Architecture to Access OLTP data "On the Fly" from OLAP Client by Query Decomposition

TO 7349

Research Dissertation Submitted By,

Mr. Muhammad Naveed

302-FAS/MSCS/F06

Supervisor

Mr. Muhammad Imran Saeed
Assistant Professor,
Department of Computer Science.
International Islamic University, Islamabad.

Co Supervisor

Mr. Owais Usman Butt
Software Engineer,
PROSOL, Islamabad.

Department of Computer Science,
Faculty of Basic & Applied Sciences,
International Islamic University,
Islamabad.
2010

International Islamic Univ.
CENTRAL
LIBRARY
ISLAMABAD.

Submitted to the Department of Computer Science in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science

Department of Computer Science
International Islamic University Islamabad

Dated: 30-12-2010

Final Approval

It is certified that we have examined the thesis report submitted by **Mr. Muhammad Naveed, Reg No: 302-FAS/MSCS/F06**, and it is our judgment that this research project is of sufficient standard to warrant its acceptance by the International Islamic University, Islamabad for the Degree of Master of Science in Computer Science.

Committee:

External Examiner

Prof. Dr. Maqbool Uddin Shaikh
Department of Computer Science,
COMSATS Institute of Information Technology,
Islamabad.

Internal Examiner

Mr. Asim Munir
Assistant Professor
Department of Computer Science.
International Islamic University,
Islamabad.

Supervisor

Mr. Muhammad Imran Saeed
Assistant Professor
Department of Computer Science.
International Islamic University,
Islamabad.

Co-Supervisor

Mr. Owais Usman Butt
Software Engineer,
PROSOL, Islamabad.

Declaration

I hereby declare that this research dissertation is not copied as a part or whole, from any source. It is further declared that no portion of this report has been submitted in support of any other degree or qualification of this to any other university or institute of learning.

Mr. Muhammad Naveed

ACKNOWLEDGMENTS

There is no success without the will of God. I am grateful to Almighty, for showing me the path to success in this work and ever before. I am grateful to all who have made this thesis possible.

I am very grateful to my supervisor Mr. Muhammad Imran Saeed for his guidance and technical support in the project completion. It has been a pleasure to work with and learn from him.

I also want to express my sincere thanks to Dr. Muhammad Sher for his advices and tremendous support. I also thank all of them for sparing their invaluable time reviewing my work.

I owe my deepest thanks to my family who have always stood by me and guided me through my career, and have pulled me through against impossible odds at times.

DEDICATION

I would like to dedicate it to

My Family

ABSTRACT

The competition among the business enterprises increased, so they need to have efficient and right decisions in right time to remain top of the line. The success of any organization demands that the decisions made at different hierarchical levels are correct and referring not only to the historical precedence's but are also incorporating current market scenarios. Data warehousing is used by top-level management for strategic decision making in most of the organizations to facilitate analysis-based decision making. However decisions are not only to be made at the top level but this important task is also to be carried out by middle and lower levels in the organization. Better implications are guaranteed if these decisions are faster and incorporating current data along with historical data.

The conventional DW architecture does not support the real time decision making because it does not support incorporation of fresh data, so there is a need to extend the architecture to facilitate decision makers at all levels of an organization.

The proposed architecture is an extension of conventional DW architecture, which incorporates the historical data from data warehouse and fresh data from OLTP concurrently at real time (if required). The new components are query recognizer, query decomposer and query converter.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

1.	**OLAP**	On-Line Analytical Processing
2.	**DSS**	Decision Support System
3.	**DW**	Data Warehouse
4.	**DBMS**	Database Management System
5.	**GUI**	Graphical User Interface
6.	$\wedge$	Logical AND operator
7.	≬	Between symbol
8.	**DI**	Data Integrator
9.	**ETL**	Extract, transform, load
10.	**BPM**	Business Performance Management
11.	**BI**	Business Intelligence
12.	**SQL**	Structured Query Language
13.	**XML**	extended Markup Language
14.	**GM**	Global Metadata

Chapter 1

INTRODUCTION

1.1 Problem Statement

The research is aimed to propose such an architecture that caters the refresh cycle of conventional Data Warehou40se by incorporating current data from the OLTP and the data from OLAP so that the data "on the fly" is accessible. Any query coming to the warehouse is decomposed at runtime and to fetch data from OLTP and OLAP without launching ETL process to update the Data Warehouse.

1.2 Background

Before penetrating into the details of core functionality of designed system, it is appropriate way to introduce with key terminologies which are frequently used throughout thesis. Understandings to all these terminologies build a basic setup to align all readers at same level of knowledge.

1.2.1 OLTP Database

OLTP applications are designed to automate routine business processes of an enterprise such as an order entry or a sale record etc, which are routine operations. The automated operations are repetitive in nature. The records in OLTP databases are accessed or identified by primary keys. The data size varies according to organization requirements. The databases are designed to increase performance and time to complete transaction. [1]

1.2.2 Data warehouse

Inmon defines Data warehouse as

"It is subject-oriented, integrated, nonvolatile and time variant collection of data in support of management's decisions" [2].

Data warehouse stores historical data of an organization. Its purpose is to support decision making and provide data for analysis. The data warehouse is used for information reporting, multidimensional analysis and data mining.

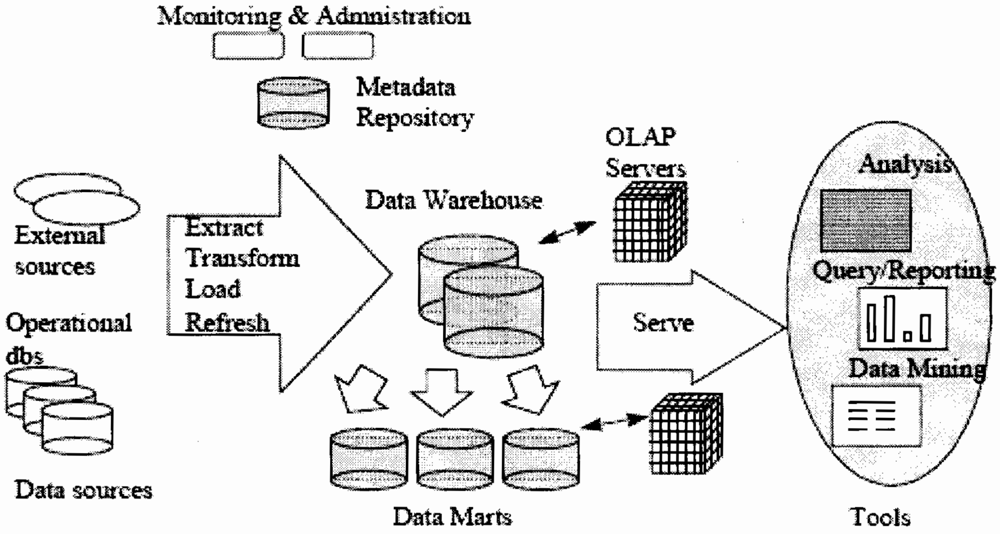
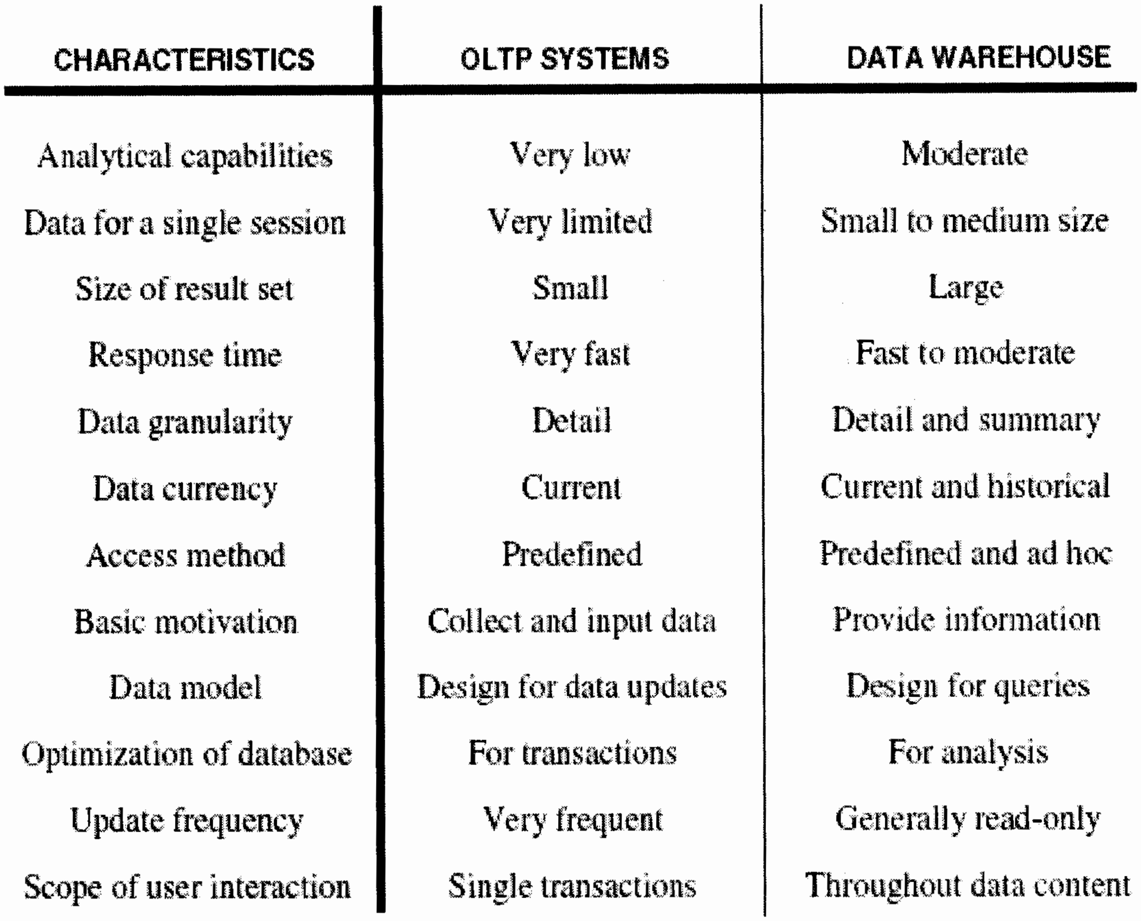


Fig 1.1: Data Warehousing Architecture [1]

Figure 1.1 is depicting the Data Ware-Housing Architecture along with essential components & their interactions. R. Kimball defines the Data warehouse as, "***It is a copy of transactional data specifically structured for querying and analysis***" [3].



CHARACTERISTICS	OLTP SYSTEMS	DATA WAREHOUSE
Analytical capabilities	Very low	Moderate
Data for a single session	Very limited	Small to medium size
Size of result set	Small	Large
Response time	Very fast	Fast to moderate
Data granularity	Detail	Detail and summary
Data currency	Current	Current and historical
Access method	Predefined	Predefined and ad hoc
Basic motivation	Collect and input data	Provide information
Data model	Design for data updates	Design for queries
Optimization of database	For transactions	For analysis
Update frequency	Very frequent	Generally read-only
Scope of user interaction	Single transactions	Throughout data content

Fig 1.2: Difference between OLTP and Data Warehouse [4]

Fig 1.2 shows the differences of data warehouse from the OLTP systems.

1.2.3. Online Analytical Processing (OLAP)

OLAP is an application to support multidimensional analysis of data. There are different OLAP operations i.e. slice-dice and pivoting etc [5].

