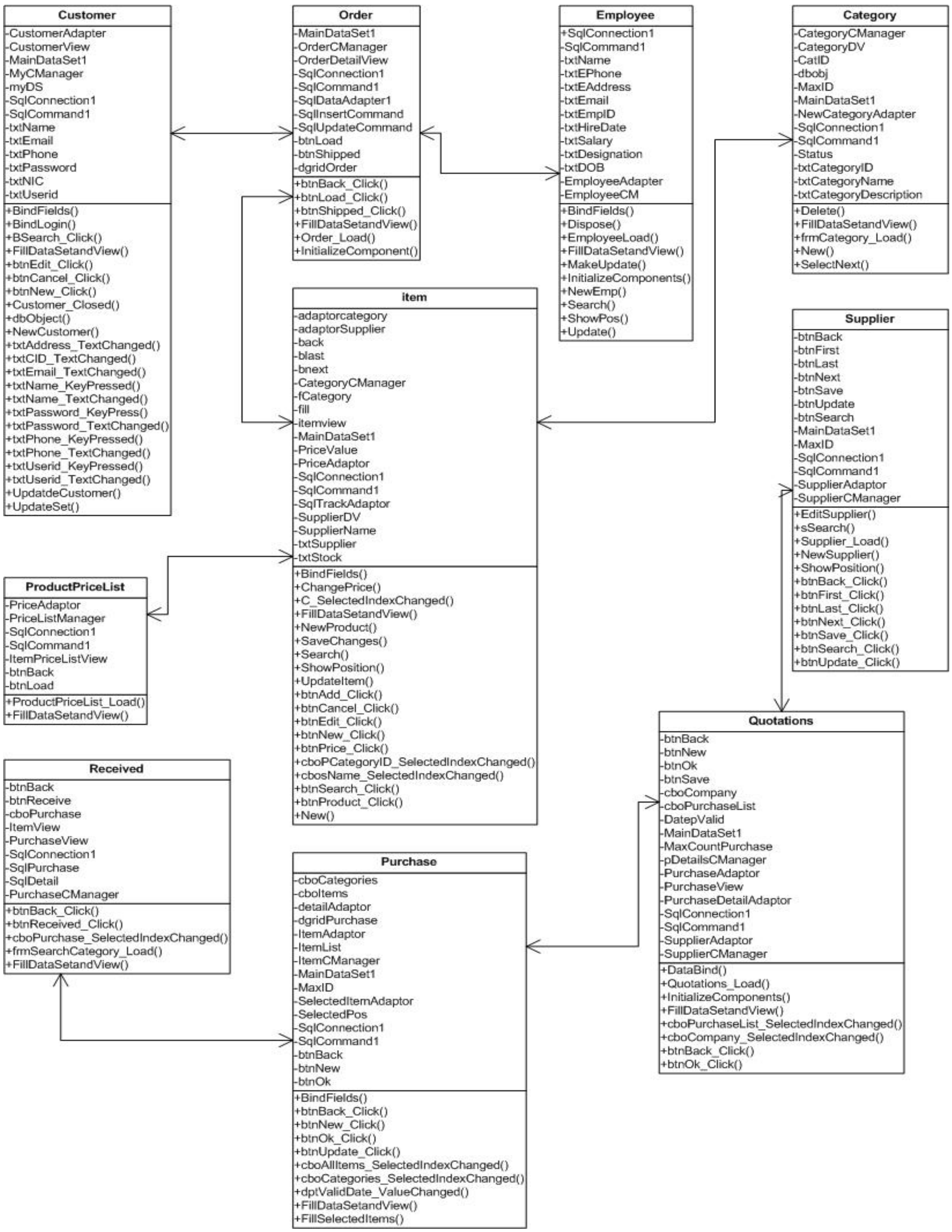


Figure 5-2 Class Diagram

Sequence diagram is the diagram or design artifact, which represents the sequence of activities.