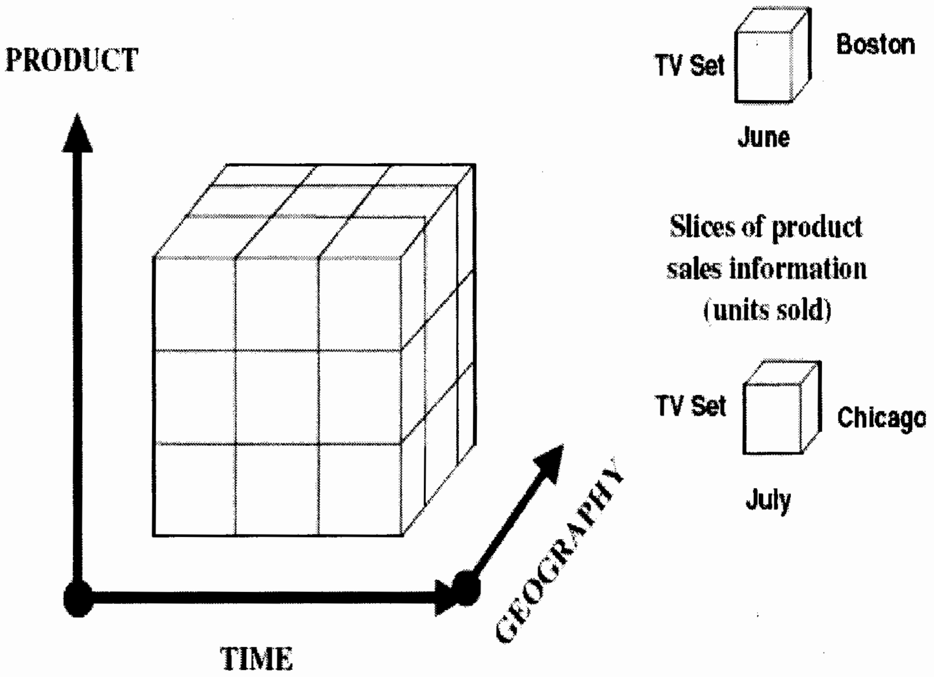


Fig 1.3: An OLAP Cube [5]

1.2.4. Data Mart

A data mart (DM) is a small data warehouse specified for a subject. Data mart contains historic data likes DW. The emphasis is on access to information relevant to a specific subject [5].

1.2.5. Extract, Transform, and Load (ETL)

ETL is an important process of DW architecture by which data from different sources and in different formats moved in to DW. In first step the data is extracted form source and then transformed into desired format by different transformation functions. Finally, the load function is used to populate DW structure [5].

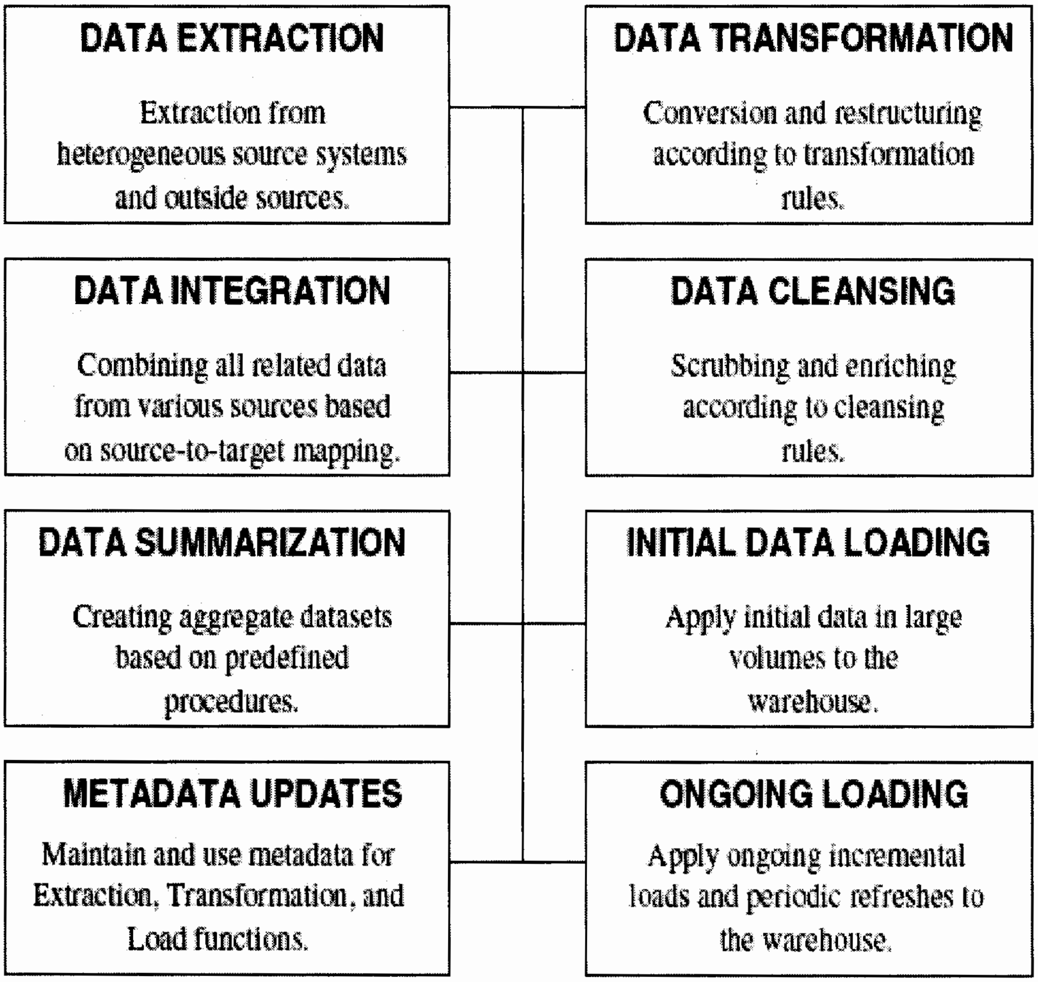

DATA EXTRACTION
Extraction from heterogeneous source systems and outside sources.
DATA TRANSFORMATION
Conversion and restructuring according to transformation rules.
DATA INTEGRATION
Combining all related data from various sources based on source-to-target mapping.
DATA CLEANSING
Scrubbing and enriching according to cleansing rules.
DATA SUMMARIZATION
Creating aggregate datasets based on predefined procedures.
INITIAL DATA LOADING
Apply initial data in large volumes to the warehouse.
METADATA UPDATES
Maintain and use metadata for Extraction, Transformation, and Load functions.
ONGOING LOADING
Apply ongoing incremental loads and periodic refreshes to the warehouse.

Fig 1.4: Summary of ETL operations [5]

1.2.6. Active Data warehouse

It is an implementation of DW whose aim is to support near-time or near-real-time decisions. The data in DW is populated on an event or on requirement by a user query. The use of active DW is a new dimension or area of research [6].

1.2.7. Real-Time Data Warehousing

In Real-time Data warehousing the DW is refreshed for every activity in the operational system. The data is available in DW once the activity is complete.

There are two key concepts that are most commonly used which are ***Real Time & Near Real Time***. They seems to be similar but if one has to summarize difference between them in single word then one can say that its ***"Latency"***, the time interval between an activity completion and availability of its data in DW. The latency is negligible in real time. [6]

1.2.8. Business intelligence (BI)

BI is the process in which data is turned into information and knowledge. BI is emerged in the early 90's as a tool for analysis of enterprise data to enhance the efficiency and decision making. BI is used in an enterprise to help their managers in situation awareness and analysis of data.

BI applications often work on historical data which is already populated in the DW and occasionally works on real time data from information systems. These applications give different views of data about business operations in form of reporting, multidimensional analysis, data mining, dashboards and KPI etc.

1.2.9. Business Performance Management (BPM)

BPM is process to enhance the performance of an organization by encouraging the effectiveness of process and efficiently use of different resources.

1.2.10. Decision levels

Three levels of decisions are generally imposed in an organization. These levels are

- Strategic
- Tactical
- Operational

Let's go through each one quickly.

Strategic

The global strategy of the enterprise is decided at strategic level.

Tactical

At tactical level there are actions and decisions corresponding a division or department. The decisions at this level must be complied with strategic level.

Operational

The decision power at this level is limited and the focus is on improving the performance of ongoing activity to meet the target values set by strategic level.

1.3 Research Problem

Research question is that is it possible to build such architecture to overcome the problem to handle the fresh data query from OLTP. The business problem is that OLAP queries are not real-time queries because of the refresh cycle of data into the OLAP data repository

- Current data warehouse solutions are designed for supporting top-level management in making decisions
- However, decisions are made on three levels in an organization, which are strategic, tactical and operational levels

- 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

1.4 Problem Domain

In conventional data warehouses transactional and operational systems are integrated into the operational data store for cleansing, transformation, extraction and loading in the OLAP repository using ETL tools. The process of transforming data into un-normalized form so that it is in sync with the OLAP structures is a time consuming and complicated process. Most of the data warehouses have a pre specified time for such tedious tasks where the frequency of such tasks is deliberately kept low.

The strategic level decision making is supported by the conventional data warehousing solutions. But the decisions are also made at tactical and operational level [7]. The decisions at these levels must be efficient and must incorporate current data from OLTP and historical data from OLAP [7]. Currently no such mechanism is evolved where logical integration of fresh data into OLAP on the fly from OLTP is taken care of.

1.5 Proposed Methodology

In order to design such an architecture, that access OLTP data "on the fly" from OLAP client by query decomposition, Data integrator and Coordinator will be integrated in data warehouse that has OLTP data incorporated with historical data along with efficient decision making . Coordinator is controller layer between end user tool & OLAP cube that facilitates and coordinates execution of query. Initially it recognizes the requirement of data. It proceeds further in its functionality by decomposing the query and then invokes different components. Query conversion is another important function of coordinator. The aim of research is to design a new architecture to incorporate fresh data(real time data, quickly changing data) which is stored in OLTP database with OLAP data so the fresher data is accessible. The query will be decomposed on runtime and no need to run ETL process of updating the DW schema for query

1.6 Thesis Organization

Rest of the thesis is organized in different chapters as follows. Chapter 2 shows the flashback of the prior research in an effort to concurrently query OLTP and Data Warehouse at real time. Chapter 3 illustrates the detailed design of proposed architecture along with methodology adapted to achieve goal. Chapter 4 gives the implementation & deployment detail. This section also includes the flow controls to give detail view of implementation and deployment of proposed system. Chapter 5 presents in detail the testing and performance evaluation of developed system. Chapter 6 gives the conclusions of research work and Chapter 7 winds up the research work along with enhancement adopted in future to increase functionality.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This chapter includes the work done by different researchers in the field of accessing OLTP data from OLAP Client. The chapter initially describes different approaches used for said purpose. Then it covers in detail the key ideas of different researchers. Aim is to provide the comparison of different techniques to achieve the same goal.

2.2 Operational Business Intelligence

DI Sandu [24] explains the concept of Operational and real-time Business Intelligence. According to [24] there are three types of Business Intelligence. The types are strategic, tactical and operational.

2.3 Closed-Loop in BPM Approach

In BPM approach the focus is on enterprise level goals for every process. The people synchronize their working by share their business goals and strategy. In this approach the strategic level decisions translated to get results into multiple lower level goals. For lower level management a target value for a given indicator is defined, which measures the operation. Fig. 2.1 shows this approach which is based on a closed-loop where:

- *The enterprise performance is influenced by the strategy and corresponding targets on indicators which are extracted from operational systems.*
- *At tactical level the actions/decisions are focused on matching the current values for indicators.*
- *The company strategy fulfilled by actions/decisions and determine its*

performance.

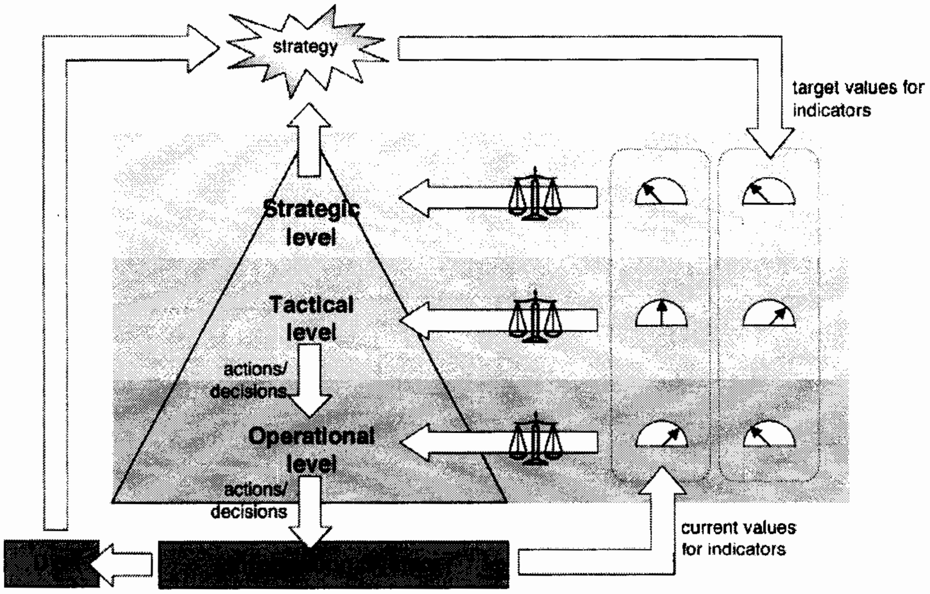


Fig 2.1: The Closed-loop in the BPM [7]

As shown in Fig 2.1 there is a DW in which data is extracted from information system. There are different levels of decisions in the organization. The DW is used at strategic level for analysis and to define targets for different operations. These target values will be used by tactical and operational level decision makers to compare their current performance.

2.4 OLAP-XML Federations Approach for Logical Integration

To meet the data requirements of different processes in the business environment is not catered well by physically integrating external data to data warehouses. Physical integration is a process which takes long time so logical integration, i.e., federation is a better choice.

"A federation consists of a cube, a collection of XML documents, and the links between the cube and the documents."[8]

The examples of such types of data are stock quotes or price lists.

Pedersen, Riis, and Pedersen [8] presented the concept of 'OLAP-XML federation' to logically integrate this type of data. Their approach has been summarized in figure 2.2

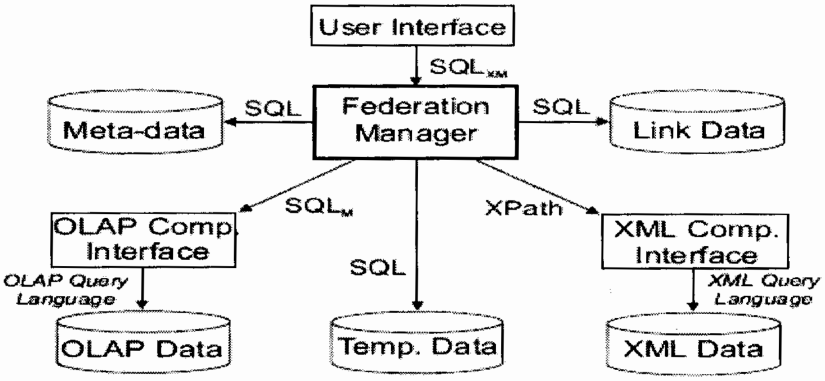


Fig 2.2: OLAP-XML Federations Approach [8]

2.5 OLTP and OLAP Data Integration Approach

Samuel S. Conn [9] reviewed different methods for real time data analysis. He answers different questions relating to integration of OLTP and OLAP. He discusses all the possible scenarios of OLTP and OLAP data integration. This study explains all aspects and justifications of data integration for real time data analysis.

The key points of his work are:-

- Because of the refresh cycle of data the OLAP queries are not real-time
- Both the environments are different because there design is based on different purposes.
- OLTP and OLAP are different environments, they need different data models.
- Logical integration of OLTP and OLAP data is feasible through materialized views.
- Same DBMS or database engine can be used for OLTP and OLAP.

2.6 Existing Approaches to access OLTP data from OLAP Client

The prior work done, to achieve the similar goal, is always a source of inspiration and provides one's the way to explore different aspects of same problem. On one stage it guides the new comer and on other provides him a chance to come up with better & improved version.

If we throw light on the work done by different researchers in same field, we find that almost all of them have based their work on four things which are [9]:-

- OLAP plays a role of decision support environment.
- Main architecture for OLAP system is data warehousing.
- Integration of OLAP & OLTP.

Everyone tries to add different flavors to make his work a bit unique from others. Let's have a look on research done by few of them.

- Jones [12] pointed out that operational characteristics of the data environment are the main reason to separate OLTP & OLAP. OLTP system fulfills the operational requirements Decision support is provided by OLAP environment.
- The distinction between OLAP & OLTP environments is discussed by Bontempo [10].
- Gray & Watson [11] in their work analyze different requirements of data in a data warehouse. Their focus is on following issues that:-
  - What sort of data structure used?
  - Data is in which form?
  - What is normal flow of data?

    - Which OLAP architecture is used as decision support & knowledge management?

- Eisenberg & Melton [14], emphasis the salient features of SQL which have been summarized as follow
    - Standards for object extensions & capability.
    - Analytical functions for summarization of groups & aggregation of data for decision support applications.
- From technical point of view [15], data mining on an OLTP is possible.
- The integration of OLTP & OLAP environments is complex. [17]. There are difficulties in using OLTP to support multidimensional analysis.
- According to [18] materialized views can be used to support OLAP operations.
- The data in the operational system can be used for analysis by grouping and aggregation functions [19].
- RiTE ("Right-Time ETL") [25], a middleware system to insert real time data that becomes available to data consumers on demand.
- [26] describe the requirements for data integration flows of operational BI system, the limitations
- BIAEditor [27] allows business impact analysis by matching operational data models

Many researchers have published & shared their master pieces in same field and some are in effort to hit the target. The need there, to have glimpse in the work of few people, is that one can have fair idea regarding the technology in question and one can easily build contingency plan for upcoming short-comes in this way.

2.7 Conditions and Assumptions

In order to efficiently and accurately demonstrate the functionality of proposed system, certain pre-conditions and assumptions should be met.

- System must feed ***"Real Time Data"*** for normal processing.
- ***"Real Time Data" & "Historical Data"*** should be merged.
- Functionality at backend should be transparent from user as he shouldn't be aware of data type.
- Every Query should be based on ***"Time Stamp"***

2.8 Summary

This chapter reviews existing approaches to access OLTP data from OLAP Client. Salient features which are core of focus have been discussed briefly. Researchers work is almost similar apart from few distinguishing features that upright one from the rest. Moreover the conditions and assumption that must be in mind before analyzing the success rate has been discussed in detail.

Chapter 3

ARCHITECTURE DESIGN

3.1 Introduction

This chapter gives the design and implementation details of the proposed framework. The chapter will start with basic level design and than proceeded further down till the roots of detailed design architecture have been elaborated. This will not only enhance the understanding but also throw light on different components of system under discussion.

3.2 Characteristics of Proposed Architecture

Proposed Architecture has some differences as compared to Basic Data Warehouse Architecture. Some of characteristics that mark difference between the two are:-

- As contrast to basic data warehouse architecture, this architecture is workable for an enterprise where historical data is used on the strategic level for decision making.
- The proposed architecture feed with fresh data at real time to support all levels of management in better decision making.
- It extracts historical data from OLAP cube and fresh data from operational system, when given dates are after the load date of data warehouse.

3.3 Designing Levels of Framework

"Designing Levels" are one of important artifacts produced by designer. This enhances clarity and help to understand how different sub-components will be integrated and communicated. One can have multiple levels of designs, depending upon the complexity of project. In our scenario, I have explained the proposed architecture in two levels. The level 1 is an overview and then details in level 2. Let's have a look on each of them one by one.

3.3.1 Basic Level Design of Proposed Architecture

This level of design is an extension of conventional warehouse architecture. Figure 3.1 shows the components that make up a basic architecture along with order in which they communicate to perform the job.

Basic Level Design Architecture

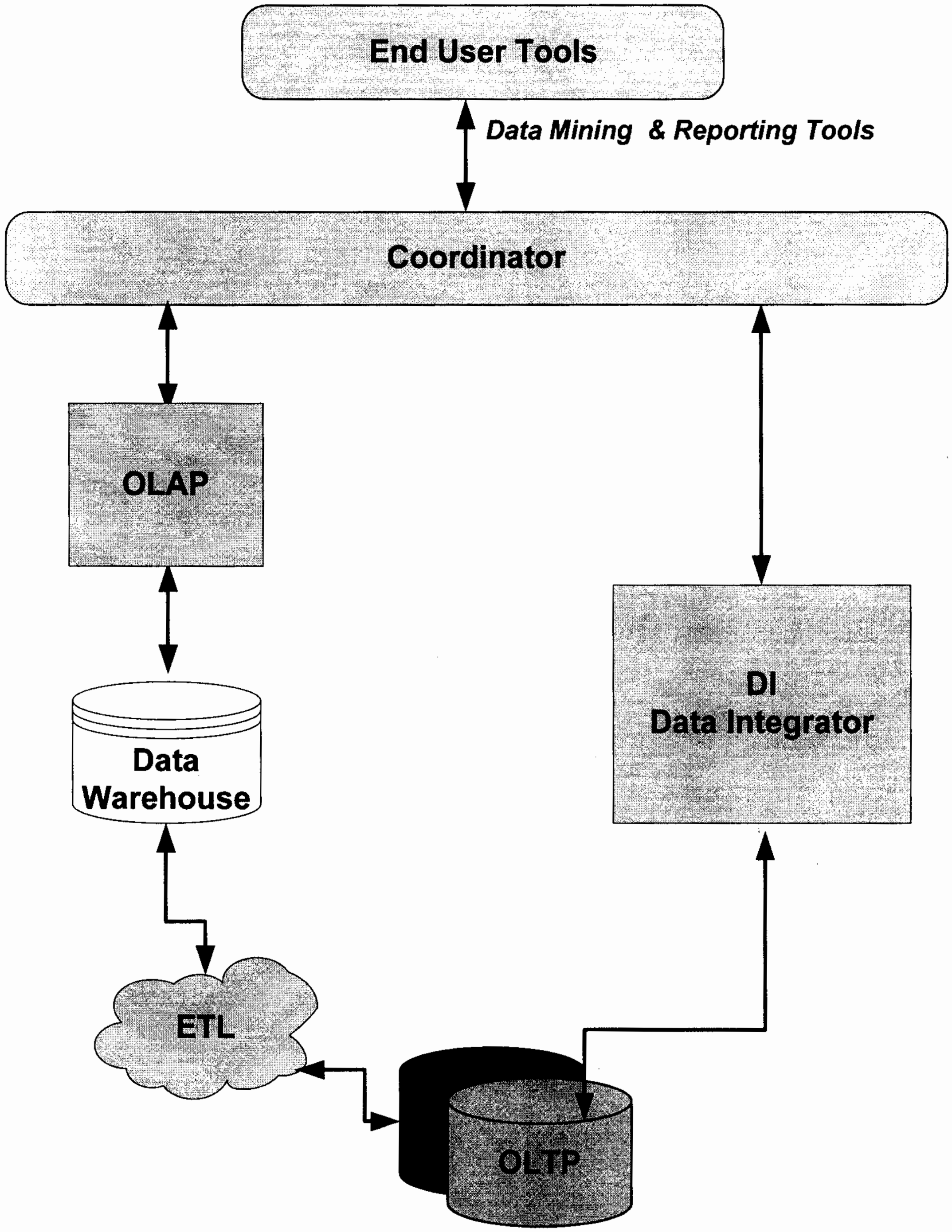


Figure 3.1: Overview of Proposed Architecture

The components of the proposed architecture are as following:-

- Global Metadata
- Coordinator
- Data Integrator
- OLAP
- Data Warehouse

End User Tool, OLAP and Data warehouse are the parts of conventional warehouse architecture. The extension to conventional setup is coordinator and Data integrator.

Coordinator

It decides the path of the query processing and coordinates execution of user's query. Its main functions are query recognition, decomposition and conversion. It recognizes the query and check that which type of data is queried by the user. Its important function is query decomposition, in which the query will be decomposed in two parts.

Data Integrator (DI)

The DI manages the access of fresh data from OLTP system. This is an interface to connect to OLTP, which facilitates in the availability of fresh data at real time.

3.3.2 Detailed Level Design of Proposed Architecture

Detailed level design of proposed architecture is shown in figure 3.2. The components are further elaborated in detail to have complete picture. The basic components of any conventional architecture have not been decomposed further. Focus is only on those components that make proposed architecture unique than any other setup.