Sequence Diagram for Add Category Use Case

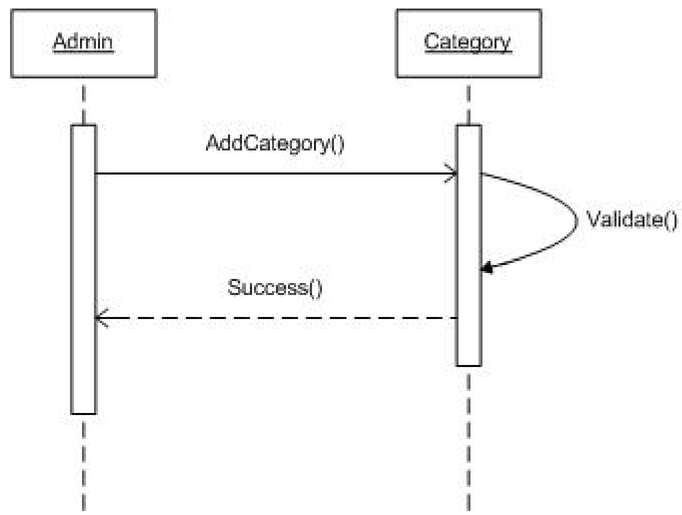


Figure 5-3 Add Category Sequence Diagram

Sequence Diagram for Add Product Use Case

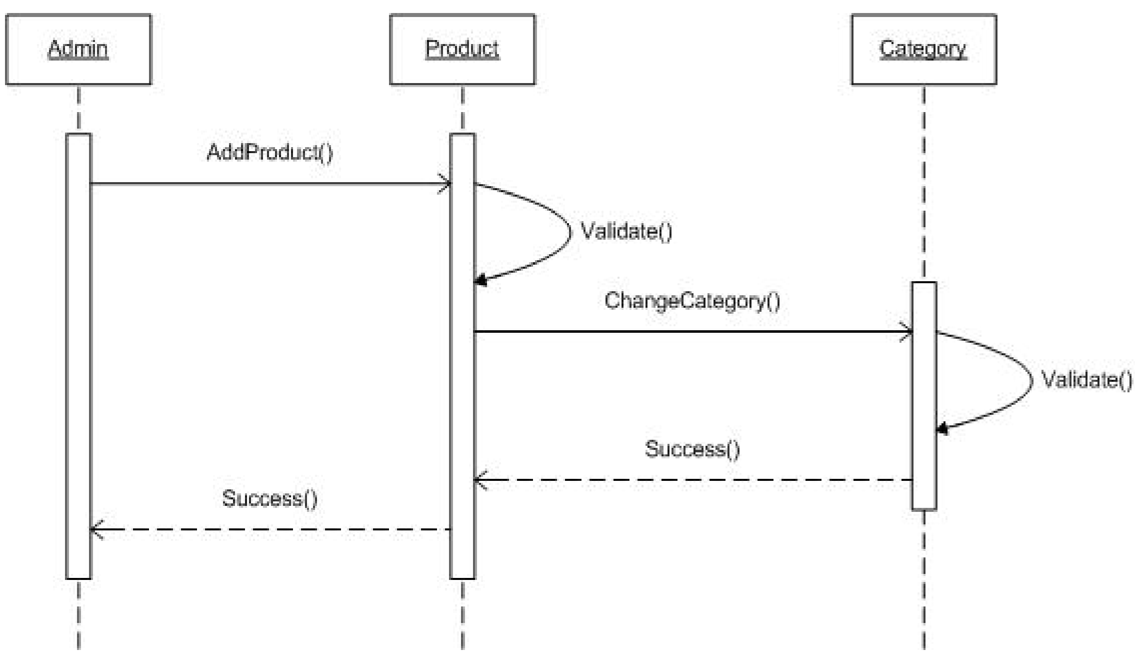


Figure 5-4 Add Product Sequence Diagram

Sequence Diagram for Delete Product Use Case

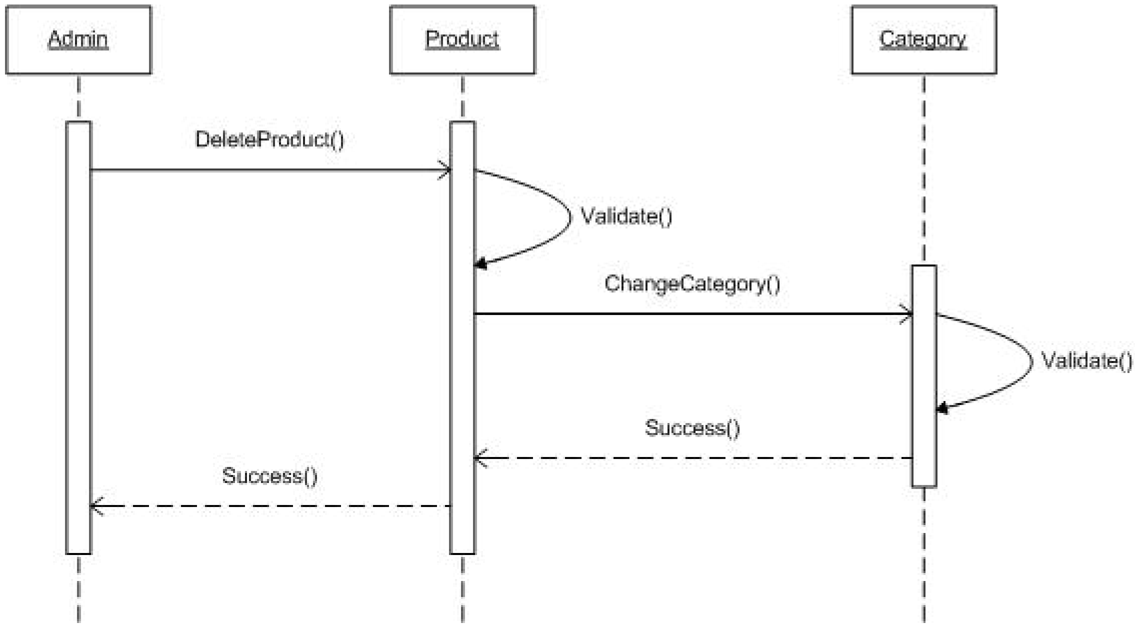


Figure 5-5 Delete Product Sequence Diagram

Sequence Diagram for Update Product Use Case

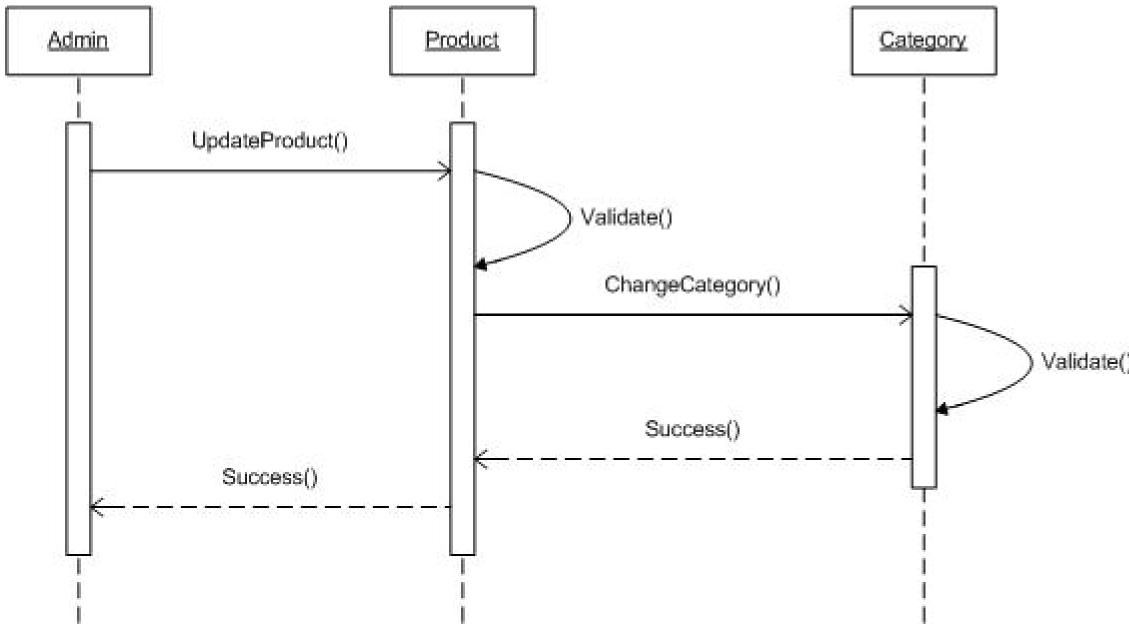


Figure 5-6 Update Product Sequence Diagram

Sequence Diagram for Change Price Use Case

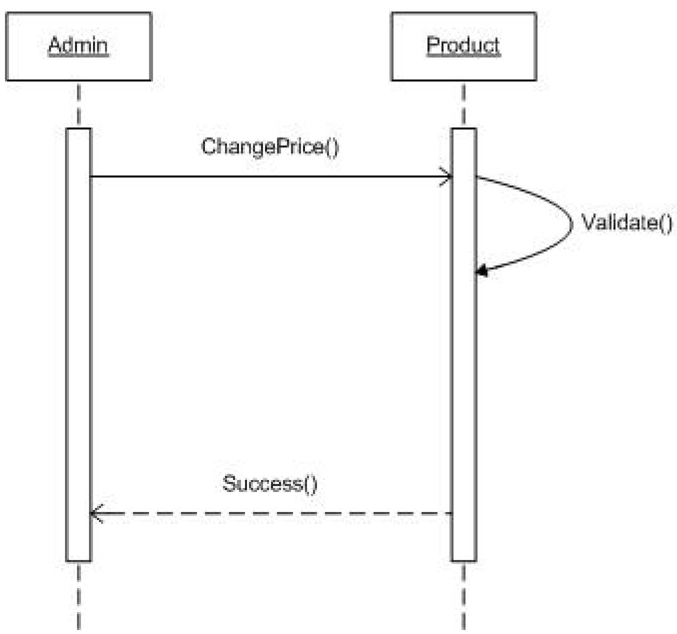


Figure 5-7 Change Price Sequence Diagram

Sequence Diagram for Stock Tracking Use Case

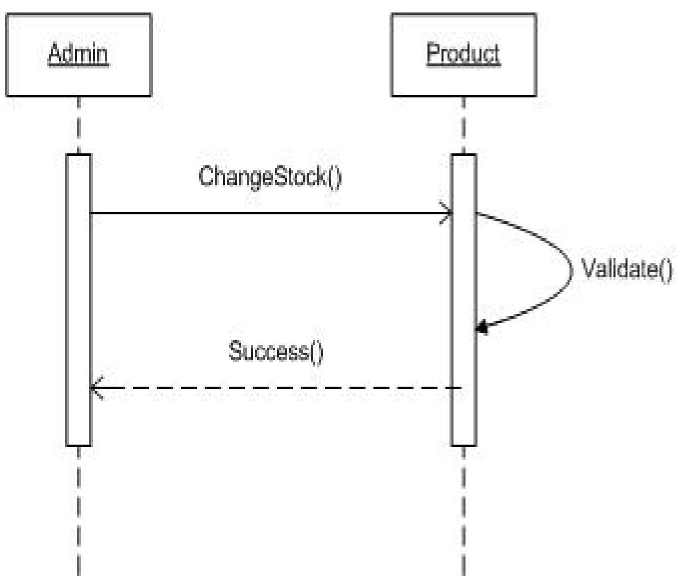


Figure 5-8 Stock Tracking Sequence Diagram

Sequence Diagram for Add Supplier Use Case

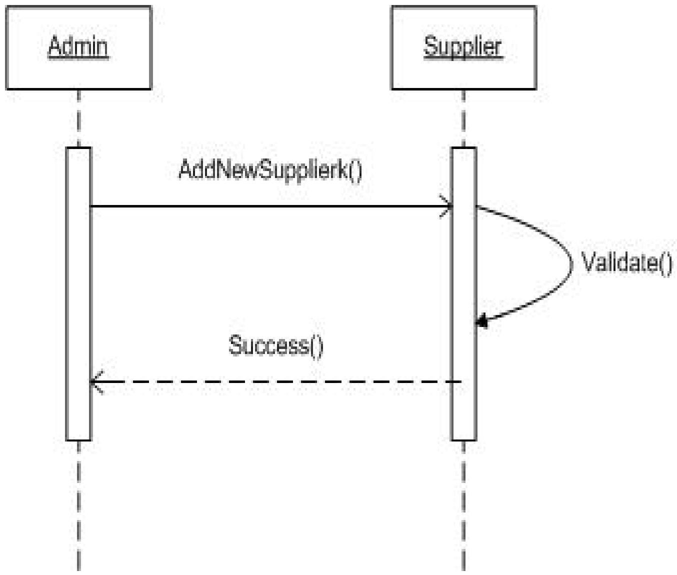


Figure 5-9 Add Supplier Sequence Diagram

Sequence Diagram for Update Supplier Use Case

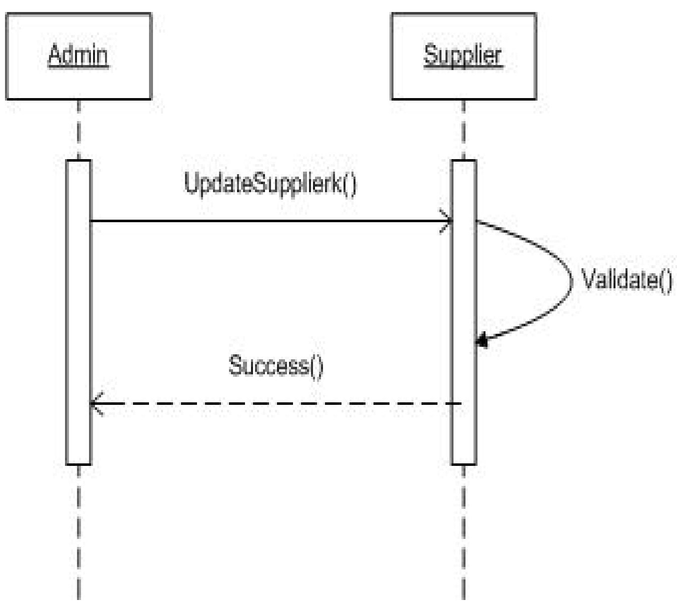


Figure 5-10 Update Supplier Sequence Diagram

Sequence Diagram for Update Category Info Use Case

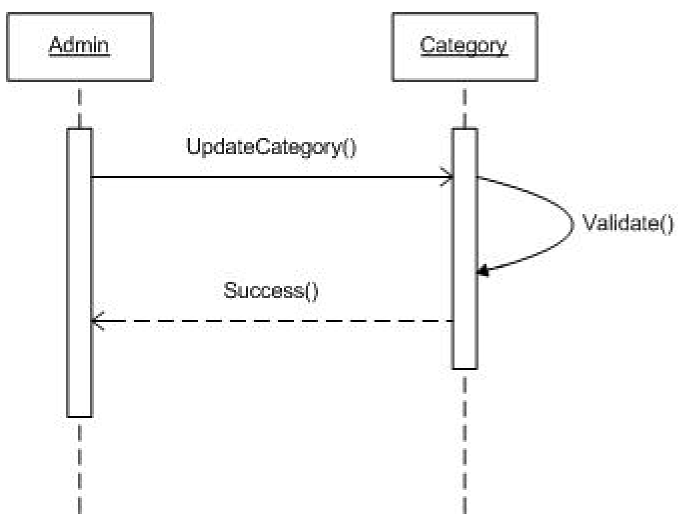


Figure 5-11 Update Category Sequence Diagram

Chapter 6

Results and Discussion

6.1 Introduction

In this chapter, we present the most important findings from our experimental study. A set of criteria have been defined below to evaluate the proposed architecture. These criteria have been identified by reviewing literature and focus on the effectiveness of the architecture.

- *Uniqueness [20]*
  Check whether the architecture is different from the conventional DW architectures.

- *Usefulness [19]*
  Check whether architecture is more useful for decision-making than the conventional DW solutions.

- *Quality of Results [11]*
  Check whether the architecture will improve the quality of results or not. With quality we mean the accuracy of results.

- *Performance [20]*
  Assess tile performance of the architecture against conventional DW solutions.

- *Information Access [11]*
  Check whether the architecture will provide access to more data compared to conventional DW solutions.

6.2 Applying Evaluation Criteria

6.2.1 Uniqueness

The proposed architecture is based on a new concept in the field of data warehousing. It is unique in a sense that different components have been added together to make a new architecture. These components already exist and are optimized but we merge them together to get new functionality.

6.2.2 Usefulness

The use of a data warehouse was focused on top level management of an organization. The proposed architecture will help to make decisions on all levels of an organization. So, the proposed data warehouse is more useful and all decisions can be based on analysis of data. This will improve the overall performance of an organization.

6.2.3 Quality of results

The proposed architecture helps to retrieve more recent data for analysis which ensures and improves the quality (accuracy) of results, and hence the quality of decisions.

6.2.4 Performance

There is a lot of emphasis on performance in data warehouse solutions. The query processing involves some extra processes like decomposition, conversion and merging of results. The proposed architecture involves three extra processes when both OLTP data and OLAP data is needed. We therefore conducted some performance testing to evaluate performance of the proposed architecture. A detail performance analysis is done in section 6.3.

6.2.5 Information Access

The information access is increased as fresh data in OLTP systems is also available as well as historical data in OLAP systems for analysis based decision making. Whenever flesh data is needed, it will be extracted "on the fly" from the OLTP systems.

6.3 Performance Analysis of Purposed Architecture

6.3.1 Implementation Tools

The implementation of conventional data warehouse architecture is done by using different sets of technologies. Oracle 10g is used for OLTP database and Oracle 10g warehouse builder is used for data warehouse.

6.3.2 OLTP Data Set

Table 6-1 contains OLTP Tables and no of records.

Table Name	No. of Records
Purchase	14,342
Sales	13,721
Costs	113
Customers	65,501
Suppliers	5,503
City	50,884
District	134
Products	73
Product Type	23
Product Category	5

Table 6-1 OLTP Data Set

6.3.3 OLAP Data Set

Table Name	No. of Records
Fact_Purchase	918,845
Fact_Sales	918,846
Dim_Costs	82,113
Dim_Customers	55,501
Dim_Suppliers	55,503
Dim_City	50,884
Dim_Time	1826
Dim_Products	73

Table 6-2 OLAP Data Set

6.3.4 Distinct Count Analysis

Distinct Count Analysis is one of the most popular types of analyses, and yet, it is also one of the toughest problems for an OLAP system. The toughest problems refer to the problem as the many-to-many problem because analysis of the relationship between entities that have many-to-many relationships is involved.

The following represents typical usage for Distinct Count analysis.

- Sales and marketing: counting distinct numbers of customers and numbers of Sales Representatives, and so on.

6.3.5 Defining Business Questions

Table 6-3 lists questions that involve Distinct Counts. These questions are representative of typical questions a Sales and Marketing analyst may need to answer. These questions served as a starting point for the Distinct Counts study.

Query No.	Question
1	How many distinct customers bought product X in year 2010?
2	How many distinct products in category X were sold in each of the regions in last year?
3	How many distinct sales people sold product X in particular months?
4	How many distinct customers shopped at various stores in a particular quarter?
5	What is the average sales amount per distinct customer for various years?

Table 6-3 Business Questions

6.3.6 Query Performance

All queries used for this performance testing involved Distinct Counts.

6.3.6.1 Testing scenario

- We assume a situation when last loading date is 30-Dec-2010 and report needs data till end of year 2010.
- Sales cube contains data till 30-Dec-2009, as last data loading date is 30-Dec-2010.
- For the month of December 30 day's data needed from OLAP and 31st day needed from OLTP.
- Average Query Response Time (ART) is calculated for 10, 20, 30, 40 and 50 Queries.
- ART is calculated for cold and hot cache.
- For cold cache measurements systems were rebooted.
- Query results were held for hot cache measurement. This is achieved by immediately executing the same test after cold cache test.

The following section shows query response results. Average query response times were measured for various concurrent users for cold and warm cache.

6.3.6.2 Average and Median QRT (Cold Cache/Up to 50 Concurrent Users)

For the cold cache measurement, no query results were in cache; the OLAP Services server was restarted prior to the cold cache tests.

For this test, all queries were going against the Fact_Sales cube for OLAP and Sales table in OLTP database. The figure 6-1 graph shows the average query response time up to 50 queries.

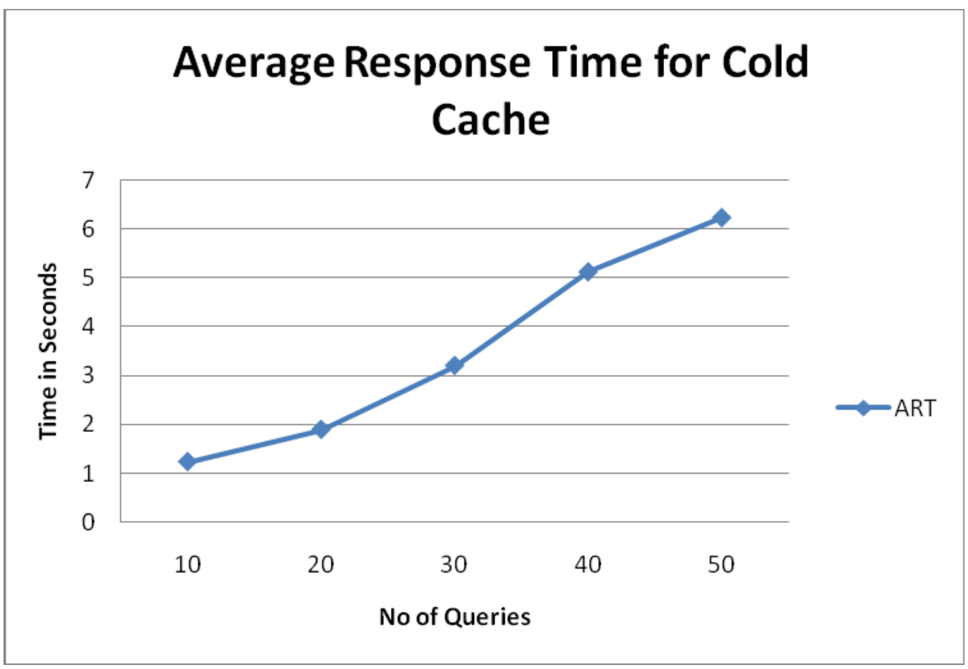


Figure 6-1 ART for cold cache

The data in figure 6-1 shows that:

- Average response time (ART) of 1.89 seconds was observed for 10 queries requiring data from both OLAP and OLTP.
- Average response time of 5.21 seconds was observed when 30 queries were executed on both data. Considering the fact that all the queries were fairly complex and all of them involved Distinct Counts, this time is reasonable.
- Average response time of 7.12 seconds was observed when 50 queries were executed.