Detailed Level Design Architecture

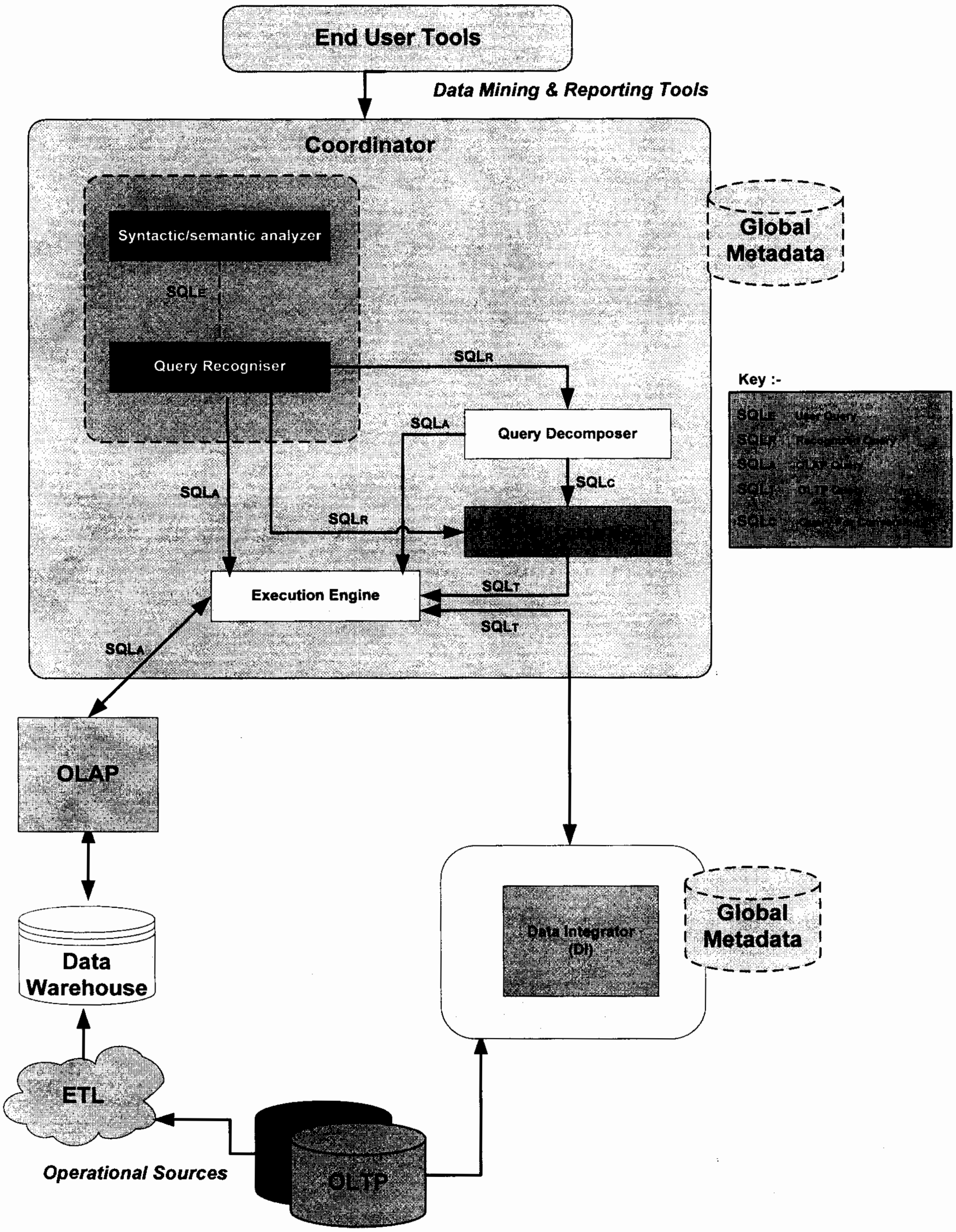


Figure 3.2: Detailed Design of Proposed Architecture

3.3.2.1 Inner Core of Coordinator

The controller of proposed architecture has to do many jobs simultaneously which includes data fetching, decomposition and conversion of queries. All this is not possible without cooperation and communication of different sub-components. Each of these sub-components has been assigned a particular job and successful completion of all milestones will together bring one goal. Following are the main sub-components of Coordinator:-

- Syntactic /semantic analyzer
- Query Recognizer
- Query Decomposer
- Query Converter
- Execution Engine
- Global Metadata

As indicated in figure 3.2, the output of one may be the input of other. Which means that this closely integration has effect on overall efficiency and accuracy of the system. Let's have look on each of these sub-components functionality.

Syntactic /semantic analyzer

The query processing starts by this component. The syntax of the query will be checked. The syntax rules and meanings of the query will be checked. Finally the query is simplified and restructured to make it ready for execution. It can be broken down into different processes.

Query Recognizer

An important component and is used to identify whether end user query needs OLAP data or OLTP data (fresh data) or OLAP and fresh both. The query will be recognized by the date given in user query. The query date will be matched with the last load date of the DW. The execution engine will be invoked for historic data requirement. If fresh data is queried by user as well as historical data than decomposition is required to divide the query. For the fresh data requirement the query converter will be invoked.

Query Decomposer

It is invoked by query recognizer to divide the query. The first query will be executed on OLAP by execution engine to get data up to load data. The other query will be converted by query converter.

Query Converter

This is invoked by query decomposer to convert an OLAP query into an OLTP query. This component use Global Metadata for conversion.

Execution Engine

This component invoked from different components for execution of queries on OLAP and OLTP systems. The query for OLTP will be executed by Data Integrator (DI). When it receives query data it sent back to end user.

Global Metadata

It stores data about the different components of data warehouse architecture. It stores information about operational systems, OLAP and Data warehouse structure.

3.3.2.2 Inner Core of Data Integrator

The Data Integrator (DI) integrates the data from OLTP at 'the right time. The data is integrated on requirement. [7].

3.4 Flow Graph

Flow Graph is a pictorial way to depict the flow of system under all circumstances. At this stage reader is familiar with main components of the system, that's why it is appropriate to further educate him by demonstrating the flow graph.

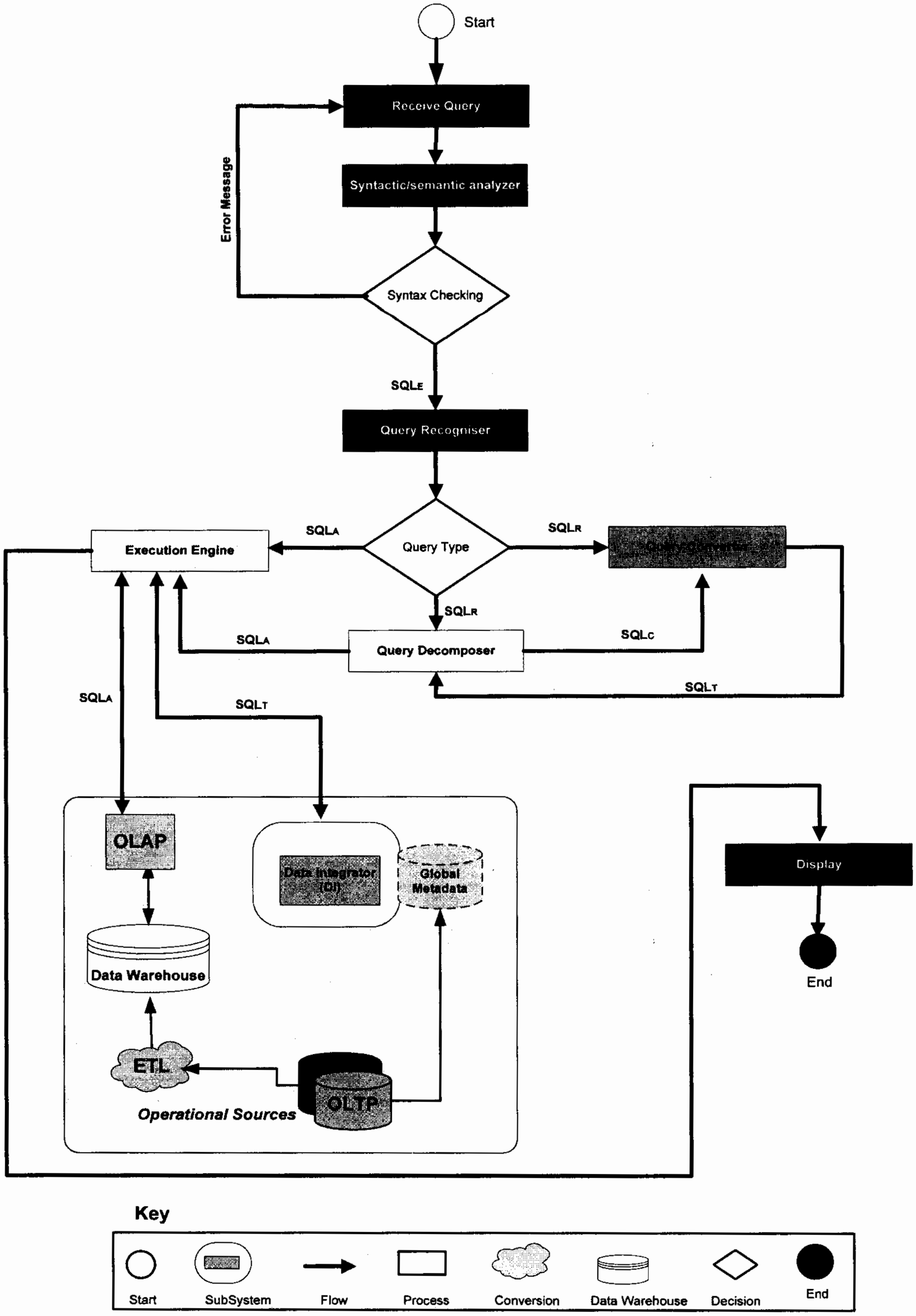


Figure 3.3: Flow Graph

3.5 Different components origin

The main core of design work has been elaborated at two levels. Their details have been described below:-

Level 1

The concept of fresh data requirement for analysis was introduced by Golfarelli [7]. Pederson et al [2] introduced the idea of "OLAP-XML federation" for logical integration of OLAP and XML data.

Level 2:

Level 2 explain detailed working. I did not develop any new component to achieve my objectives. I have used existing components from different research works. I have modified them to achieve my functionality. Pedersen et al [8] has implemented the concept of recognition of query. In distributed databases the process of query decomposition is used [21] and is also implemented by Pederson et al [8]. The concept of Query converter is from [8] and [23] and the idea of DI from [7].

3.6 Database Designing of Proposed Architecture

Database designing comprises of two parts which are as follow:-

1. Data Warehouse Schema Design
2. OLTP Schema Design

Let's have look on each of them.

3.6.1 Data Warehouse Schema Design

Both of these designs are well explanatory when depicted through ERD snapshot. The figure 3.3 is a data warehouse design of example scenario for the implementation of proposed architecture. The tables along with their attributes and relationship show how they are linked and interact with each other.