6.3.6.3 Median vs. Average Query Response times

In statistics, the median represents the middle value in a group of measurements. It is a commonly used indicator of what measurement is typical or normal for a group. For various concurrent users, look at the median response times, and then observe that for 50 queries, median response time is only 2.1 seconds as compared to an average response time of 7.12 seconds.

Although the average is used more frequently than the median, the median is still an important measure of central tendency. Median is not affected by the presence of a number that is extremely high or extremely low relative to the other numbers in the group. In other words, median is not skewed as a result of some 20 percent or so queries sent by users that really take a long time to return results while 80 percent of the queries could be taking only couple of seconds to return results. Sometimes median response time makes more sense than average response times.

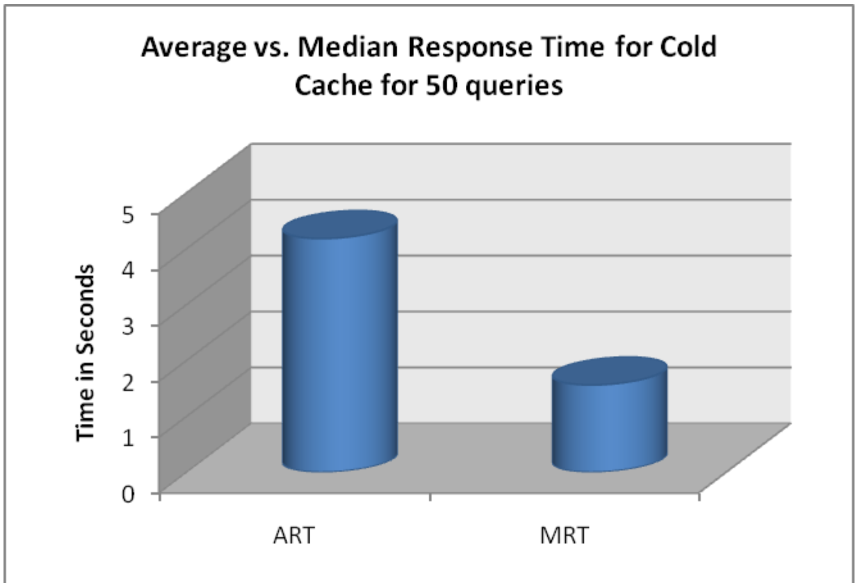


Figure 6-2 MRT vs. ART for cold cache on 50 queries

6.3.6.4 Average and Median QRT (Hot Cache/Up to 50 Concurrent Queries)

For the warm cache measurement, query results were held in cache. Executing the cold cache test and then immediately executing the same test again resulted in a warm cache-testing scenario.

In this test, all queries were executed against the Fact_Sales cube. The cube that was used was not optimized for performance. Queries need to get 1 day data from OLTP.

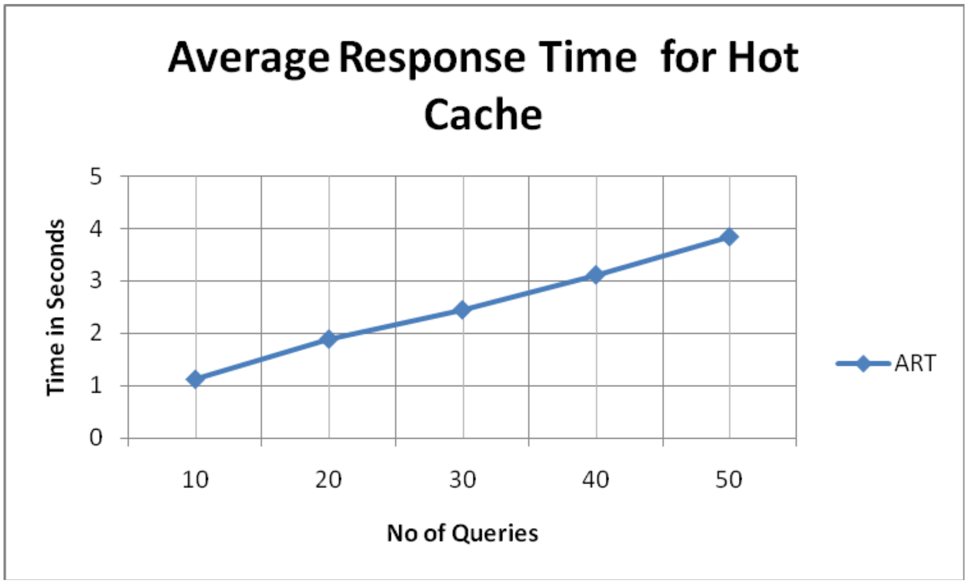


Figure 6-3 ART of Hot cache for 50 queries

The data in figure 6-3 shows that:

- Average response time (ART) of 3.85 seconds was observed for 50 queries users were executed, half of the time same users querying with cold cache. Although it seems obvious to get good performance when cache is warm, this performance is commendable, especially with 50 concurrent users querying the cube with 15 percent data from OLTP.

6.3.6.5 Comparison Average QRT (Cold vs. Warm Cache /Up to 50 Queries)

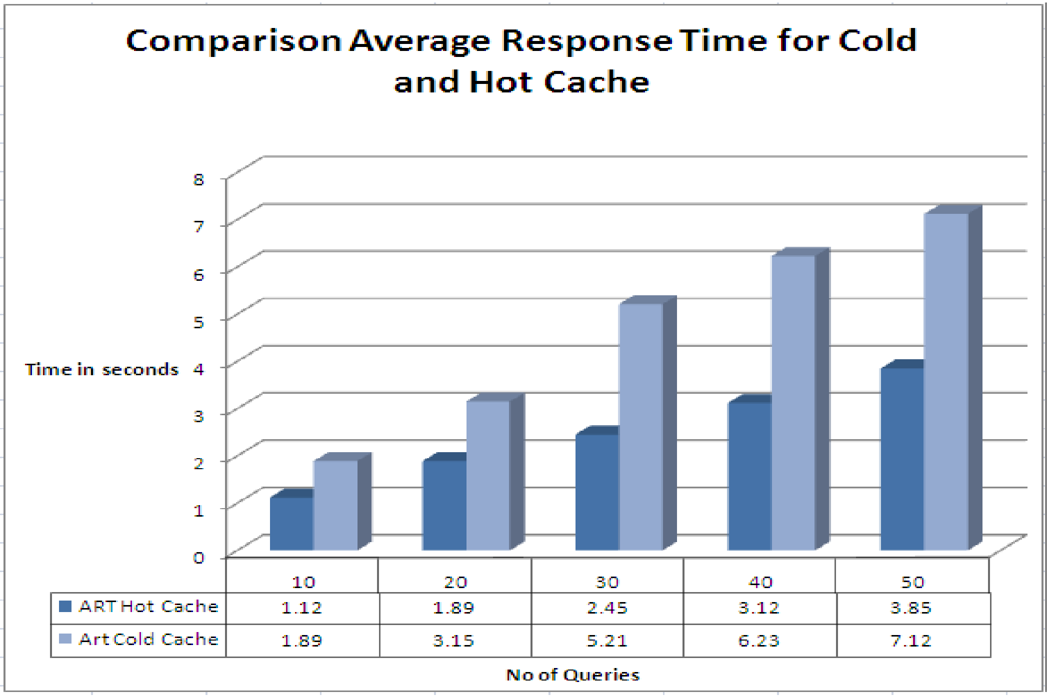


Figure 6-4 Comparison ART for cold and hot cache

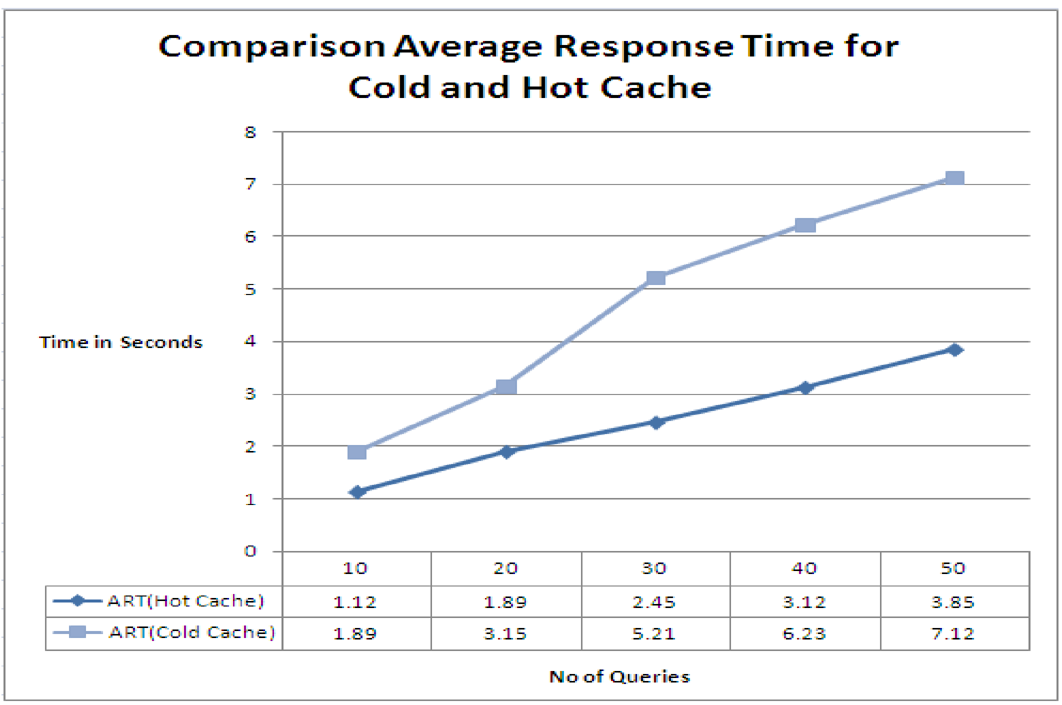


Figure 6-5 Comparison ART for Cold and Hot Cache

The data in figure 6-4 shows that:

- For cold cache, Average Response Time increased from 1.89 seconds to 7.12 seconds when the number of queries was increased from 10 to 50. For warm cache, ART of 1.12 to 3.85 seconds was observed when the number of concurrent users querying the OLAP cube and OLTP increased from 10 to 50..
- The figure 6-5 shows that once the cache was warm, the results were less than a half of that cold cache. You can execute the most often used queries as a batch job immediately after finishing the cube processing. Using this method will result in improved query performance for business users in the production environment.

6.3.6.6 Average QRT (All data from OLAP up to 50 Queries)

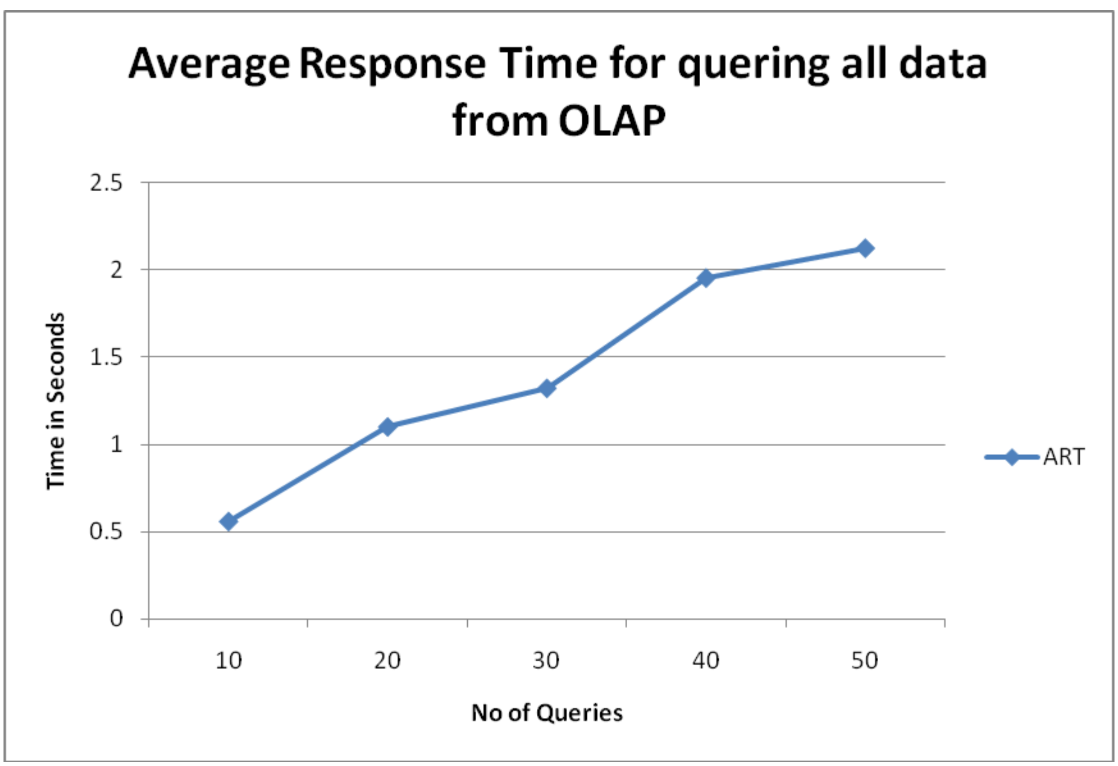


Figure 6-6 ART for querying all data from OLAP

The data in figure 6-6 shows that:

- Query response time to query all data from OLAP without involving OLTP data.
- All the results show that QRT is less than hot cache results from proposed architecture.

6.3.6.7 Comparison Average QRT (Cold vs. Warm Cache vs. OLAP for up to 50 queries)

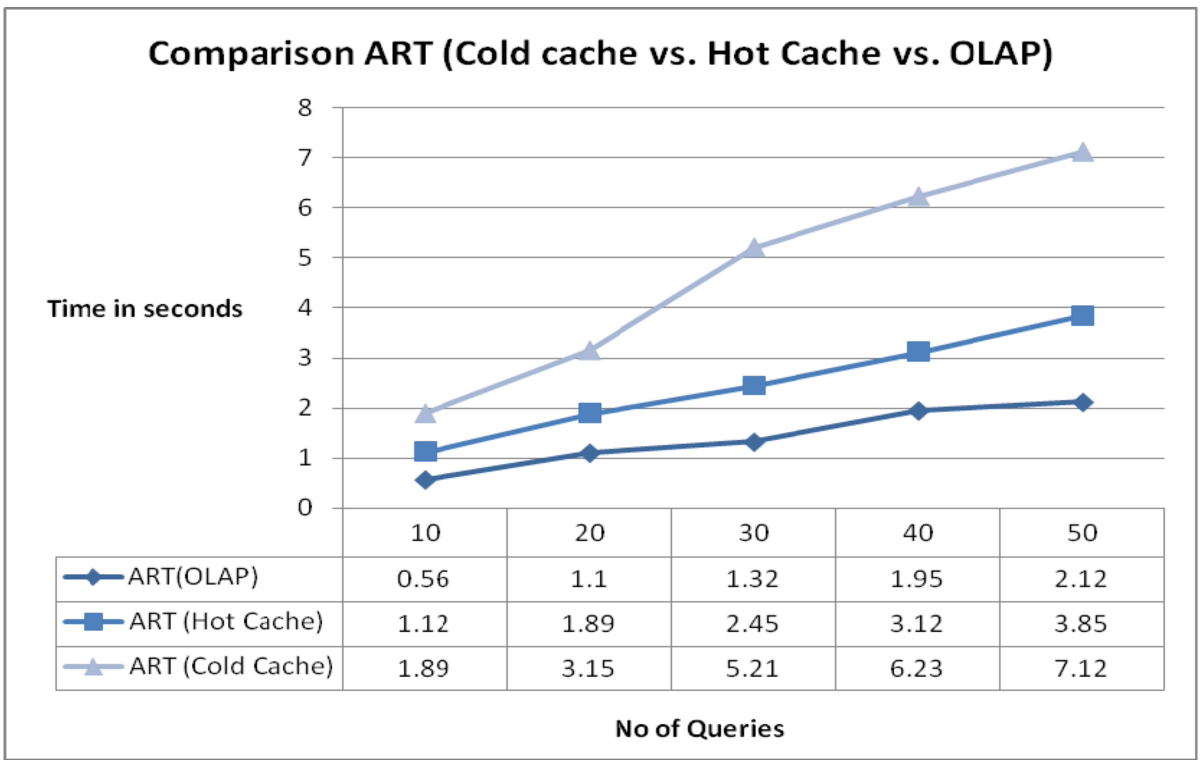


Figure 6-7 ART (cold Cache vs. Hot cache vs. OLAP)

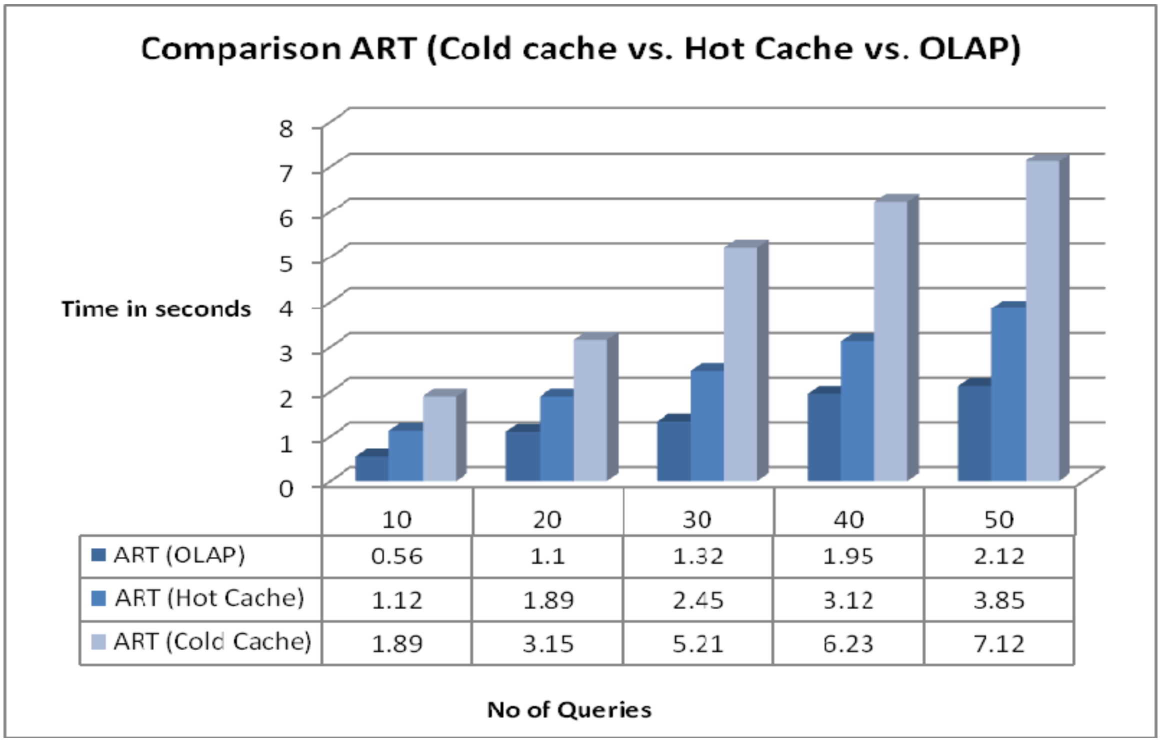


Figure 6-8 Comparison (Cold Cache vs. Hot Cache vs. OLAP)

The figure 6-8 shows that

- Average response time for hot cache is slightly higher than same data query from OLAP.
- Significant difference when comparing Cold Cache results with OLAP.

6.4 Why proposed architecture better than OLAP

The analysis in section 6.3 shows that Query average response time for the proposed architecture is slightly higer than against conventional DW architecure. We further analyze our architecture against coventional architecure by including ETL process time. We analyze our architecure invloving real time ETL process and routine ETL process.

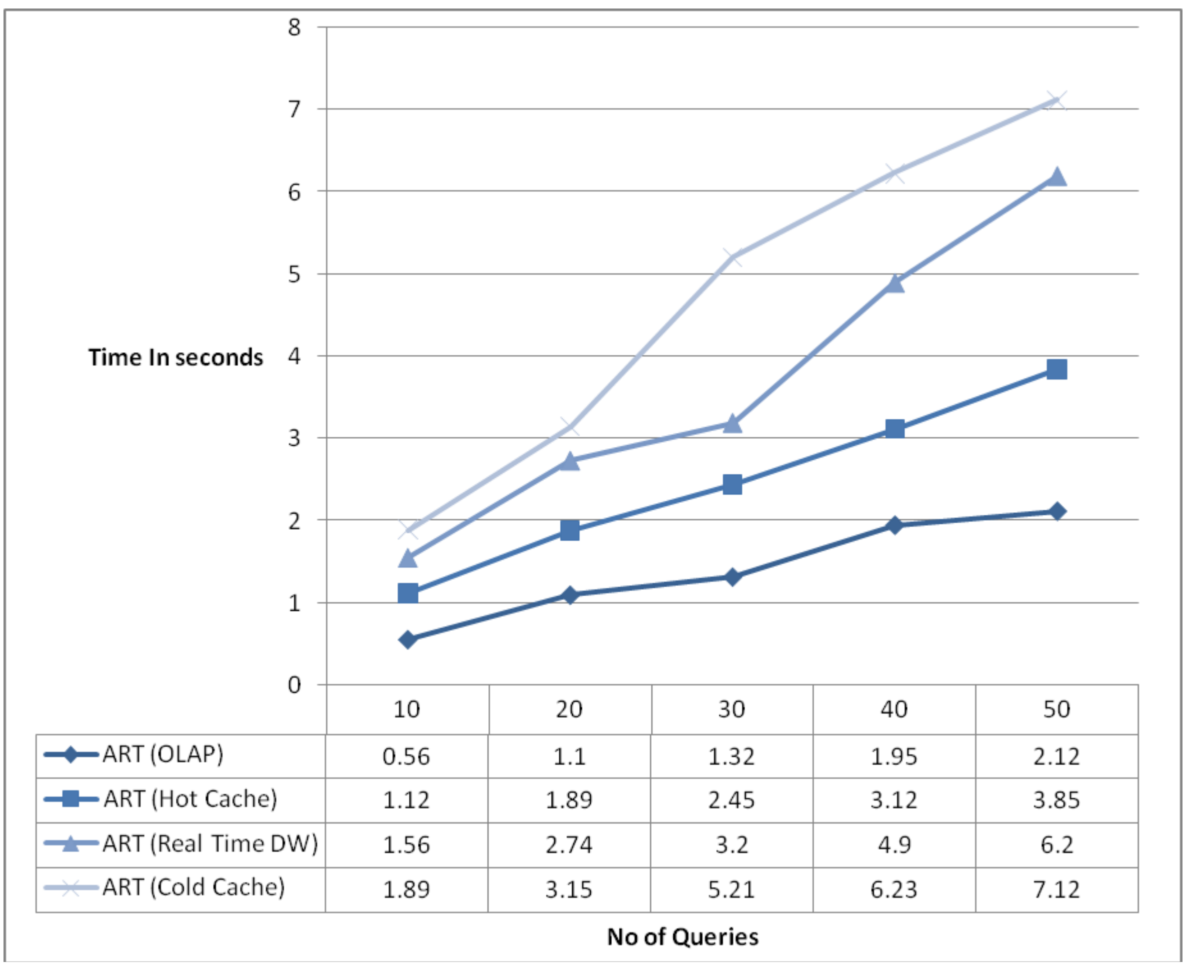
6.4.1 Proposed Architecure Vs. Conventional invloving Routine ETL process

Normal ETL processes are done on daily or weekly process. If we take daily process than conventional architecture will give results after 1 day. That provides a significant advantage in using our proposed architecture which will give results at the hand of some milliseconds not a whole day.

6.4.2 Proposed Architecure Vs. Conventional invloving Real time ETL process

We further analyze our architecure using real time ETL process. The following graph includes ETL process when quering all data from OLAP.