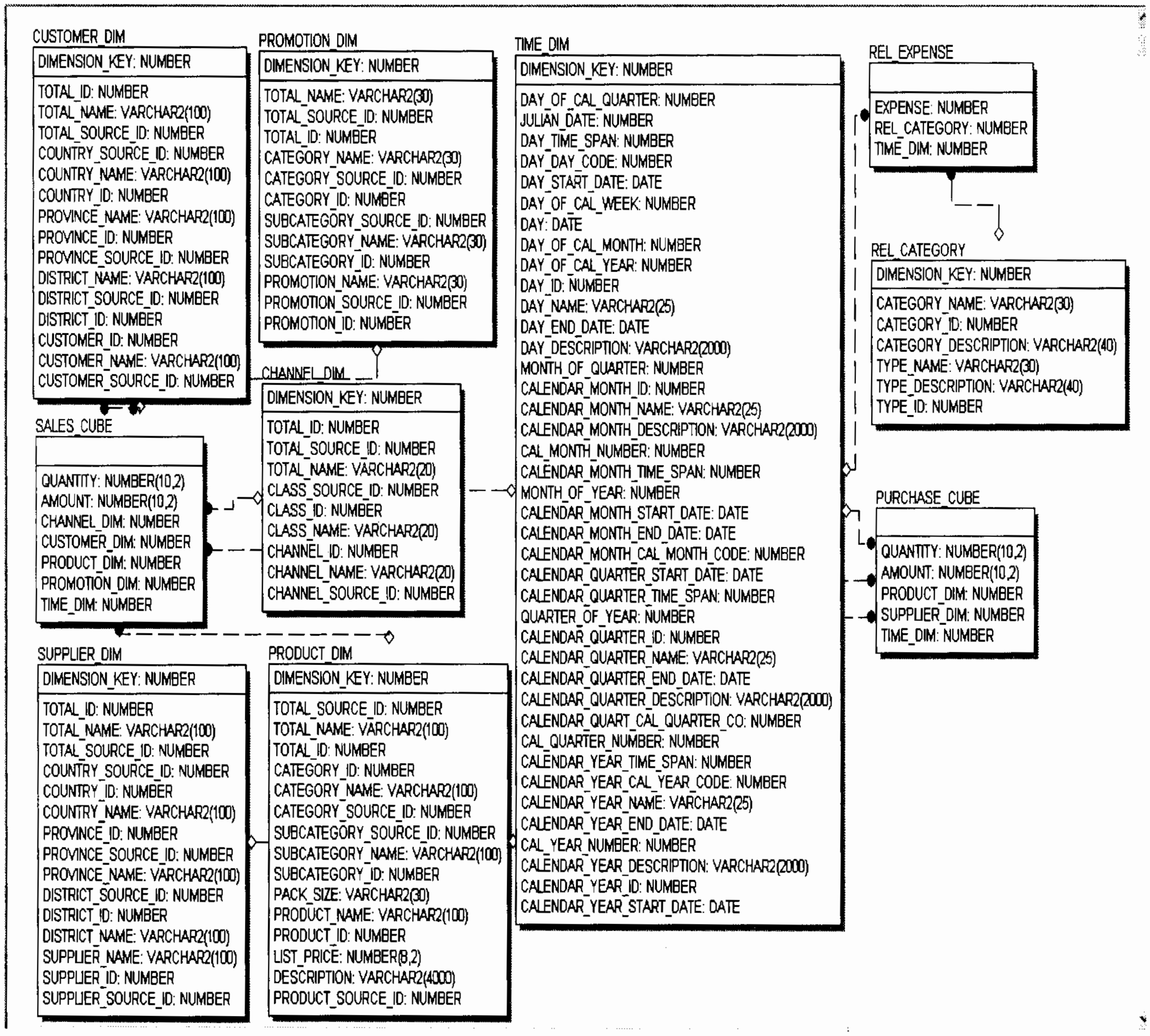

CUSTOMER_DIM
DIMENSION_KEY: NUMBER
TOTAL_ID: NUMBER
TOTAL_NAME: VARCHAR2(100)
TOTAL_SOURCE_ID: NUMBER
COUNTRY_SOURCE_ID: NUMBER
COUNTRY_NAME: VARCHAR2(100)
COUNTRY_ID: NUMBER
PROVINCE_NAME: VARCHAR2(100)
PROVINCE_ID: NUMBER
PROVINCE_SOURCE_ID: NUMBER
DISTRICT_NAME: VARCHAR2(100)
DISTRICT_SOURCE_ID: NUMBER
DISTRICT_ID: NUMBER
CUSTOMER_ID: NUMBER
CUSTOMER_NAME: VARCHAR2(100)
CUSTOMER_SOURCE_ID: NUMBER
PROMOTION_DIM
DIMENSION_KEY: NUMBER
TOTAL_NAME: VARCHAR2(30)
TOTAL_SOURCE_ID: NUMBER
TOTAL_ID: NUMBER
CATEGORY_NAME: VARCHAR2(30)
CATEGORY_SOURCE_ID: NUMBER
CATEGORY_ID: NUMBER
SUBCATEGORY_SOURCE_ID: NUMBER
SUBCATEGORY_NAME: VARCHAR2(30)
SUBCATEGORY_ID: NUMBER
PROMOTION_NAME: VARCHAR2(30)
PROMOTION_SOURCE_ID: NUMBER
PROMOTION_ID: NUMBER
TIME_DIM
DIMENSION_KEY: NUMBER
DAY_OF_CAL_QUARTER: NUMBER
JULIAN_DATE: NUMBER
DAY_TIME_SPAN: NUMBER
DAY_DAY_CODE: NUMBER
DAY_START_DATE: DATE
DAY_OF_CAL_WEEK: NUMBER
DAY: DATE
DAY_OF_CAL_MONTH: NUMBER
DAY_OF_CAL_YEAR: NUMBER
DAY_ID: NUMBER
DAY_NAME: VARCHAR2(25)
DAY_END_DATE: DATE
DAY_DESCRIPTION: VARCHAR2(2000)
MONTH_OF_QUARTER: NUMBER
CALENDAR_MONTH_ID: NUMBER
CALENDAR_MONTH_NAME: VARCHAR2(25)
CALENDAR_MONTH_DESCRIPTION: VARCHAR2(2000)
CAL_MONTH_NUMBER: NUMBER
CALENDAR_MONTH_TIME_SPAN: NUMBER
MONTH_OF_YEAR: NUMBER
CALENDAR_MONTH_START_DATE: DATE
CALENDAR_MONTH_END_DATE: DATE
CALENDAR_MONTH_CAL_MONTH_CODE: NUMBER
CALENDAR_QUARTER_START_DATE: DATE
CALENDAR_QUARTER_TIME_SPAN: NUMBER
QUARTER_OF_YEAR: NUMBER
CALENDAR_QUARTER_ID: NUMBER
CALENDAR_QUARTER_NAME: VARCHAR2(25)
CALENDAR_QUARTER_END_DATE: DATE
CALENDAR_QUARTER_DESCRIPTION: VARCHAR2(2000)
CALENDAR_QUART_CAL_QUARTER_CO: NUMBER
CAL_QUARTER_NUMBER: NUMBER
CALENDAR_YEAR_TIME_SPAN: NUMBER
CALENDAR_YEAR_CAL_YEAR_CODE: NUMBER
CALENDAR_YEAR_NAME: VARCHAR2(25)
CALENDAR_YEAR_END_DATE: DATE
CAL_YEAR_NUMBER: NUMBER
CALENDAR_YEAR_DESCRIPTION: VARCHAR2(2000)
CALENDAR_YEAR_ID: NUMBER
CALENDAR_YEAR_START_DATE: DATE
REL_EXPENSE
EXPENSE: NUMBER
REL_CATEGORY: NUMBER
TIME_DIM: NUMBER
REL_CATEGORY
DIMENSION_KEY: NUMBER
CATEGORY_NAME: VARCHAR2(30)
CATEGORY_ID: NUMBER
CATEGORY_DESCRIPTION: VARCHAR2(40)
TYPE_NAME: VARCHAR2(30)
TYPE_DESCRIPTION: VARCHAR2(40)
TYPE_ID: NUMBER
CHANNEL_DIM
DIMENSION_KEY: NUMBER
TOTAL_ID: NUMBER
TOTAL_SOURCE_ID: NUMBER
TOTAL_NAME: VARCHAR2(20)
CLASS_SOURCE_ID: NUMBER
CLASS_ID: NUMBER
CLASS_NAME: VARCHAR2(20)
CHANNEL_ID: NUMBER
CHANNEL_NAME: VARCHAR2(20)
CHANNEL_SOURCE_ID: NUMBER
SALES_CUBE
QUANTITY: NUMBER(10,2)
AMOUNT: NUMBER(10,2)
CHANNEL_DIM: NUMBER
CUSTOMER_DIM: NUMBER
PRODUCT_DIM: NUMBER
PROMOTION_DIM: NUMBER
TIME_DIM: NUMBER
PURCHASE_CUBE
QUANTITY: NUMBER(10,2)
AMOUNT: NUMBER(10,2)
PRODUCT_DIM: NUMBER
SUPPLIER_DIM: NUMBER
TIME_DIM: NUMBER
SUPPLIER_DIM
DIMENSION_KEY: NUMBER
TOTAL_ID: NUMBER
TOTAL_NAME: VARCHAR2(100)
TOTAL_SOURCE_ID: NUMBER
COUNTRY_SOURCE_ID: NUMBER
COUNTRY_ID: NUMBER
COUNTRY_NAME: VARCHAR2(100)
PROVINCE_ID: NUMBER
PROVINCE_SOURCE_ID: NUMBER
PROVINCE_NAME: VARCHAR2(100)
DISTRICT_SOURCE_ID: NUMBER
DISTRICT_ID: NUMBER
DISTRICT_NAME: VARCHAR2(100)
SUPPLIER_NAME: VARCHAR2(100)
SUPPLIER_ID: NUMBER
SUPPLIER_SOURCE_ID: NUMBER
PRODUCT_DIM
DIMENSION_KEY: NUMBER
TOTAL_SOURCE_ID: NUMBER
TOTAL_NAME: VARCHAR2(100)
TOTAL_ID: NUMBER
CATEGORY_ID: NUMBER
CATEGORY_NAME: VARCHAR2(100)
CATEGORY_SOURCE_ID: NUMBER
SUBCATEGORY_SOURCE_ID: NUMBER
SUBCATEGORY_NAME: VARCHAR2(100)
SUBCATEGORY_ID: NUMBER
PACK_SIZE: VARCHAR2(30)
PRODUCT_NAME: VARCHAR2(100)
PRODUCT_ID: NUMBER
LIST_PRICE: NUMBER(8,2)
DESCRIPTION: VARCHAR2(4000)
PRODUCT_SOURCE_ID: NUMBER

Figure 3.4: Data Warehouse Schema (Example Scenario)

3.6.2 OLTP Schema Design

Similar is the case of OLTP schema design which is shown in figure 3.4

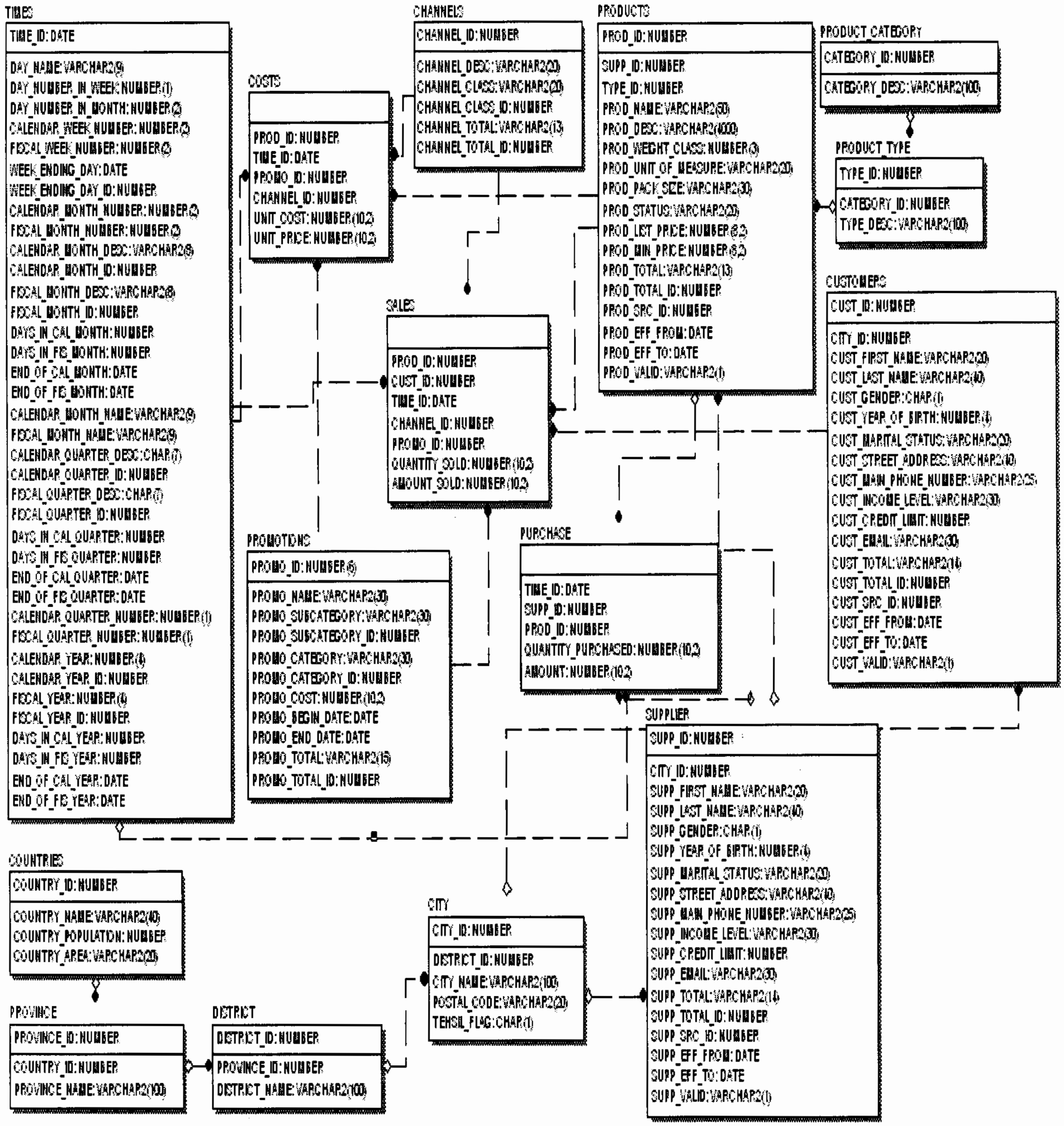


Figure 3.5: OLTP Schema (Example Scenario)

3.6.3 Class Diagram

During design phase, different classes are created which represent different functions and attributes. Class diagram shows the interaction and relationship among different classes as shown bellow.

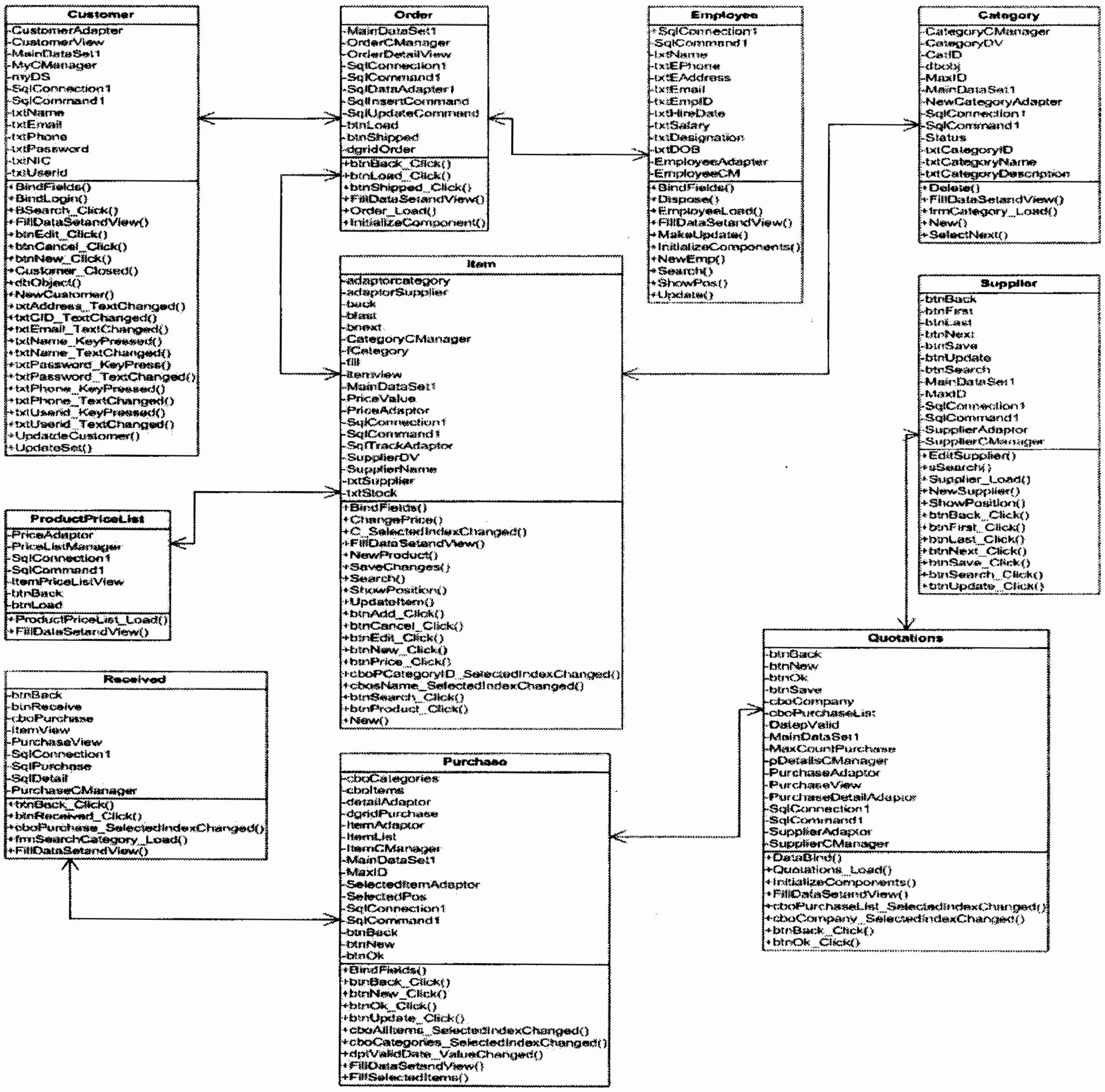


Figure 3.6: Class Diagram

3.6.4 Sales Cube

Figure 3.7 shows a sales cube. It has Fact_Sale as fact table and Channel_Dim, Customer_Dim, Product_dim, Time_Dim and Promoion_Dim dimensional tables. Quantity and amount in fact table are measures.

TH 7349

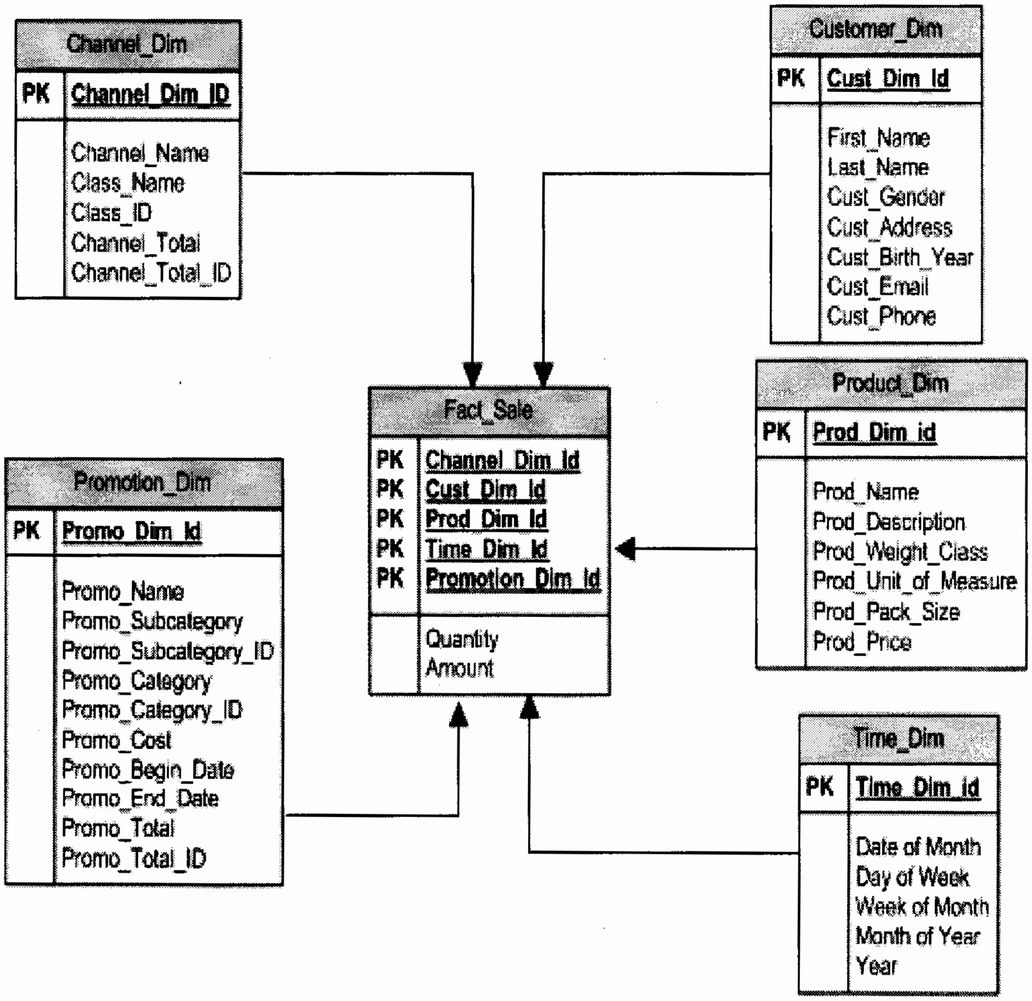


Figure 3.7: Sales Cube

3.6.5 Purchase Cube

Figure 3.8 shows Purchase cube. Fact_Purchase is fact table and Supplier_Dim, Time_Dim and Product_Dim are dimensional tables. Quantity and Amount in fact table are measures.

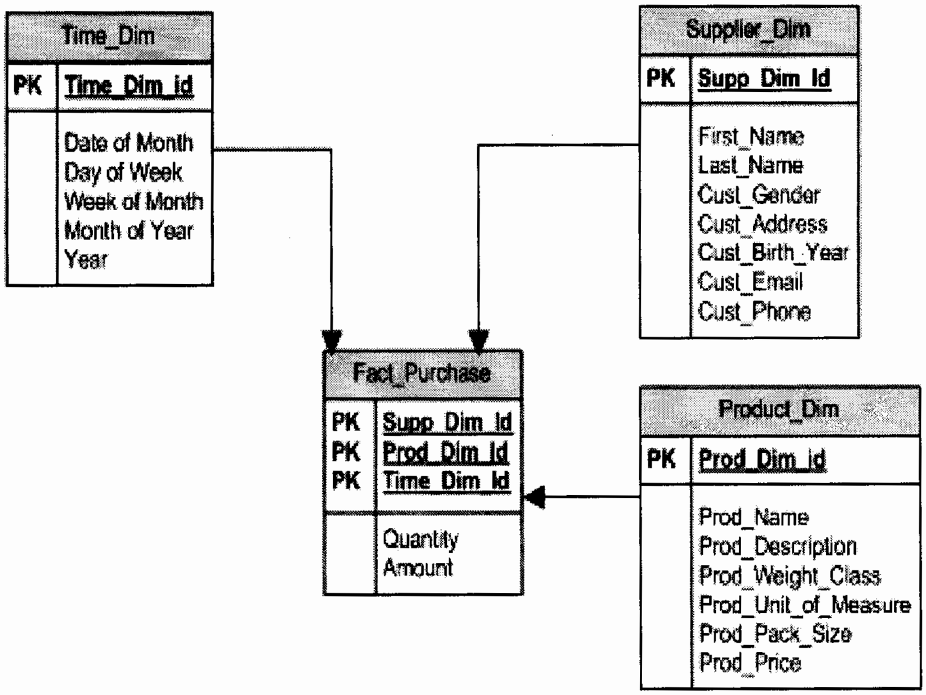


Figure 3.8: Purchase Cube

3.7 Summary

Chapter 3 glimpse in to the details of design framework by incorporating the basic and detail level designing. As we move down, more and more detail is captured and things become crystal clear from black box to the functionality of single component. The ways different components integrate and communicate have been discussed including the protocol followed to accomplish the task.

Chapter 4

IMPLEMENTING THE FRAMEWORK

4.1 Introduction

The different components of the proposed architecture will be described in detail in this chapter. It will also describe the interaction among different components and flow of query processing through algorithms. The query processing will be explained through an example scenario and will give process details of the architecture. The example given below is a scenario that can exist in a large enterprise.

4.2 Supposed Scenario

ABC is an IT equipment company. There is an OLTP system for routine sale activity and has a data warehouse. There is an OLAP cube for sales having data like sales of equipment. The dimensions for the sales cube are customer, Product and Date. The loading of data warehouse processed after every 24 hours at night. The top management set the targets of sales after analysis on historical data and the lower management monitors their progress by checking their current sale. There is a need of fresh data at the right time for performance check and to meet the targets set by higher management.

4.3 Global Metadata (GM)

There are two types of data in the global metadata i.e. OLAP system (GM_O) and for the OLTP system (GM_T). There are three types of information.

- The data for end users requirements, like specific documentation is called **Business metadata**.
- Schema definition, physical storage information, configuration etc. called **Technical metadata**.
- The data needed by ETL process of data warehouse system is called **Transformational metadata**.