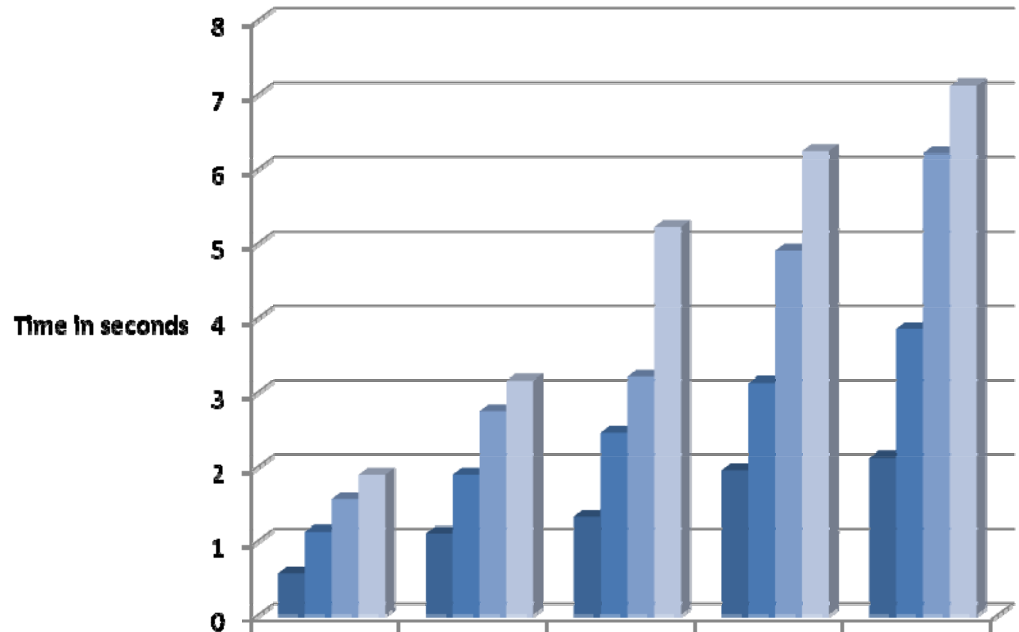
Comparison of all Techniques



	10	20	30	40	50
ART (OLAP)	0.56	1.1	1.32	1.95	2.12
ART (Hot Cache)	1.12	1.89	2.45	3.12	3.85
ART (Real Time DW)	1.56	2.74	3.2	4.9	6.2
ART (Cold Cache)	1.89	3.15	5.21	6.23	7.12

Figure 6-9 ART Proposed Architecture vs. OLAP with real time ETL

Comparison of all Techniques



	10	20	30	40	50
ART (OLAP)	0.56	1.1	1.32	1.95	2.12
ART (Hot Cache)	1.12	1.89	2.45	3.12	3.85
ART (Real Time DWH)	1.56	2.74	3.2	4.9	6.2
ART (Cold Cache)	1.89	3.15	5.21	6.23	7.12

No of Queries

Figure 6-10 Comparison of all techniques

The figure 6-10 shows that :

- Proposed architecure provides better results with hot cache when quering same data from OLAP invloving real time ETL process.
- Proposed architecure provides results with cold cache is slighlty higher when quering same data from OLAP invloving real time ETL process.

The table 6.4 explains the comparison of the proposed approach with the OLAP, indicating the average of OLAP, proposed approch average RTDW.

No of Queries	Cycle 1			Cycle 2			Cycle 3			Cycle 4			Cycle 5			Average		
	OLAP	Proposed Approach	RTDW	OLAP	Proposed Approach	RTDW	OLAP	Proposed Approach	RTDW	OLAP	Proposed Approach	RTDW	OLAP	Proposed Approach	RTDW	OLAP	Proposed Approach	RTDW
10	0.56	1.13	1.57	0.56	1.13	1.57	0.57	1.13	1.57	0.57	1.13	1.57	0.56	1.13	1.57	0.56	1.13	1.57
20	1.10	1.90	2.76	1.11	1.11	2.80	1.11	1.90	2.86	1.11	1.90	2.76	1.15	1.90	2.76	1.12	1.75	2.79
30	1.32	2.45	3.19	1.32	2.45	3.19	1.32	2.45	3.19	1.32	2.46	3.19	1.32	2.45	3.19	1.32	2.45	3.19
40	1.95	3.12	4.89	1.95	3.12	4.89	1.95	3.12	4.89	1.95	3.12	4.91	1.95	3.12	4.89	1.95	3.12	4.89
50	2.16	3.85	6.20	2.06	3.85	6.23	2.12	3.85	6.25	2.12	3.85	6.20	2.16	3.78	6.20	2.13	3.84	6.22

Table 6-4 Comparison of proposed appraoch with other Techniques

6.4.3 Comparison of Proposed Approach with RTDW (Increasing No of Updates)

The figure 6-11 shows that as the number of updates increase in a given time interval, the ART of RTWD increases. As the number of updates increases, more time consumed in insertion of data into data warehouse.

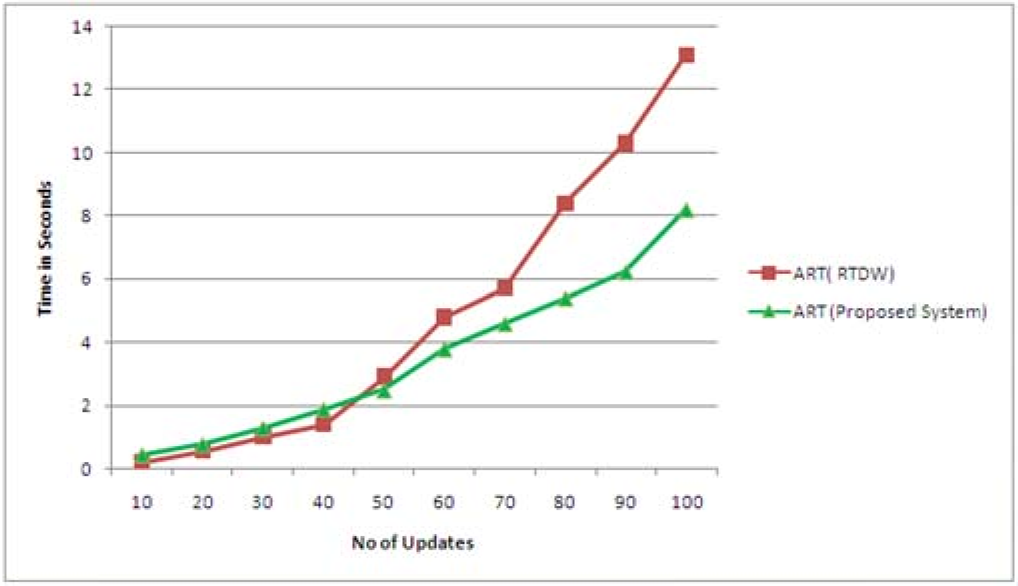
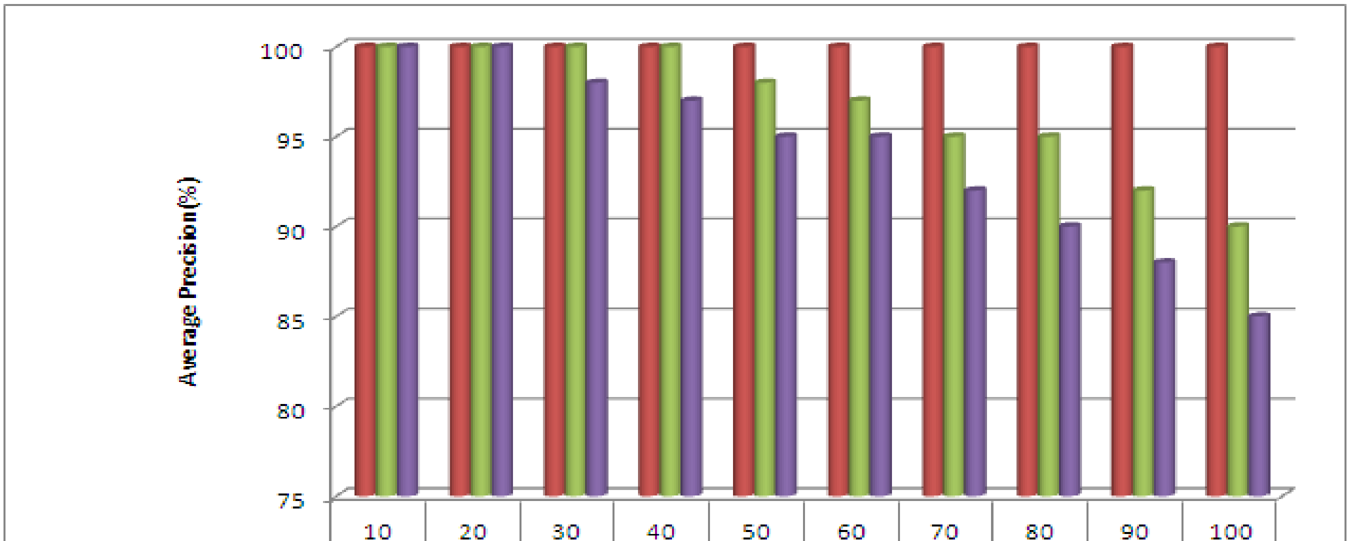


Figure 6-11Comparison (Increasing no of Updates)

6.4.4 Precision:

In an Information Retrieval scenario, Precision is defined as the number of relevant records retrieved out of the total number of records retrieved by the search. Precision can be seen as a measure of exactness.

This is the fraction of the returned results that are relevant to the information.



	10	20	30	40	50	60	70	80	90	100
All data in OLAP	100	100	100	100	100	100	100	100	100	100
RTDW	100	100	100	100	98	97	95	95	92	90
Proposed Approach	100	100	98	97	95	95	92	90	88	85

Figure 6-12 Precision Comparison Graph

6.5 Conclusion

Generally data warehouse solutions help top level management decision makers. They require fresh as well as historical data for analysis and lower level decision making.

Our proposed architecture will enable management in an organization to make strategic decisions and also to keep an eye on the progress of different tasks running in an organization at real time. Managers can take appropriate steps immediately if any task is diagnosed to be not going in accordance with strategic decisions taken by top level management in an organization.

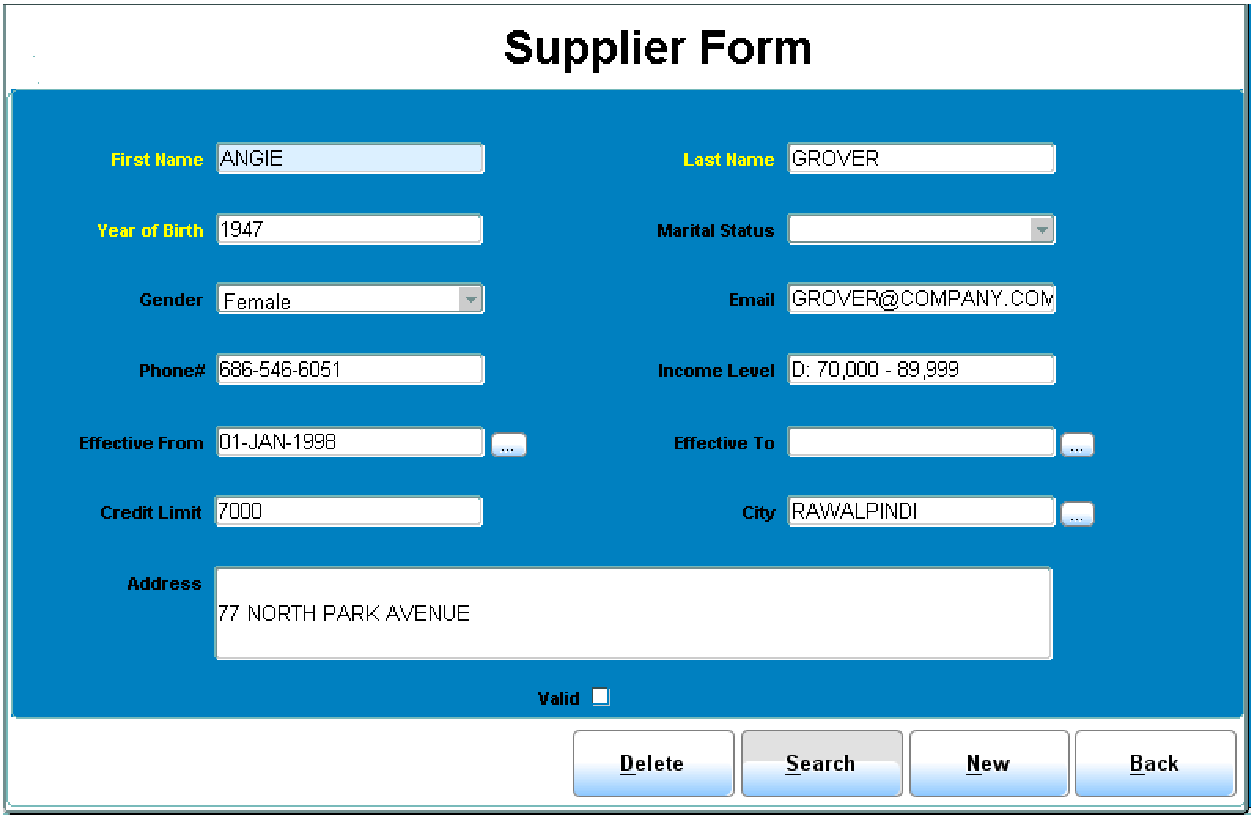
All the components of proposed architecture exists in different areas but we assemble them to get new functionality of accessing real time or quickly changing data for lower level management. Already tested and optimized components produces results of greater quality and accuracy, thus our solution provides more accurate, trusted and good quality results than conventional data warehouse. There is small decrease in performance of querying the results with cold cache but provides much better performance than conventional DW when querying the data using hot cache.