The query converter will use metadata by consulting transformational rules and data mapping for query conversion.

4.4 Syntactic and Semantic analyzer

The query processing starts by this component. The syntax of the query will be checked. The syntax rules and meanings of the query will be checked. Finally the query is simplified and restructured to make it ready for execution. It can be broken down into different processes.

4.5 Query Recognizer

It is a core of the Coordinator. It recognizes the data requirement on the base of date given in the user query. It explains the path and invokes other components.

There are three different types of data.

Historical data

If user queried the historical data then the query will be sent to the Execution Engine.

Historical and Fresh data

If both historical data and fresh data are required then Query Decomposer will be invoked.

Fresh Data

If fresh data is queried only, then it invokes the query converter.

Algorithm for Query Recognition

This algorithm titled as "Query_Recognizer ", identify the data requirement from query and direct it towards right component. Its pseudocode is as follow:

Key:-

[illegible]	[illegible]
$Date_E$	Date attribute in user query
SQL [illegible]	[illegible]
$Date_{Load}$	Last upload date of DW

Function Name: - **Query_Recognizer (SQL_E)**

Input: - SQL_E

Output: - Performed the task assigned that is to identify the data requirement from query and direct it towards right component.

Corrosponding functions:-

- **Metadata_Consult ();** // get names of tables and attributes
- **Query_Info ();** //fetches conditions within the SQL_E
- **Get_load_date();** //fetches last load date from metadata
- **Query_Decomposer()**
- **Query_Converter();** //Query conversion
- **Execution_Engine();** // Execute query

```
Begin
if (($Date_E$ $\exists$ in $SQL_E$) $\wedge$ ($Date_E$ $\in$ in $SQL_{Condition}$)) then
{       $Date_{Load}$ $\leftarrow$ Get_load_date();
        if ($Date_{Load}$ $\between$ ($SQL_E$ .Date) then        //if load date is in between query dates
                func query_decomposer($SQL_E$.date1,$Date_{Load}$,$SQL_E$.date2);
                                                    // query is based on both sources
        elseif ($SQL_E$ .Date_to < $Date_{Load}$ )then  //if load date is greater than query dates
                exec_flag$\leftarrow$ ‘OLAP’;
                func Execution_Engine();        //query based on historical data
        elseif ($SQL_E$ .Date_from > $Date_{Load}$ ) then  //if load date is smaller than query
                                                        dates
                exec_flag$\leftarrow$ ‘OLTP’;
                func Query_Converter();         //query based on fresh data
} End
```

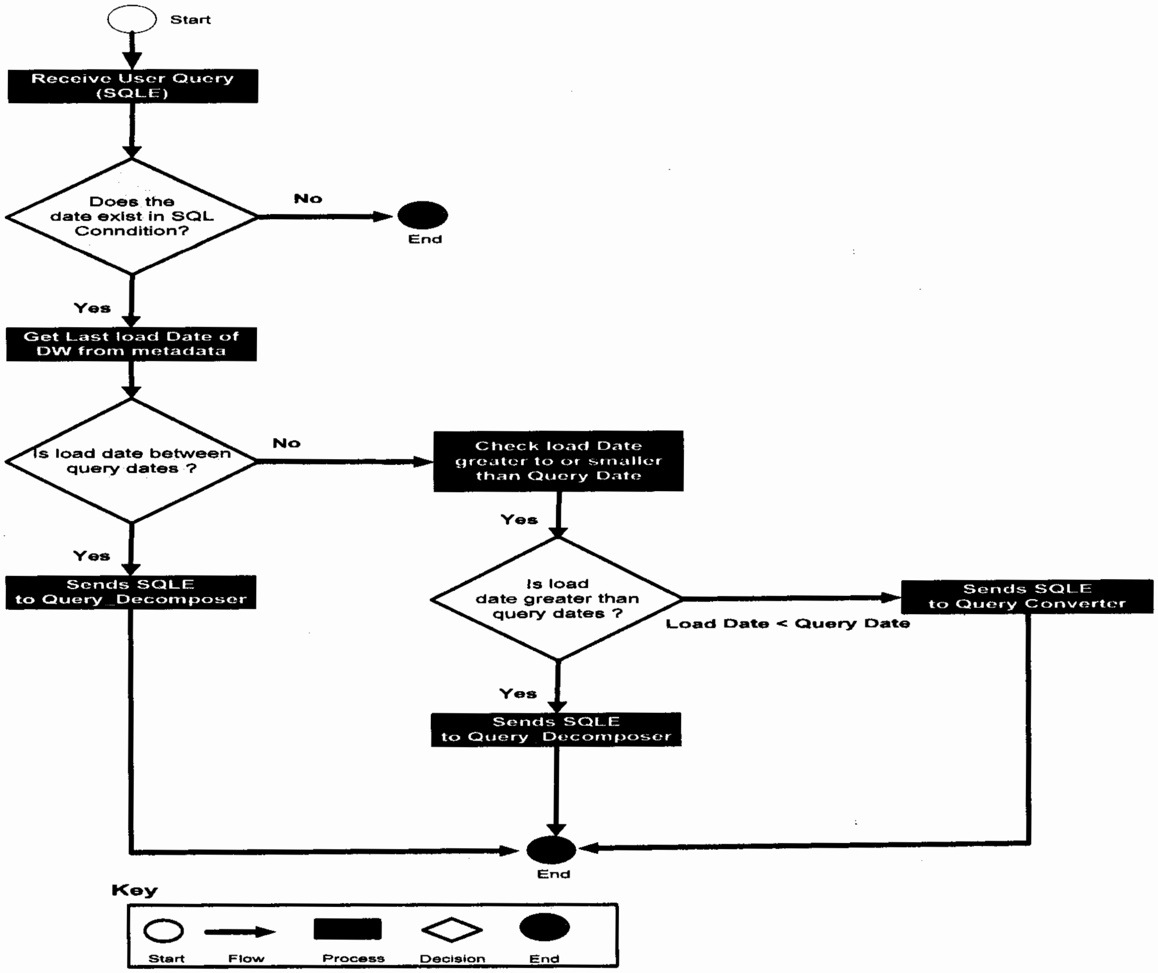

Start
Receive User Query (SQLE)
Does the date exist in SQL Conndition?
No
End
Yes
Get Last load Date of DW from metadata
Is load date between query dates ?
No
Check load Date greater to or smaller than Query Date
Yes
Yes
Sends SQLE to Query_Decomposer
Is load date greater than query dates ?
Sends SQLE to Query Converter
Load Date < Query Date
Yes
Sends SQLE to Query_Decomposer
End
Key
Start
Flow
Process
Decision
End

Figure 4.1 Activity Diagram for Query_Recognizer()

4.6 Query Decomposer

This function receives dates of user query from Query_Recognizer(). The following algorithms will describe the details of this function.

Algorithm for Func Query_Decomposer (date1,date2,load_date);

The query will be decomposed into two OLAP queries by this function. The first one up-to load date and second one from load date to end date. The first query will be sent to execution engine and the second query will be sent to converter.

Key:-

[redacted]	[redacted]
[redacted]	[redacted]
Load_Date	The date at which data loaded
[redacted]	[redacted]
SQL$_A$	SQL format passed to engine

Function Name: - **Query_Decomposer (date1,date2,load_date);**

Input: - **date1, date2, load_date**

Output: - Decomposed query, one in OLAP and other in OLTP.

Corrosponding functions:-

- **query_generator (Date1,Load_date) //**
- **execution_engine (SQL$_A$) //**
- **query_converter (SQL$_C$) //**

```
Begin
SQLA query_generator (Date1,Load_date);
exec_flag← 'both';
Func execution_engine (SQLA);
SQLC query_generator (Load_date, Date2);
Func query_converter (SQLC);
End
```

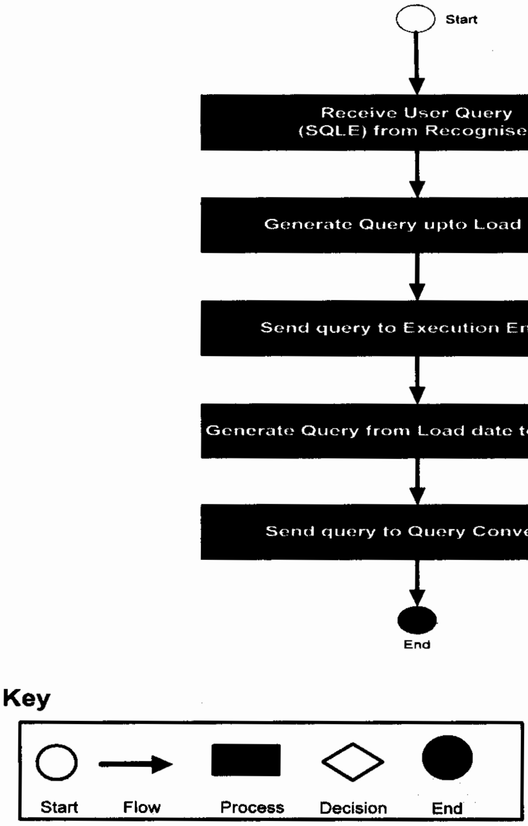

Start
Receive User Query (SQLE) from Recogniser
Generate Query upto Load date
Send query to Execution Engine
Generate Query from Load date to end date
Send query to Query Converter
End
Key
Start
Flow
Process
Decision
End

Figure 4.2 Activity Diagram for Query_Decomposer()

The decomposed OLAP query is than converted by the help of metadata. The mappings of table and columns are used for this process. Fig 4.3 shows an example. The converted query is then executed.

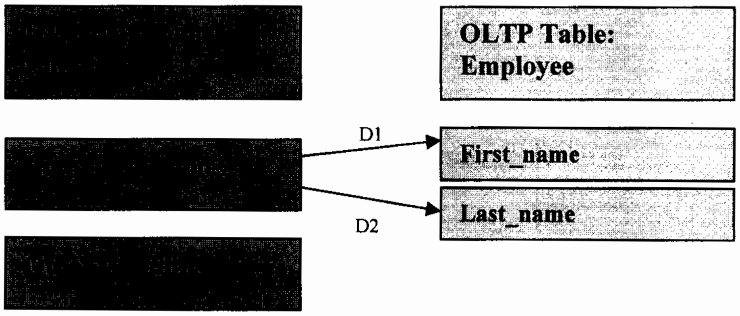


Figure 4.3 Column Mapping

4.7 Execution Engine

The execution engine sends the query to OLAP and DI. The type of data required will be checked. In case of both fresh and historical, the data of second query will be merged with first one. If there is query for fresh only or historical only than the results will be shown to user and no merging is required.

4.8 Application's Pseudocode

In order to depict that how different modules will interact with each other to perform simple application's operation, following pseducode can be used :-

```
Output_Result    Main    (input Query)
{
        Syntactic/semantic analyzer (Input Query)    // Check  Syntax of Query
If
              Error()
       then
                Display  Error Message  ()

Else
                QT =  Check Query Type ()

Switch QT

        Case  OLTP:

                Query_Converter()

        Case  OLTP+OLAP :

                Query_Decomposer()

        Default  OLAP:

                Output_Result    = Execute Engine ()
End
           Return Output_Result
}
```

4.9 Summary

In this chapter, the working of the proposed architecture is discussed by diagrams and algorithms. This chapter discusses detailed working of architecture by the help of algorithms for different components.

Chapter 5

RESULTS & ANALYSIS

5.1 Introduction

In this chapter the evaluation of the architecture is described. In this part the three queries will be formulated to evaluate the architecture and then these queries will be applied to check working of different components.

5.2 Testing Environment

A sample testing environment has been created to make evaluation process more strong. It will not only make analysis transparent but also enhances understanding of readers.