6.6 Future Work

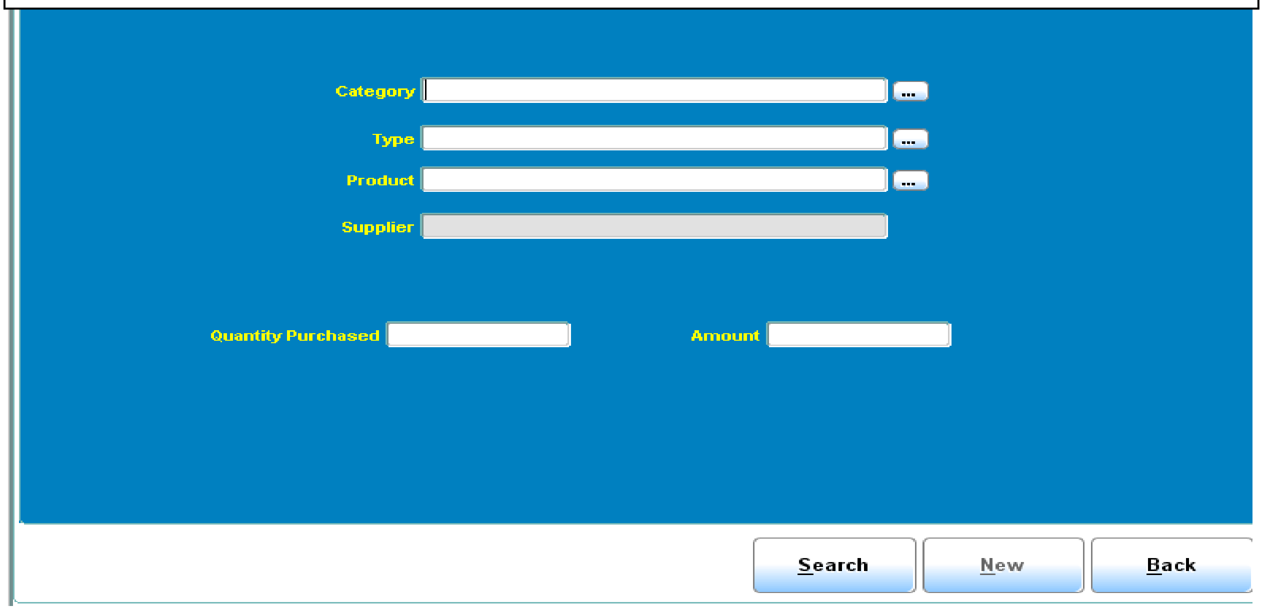
The developed architecture is a new concept and a lot to be done to emend its different areas. One of the important issues in decision making is performance; therefore, future work involves the improvement and betterment in query optimization techniques. The developed architecture will become worthier if fresher data are to be integrated from heterogeneous external sources including different OLTP systems and other Data warehouses etc.

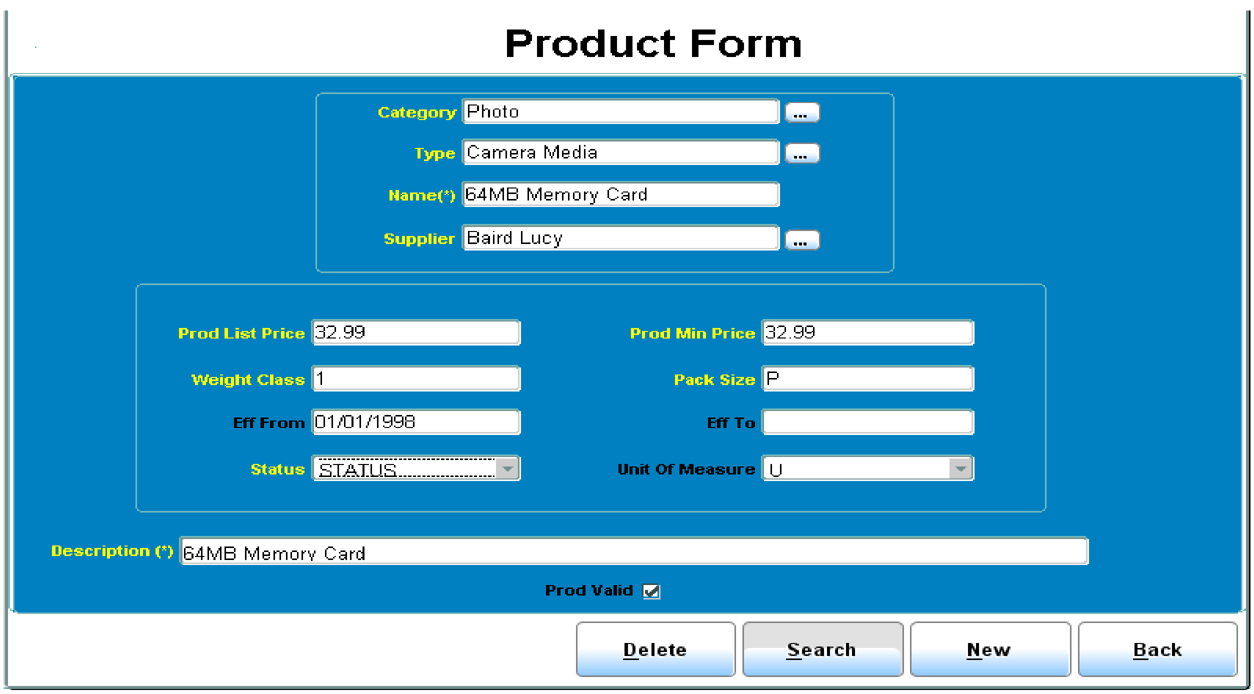
Appendix

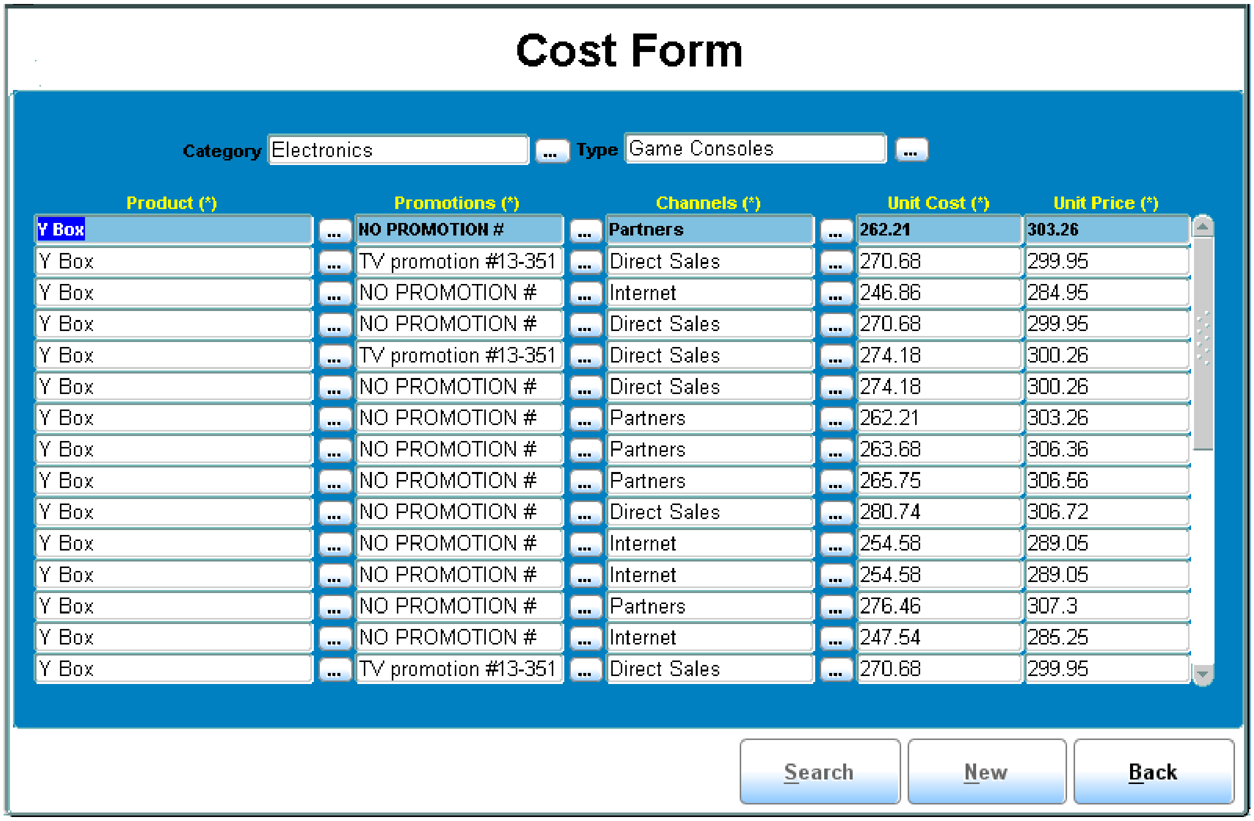
Appendix



Purchase Form





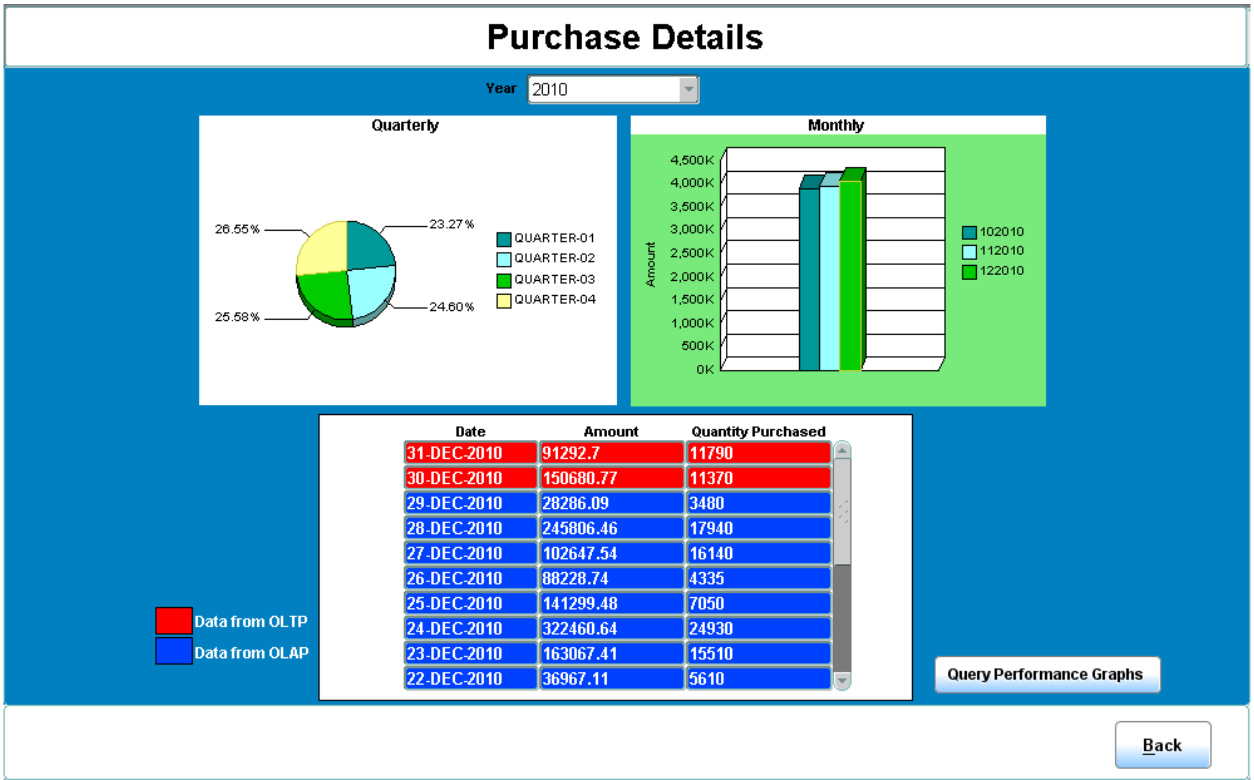


Cost Form

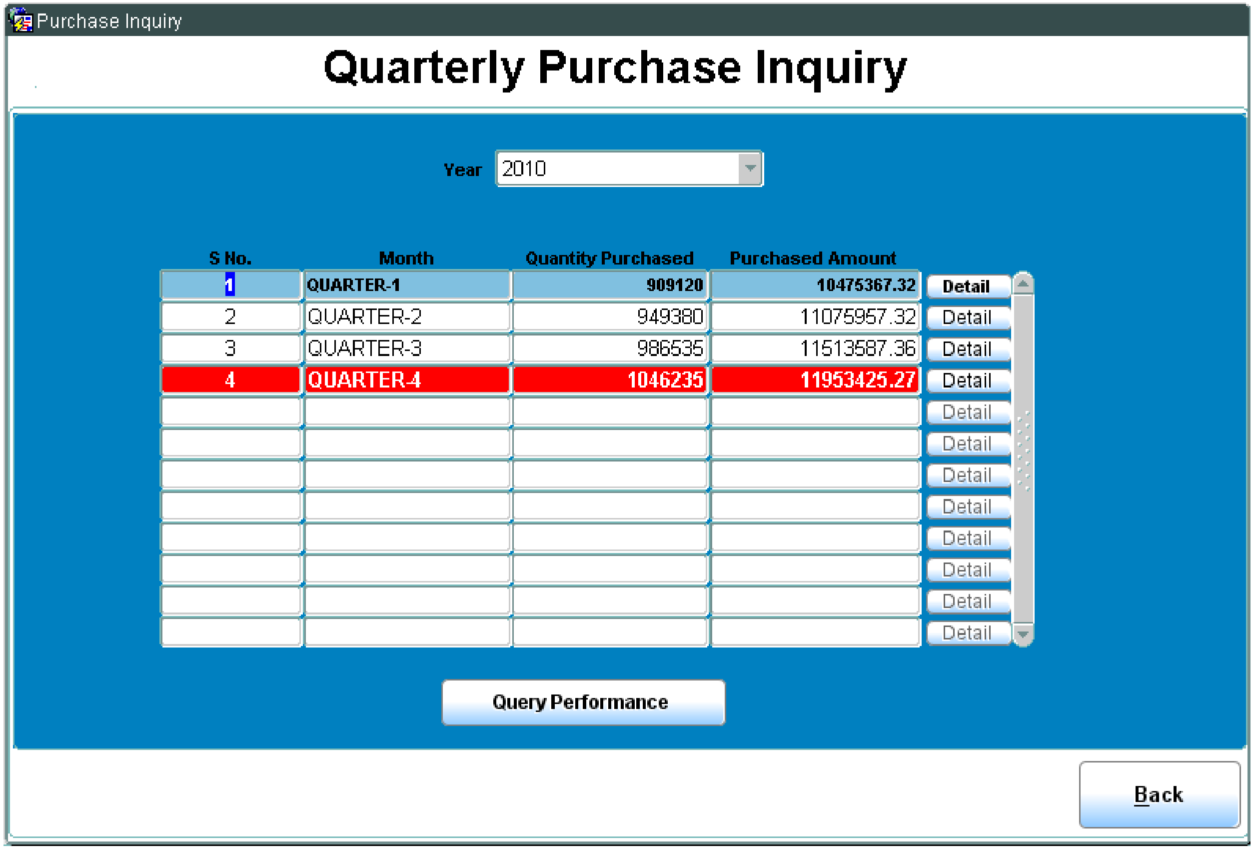
Category Electronics ... Type Game Consoles ...

Product (*)		Promotions (*)		Channels (*)		Unit Cost (*)	Unit Price (*)
Y Box	...	NO PROMOTION #	...	Partners	...	262.21	303.26
Y Box	...	TV promotion #13-351	...	Direct Sales	...	270.68	299.95
Y Box	...	NO PROMOTION #	...	Internet	...	246.86	284.95
Y Box	...	NO PROMOTION #	...	Direct Sales	...	270.68	299.95
Y Box	...	TV promotion #13-351	...	Direct Sales	...	274.18	300.26
Y Box	...	NO PROMOTION #	...	Direct Sales	...	274.18	300.26
Y Box	...	NO PROMOTION #	...	Partners	...	262.21	303.26
Y Box	...	NO PROMOTION #	...	Partners	...	263.68	306.36
Y Box	...	NO PROMOTION #	...	Partners	...	265.75	306.56
Y Box	...	NO PROMOTION #	...	Direct Sales	...	280.74	306.72
Y Box	...	NO PROMOTION #	...	Internet	...	254.58	289.05
Y Box	...	NO PROMOTION #	...	Internet	...	254.58	289.05
Y Box	...	NO PROMOTION #	...	Partners	...	276.46	307.3
Y Box	...	NO PROMOTION #	...	Internet	...	247.54	285.25
Y Box	...	TV promotion #13-351	...	Direct Sales	...	270.68	299.95

Search New Back



Purchase Details
Year
2010
Quarterly
26.55%
23.27%
25.58%
24.60%
QUARTER-01
QUARTER-02
QUARTER-03
QUARTER-04
Monthly
Amount
4,500K
4,000K
3,500K
3,000K
2,500K
2,000K
1,500K
1,000K
500K
0K
102010
112010
122010
Date
Amount
Quantity Purchased
31-DEC-2010 91292.7 11790
30-DEC-2010 150680.77 11370
29-DEC-2010 28286.09 3480
28-DEC-2010 245806.46 17940
27-DEC-2010 102647.54 16140
26-DEC-2010 88228.74 4335
25-DEC-2010 141299.48 7050
24-DEC-2010 322460.64 24930
23-DEC-2010 163067.41 15510
22-DEC-2010 36967.11 5610
Data from OLTP
Data from OLAP
Query Performance Graphs
Back



Purchase Inquiry

Quarterly Purchase Inquiry

Year 2010

S No.	Month	Quantity Purchased	Purchased Amount	
1	QUARTER-1	909120	10475367.32	Detail
2	QUARTER-2	949380	11075957.32	Detail
3	QUARTER-3	986535	11513587.36	Detail
4	QUARTER-4	1046235	11953425.27	Detail

Query Performance

Back

Query Performance

OLTP Query

	Count	CPU	Elapsed	Query
Parse	1	0.000	0.000	0.000
Execute	1	0.000	0.000	0.000
Fetch	2	0.531	0.532	0.004
Total	4	0.531	0.532	0.004

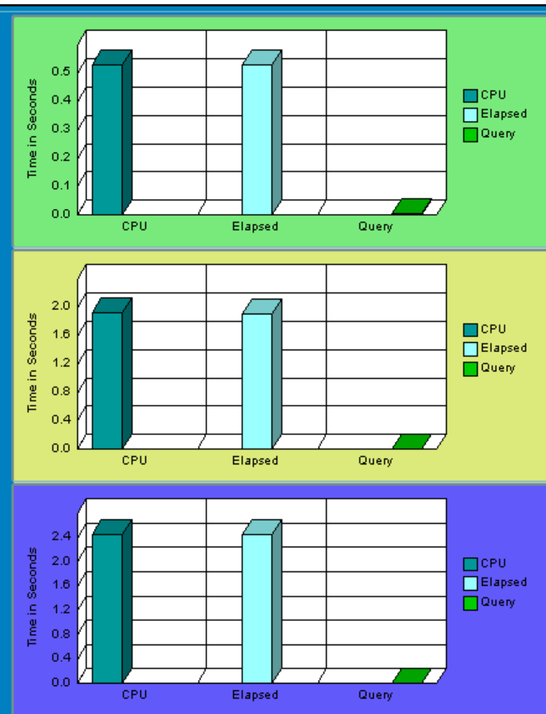
OLAP Query

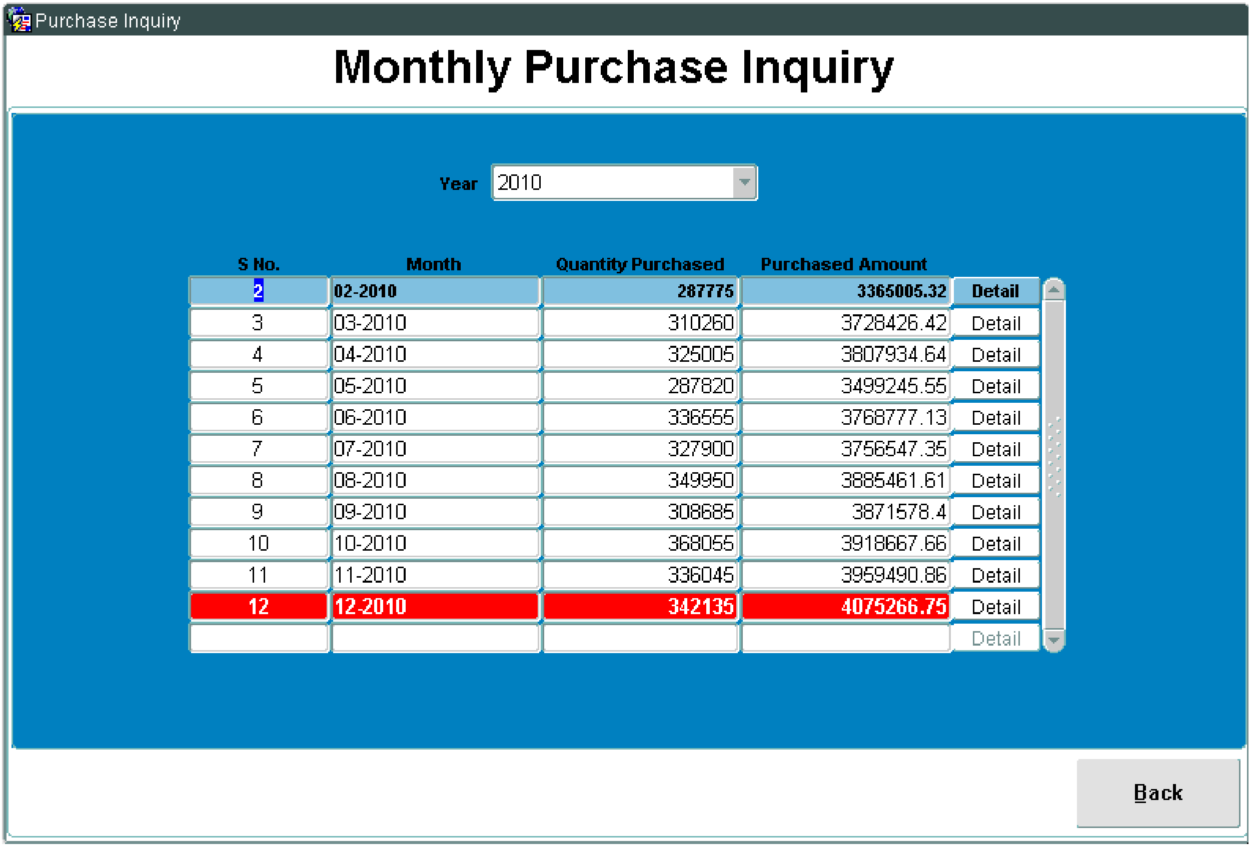
	Count	CPU	Elapsed	Query
Parse	1	0.000	0.000	0.000
Execute	1	0.000	0.000	0.000
Fetch	13	1.922	1.916	0.004
Total	15	1.922	1.916	0.004

Total

	Count	CPU	Elapsed	Query
Parse	2	0.000	0.000	0.000
Execute	2	0.000	0.000	0.000
Fetch	15	2.453	2.448	0.008
Total	19	2.453	2.448	0.008

* Mandatory Fields



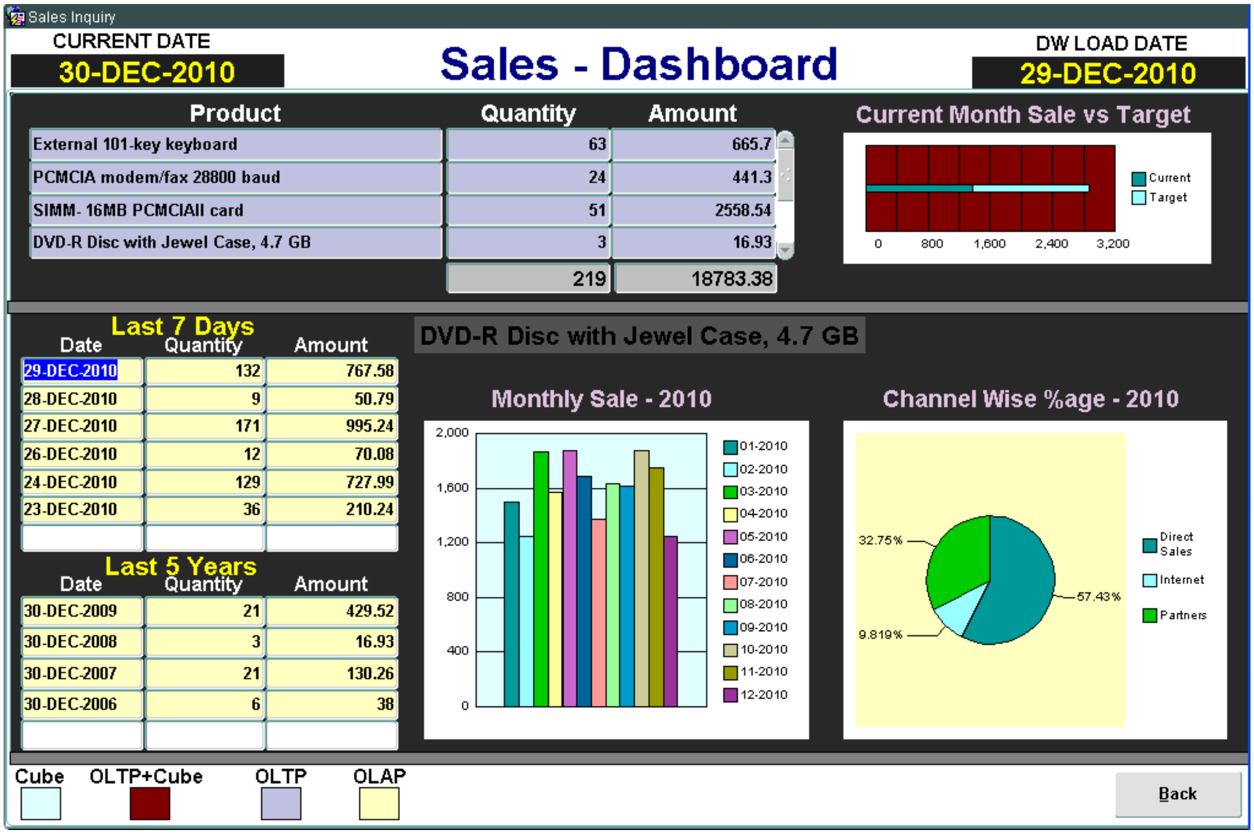

Purchase Inquiry

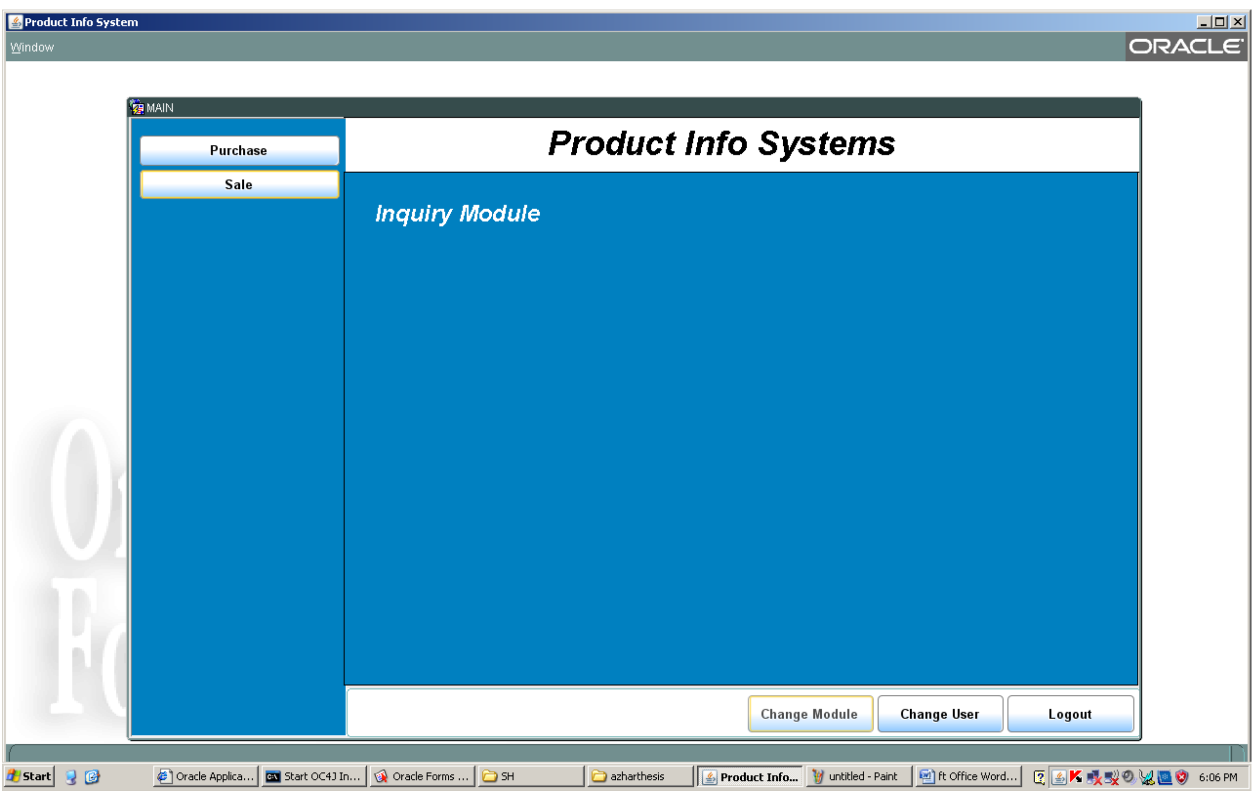
Monthly Purchase Inquiry

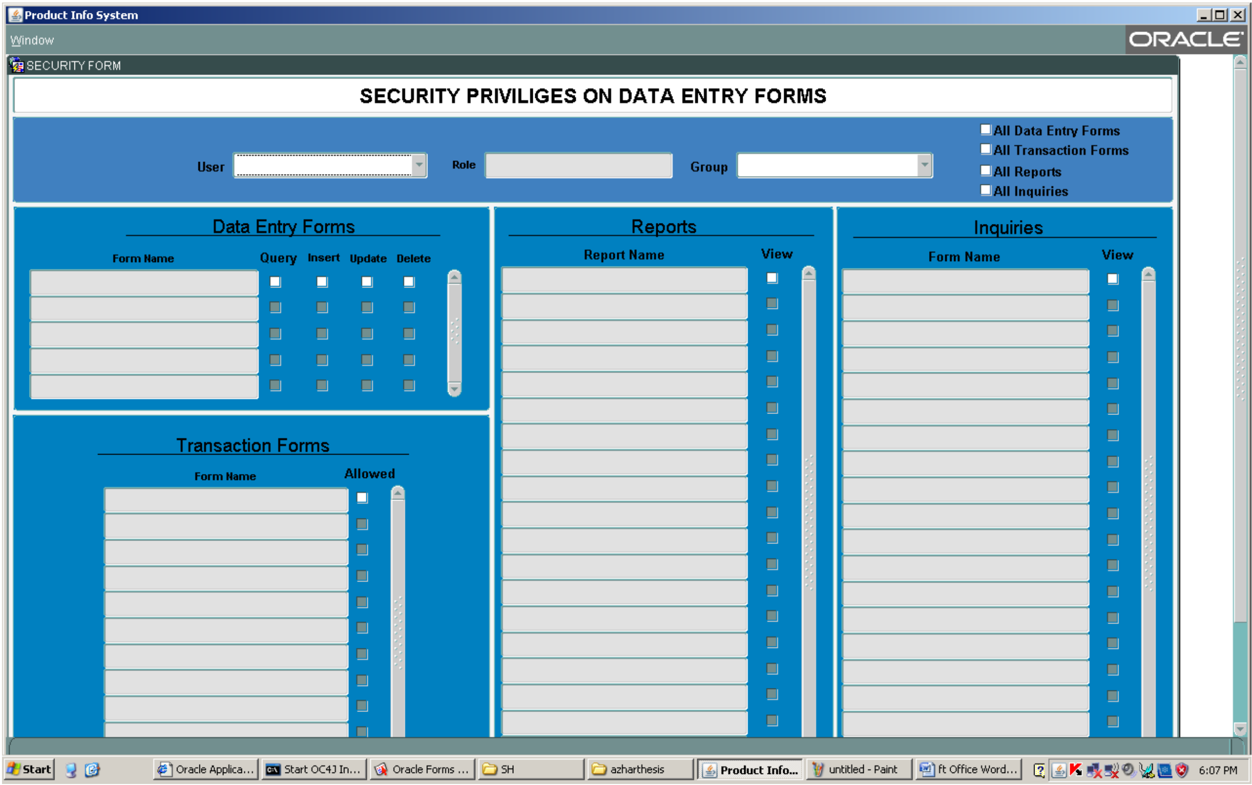
Year 2010

S No.	Month	Quantity Purchased	Purchased Amount	
2	02-2010	287775	3365005.32	Detail
3	03-2010	310260	3728426.42	Detail
4	04-2010	325005	3807934.64	Detail
5	05-2010	287820	3499245.55	Detail
6	06-2010	336555	3768777.13	Detail
7	07-2010	327900	3756547.35	Detail
8	08-2010	349950	3885461.61	Detail
9	09-2010	308685	3871578.4	Detail
10	10-2010	368055	3918667.66	Detail
11	11-2010	336045	3959490.86	Detail
12	12-2010	342135	4075266.75	Detail
				Detail

Back


Sales Inquiry
CURRENT DATE
30-DEC-2010
Sales - Dashboard
DW LOAD DATE
29-DEC-2010
Product
Quantity
Amount
External 101-key keyboard 63 665.7
PCMCIA modem/fax 28800 baud 24 441.3
SIMM- 16MB PCMCIAII card 51 2558.54
DVD-R Disc with Jewel Case, 4.7 GB 3 16.93
219 18783.38
Current Month Sale vs Target
Current
Target
0 800 1,600 2,400 3,200
Last 7 Days
Date Quantity Amount
29-DEC-2010 132 767.58
28-DEC-2010 9 50.79
27-DEC-2010 171 995.24
26-DEC-2010 12 70.08
24-DEC-2010 129 727.99
23-DEC-2010 36 210.24
Last 5 Years
Date Quantity Amount
30-DEC-2009 21 429.52
30-DEC-2008 3 16.93
30-DEC-2007 21 130.26
30-DEC-2006 6 38
DVD-R Disc with Jewel Case, 4.7 GB
Monthly Sale - 2010
2,000 1,600 1,200 800 400 0
01-2010 02-2010 03-2010 04-2010 05-2010 06-2010 07-2010 08-2010 09-2010 10-2010 11-2010 12-2010
Channel Wise %age - 2010
32.75%
57.43%
9.819%
Direct Sales
Internet
Partners
Cube
OLTP+Cube
OLTP
OLAP
Back


Product Info System
Window
ORACLE
MAIN
Purchase
Sale
Product Info Systems
Inquiry Module
Change Module
Change User
Logout


Product Info System
Window
ORACLE
SECURITY FORM
SECURITY PRIVILIGES ON DATA ENTRY FORMS
User
Role
Group
All Data Entry Forms
All Transaction Forms
All Reports
All Inquiries
Data Entry Forms
Form Name
Query
Insert
Update
Delete
Reports
Report Name
View
Inquiries
Form Name
View
Transaction Forms
Form Name
Allowed
Start
Oracle Applica...
Start OC4J In...
Oracle Forms ...
SH
azharthesis
Product Info...
untitled - Paint
ft Office Word...
6:07 PM

Publications

Muhammad Azhar Chohan and Dr.Muhammad Younus Javed, "***OLAP and OLTP Data Integration for Operational Level Decision Making***", 2010 International Conference on Database and Data Mining (ICDDM 2010)

Bibliography

[1] Golfarelli M, Rizzi S, Cella I, "Beyond Data Warehousing: What's Next in Business intelligence?" 2004

[2] Pedersen D, Riis K, Pedersen T.B, "Query Optimization for OLAP-XML Federations." 2002

[3] Pedersen D, Pedersen T.B, "Achieving Adaptivity for OLAP-XML Federation" 2003

[4] Vaduva A, Dittrich K.R, "Metadata Management for data warehousing between vision and reality" 2001

[5] Samuel S. C. "OLTP and OLAP Data Integration: A Review of Feasible Implementation Methods and Architectures or Real Time Data Analysis." 2005

[6] Özsu M.T, Valduriez P, "Principles of Distributed Database Systems" 1998

[7] Reddy M.P, Prasad B.E, Reddy P.G, Gupta A, " A Methodology for Integration of Heterogeneous Databases" 1994

[8] Thomsen E, "OLAP Solutions: Building Multidimensional Information Systems", 2nd edition. Wiley, 2002

[9] Kimbel R, "The data Warehouse Toolkit" 2002

[10] Sen A, Sinha A.P, "A Comparison of Data Warehousing Methodologies" 2005

[11] Chaudhuri S, Dayal U, "An Overview of Data Warehousing and OLAP Technology". 1997

[12] L. M. Haas,E. T. Lin,M. A. Roth "Data Integration through Database Federation". 2002

[13] Amit P. S, James A. L. "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous databases". 1990

[14] Zero-Latency Data Warehousing (ZLDWH): the State-of-the-art and experimental implementation approaches, Tho Manh Nguyen, and A Min Tjoa, Research, Innovation and Vision for the Future, 2006 International Conference on Volume , Issue , Feb. 12-16, 2006 Page(s): 167 - 176

[15] Gatierrez A. and Marotta A., An Overview of Data Warehouse Design Approaches and Techniques, Uruguay, 2000

[16] Batini C., Ceri S., Navathe S., “Conceptual Database Design-An Entity Relationship Approach”, Addison-Wesley, 1992

[17] Near Real Time ETL , Panos Vassiliadis , Alkis Simitsis , Springer 2008.

[18] Peralta V., Marotta A., Ruggia R., Towards the Automation of Data Warehouse Design, 2003