5.2.1 Testing Environment Description

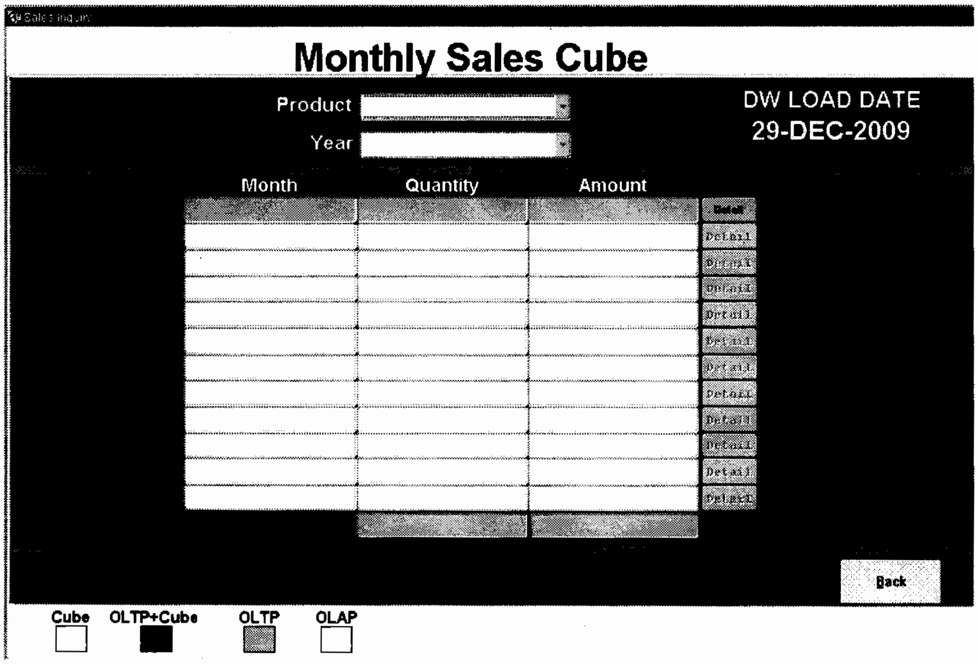
ABC is an IT equipment company. There is an OLTP system for routine sale activity and has a data warehouse. There is an OLAP cube for sales having data like sales of equipment. The dimensions for sales cube are customer, Product and Date. The loading of data warehouse processed after every 24 hours at night. The top management set the targets of sales after analysis on historical data and the lower management monitors their progress by checking current sale.

There is a need of fresh data at the right time for performance check and to meet the targets set by higher management.

Firstly the details of OLTP source system analyzed and then I implement DW using Oracle Warehouse Builder 10g realease2 and construct logical cube of sales. Then I develop a front end application in Oracle IDS 10g release2 for the implementation of the architecture. PL/SQL language is used for implementation of the different functions of proposed architecture. The description will cover the challenges faced during the implementation of these modules.The data used for testing and evaluation is extracted from Oracle Corporation Site. The data is about sales and the number of records in the main table of sales is about 1.9 million.

5.2.2 Query Processing

The Coordinator receives the query as in Figure 3.2. The Query is processed in order to run inquiry that what quantity of specified product has been sold in one specific year. Moreover how much amount is received against each transaction? There are various aspects for analysis i.e. in different dates and for different items.



Monthly Sales Cube

DW LOAD DATE
29-DEC-2009

Fig 5.1: Main GUI of Proposed System

Figure 5.1 is depicting main GUI of the system. In order to make it user friendly a key against each display color is mentioned. Following steps will be followed for normal processing.

- User will select product from 'Product's drop down list'.
- User will select year from 'drop down list of years'
- Record against each month that lies in this year, along with quantity sold and amount received will display in predefined colors.
- User can see details against each month.

After setting ground to initiate testing, it's time now to run tests. We have divide data encapsulated in query in three broad categories which are:-

- Historical Data
- Historical & Fresh Data
- Fresh Data

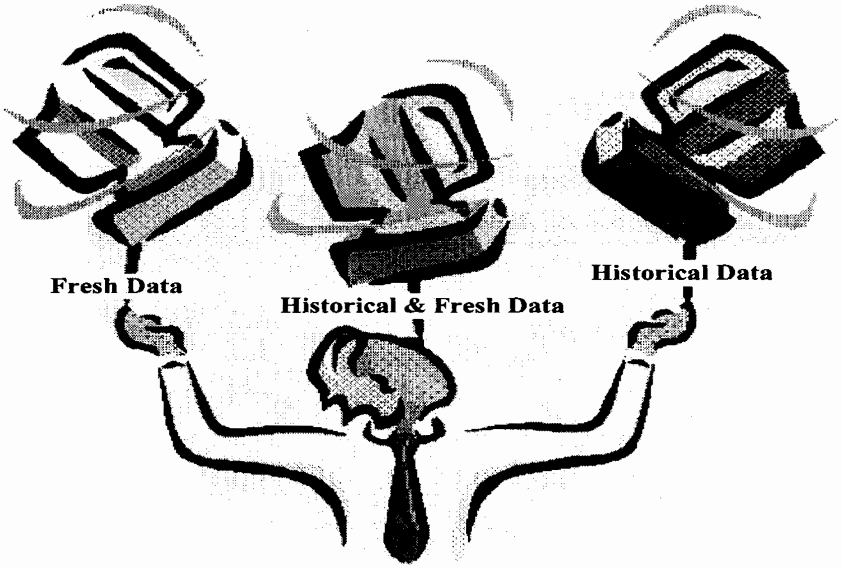


Fig 5.2: Broad Categories of User Query

5.2.3 Query for Historical Data

Quantity of IT equipment sold in all months 2008 by all channels and to all customers.

The processing is as follow:-

For this testing last upload date of DW is 29 Dec 2009

5.2.3.1 Syntactic and semantic analyzer

It checks the syntax first and then checks the attributes and relation names from global metadata.

5.2.3.2 Query recognizer

Perform following processes

- Checks that date exists in the user query
- From global metadata the data warehouse last load date is checked
- The date of the user query will be compared with load date
- The last load date in this case is greater
- So only historical data is queried
- The data from OLAP is required
- No conversion is required so invoke execution engine

5.2.3.3 Execution Engine

The query will be executed on OLAP by the execution engine and the results will be shown to user.

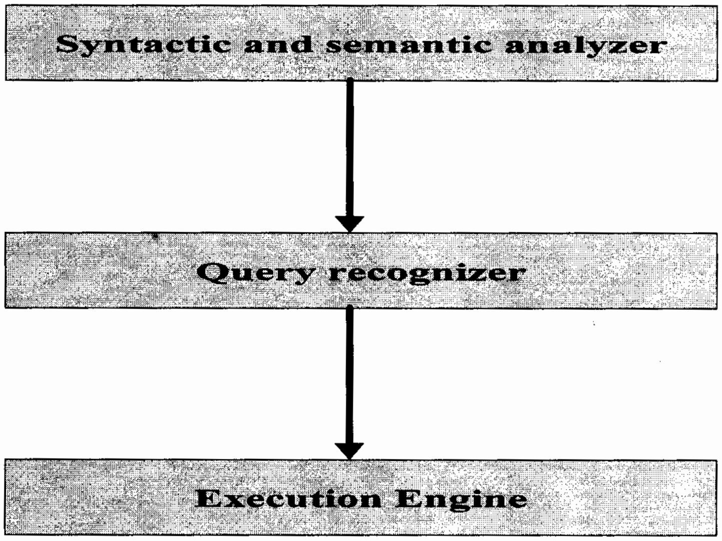
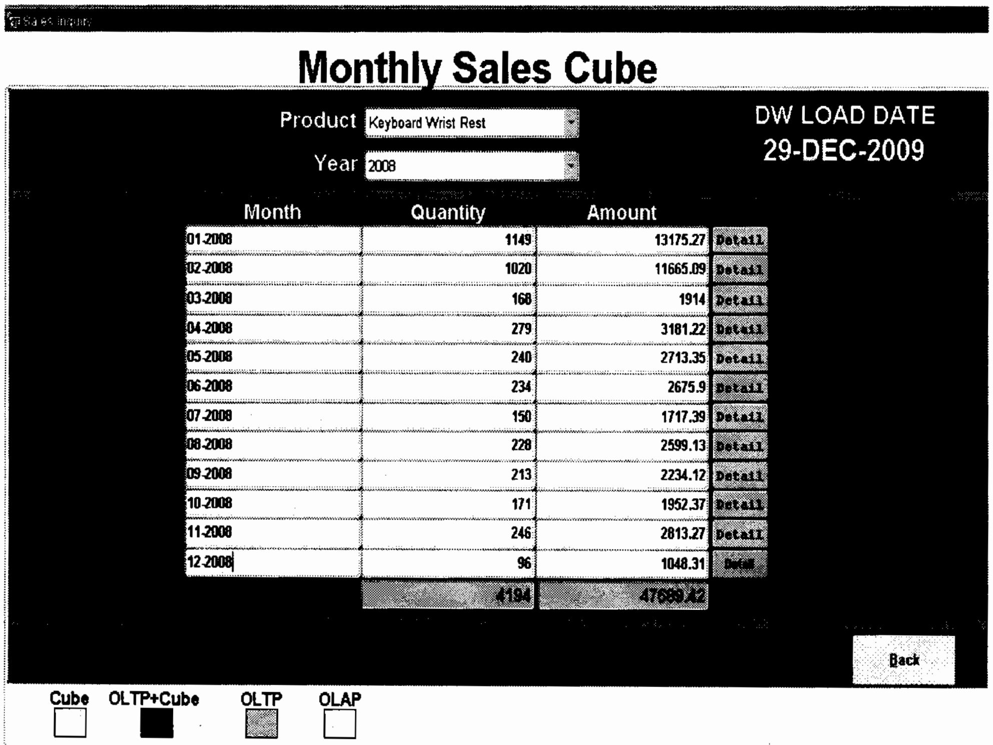


Fig 5.3: Flow of User Query for Historical Data

The executed GUI will be as shown in figure 5.4. The blue color is depicting that data is being fetched from CUBE.



Monthly Sales Cube

Product: Keyboard Wrist Rest

Year: 2008

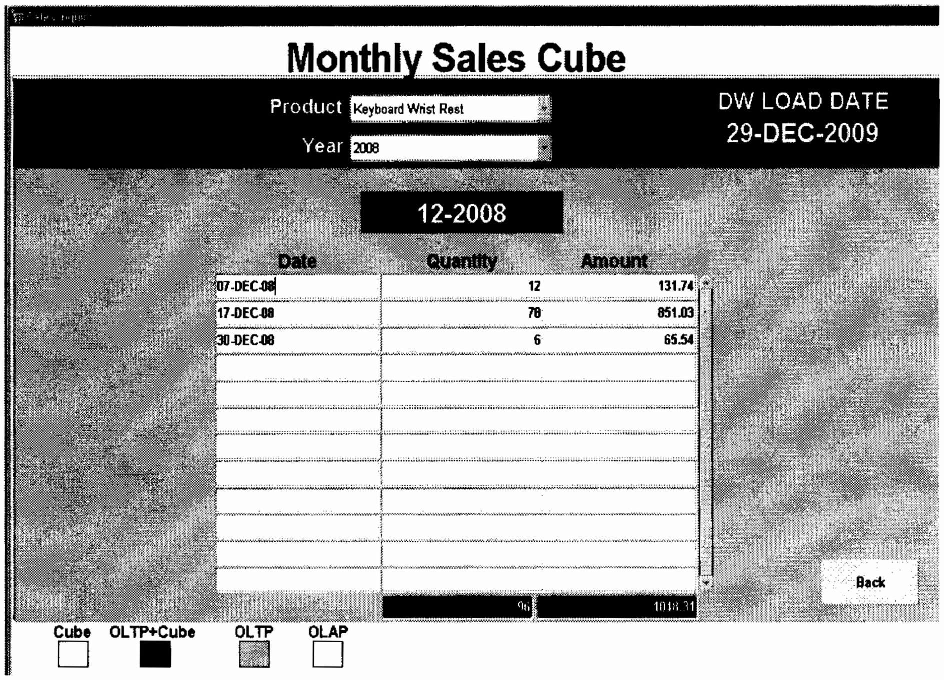
DW LOAD DATE
29-DEC-2009

Month	Quantity	Amount	
01-2008	1149	13175.27	Detail
02-2008	1020	11665.09	Detail
03-2008	168	1914	Detail
04-2008	279	3181.22	Detail
05-2008	240	2713.35	Detail
06-2008	234	2675.9	Detail
07-2008	150	1717.39	Detail
08-2008	228	2599.13	Detail
09-2008	213	2234.12	Detail
10-2008	171	1952.37	Detail
11-2008	246	2813.27	Detail
12-2008	96	1048.31	Detail
	4194	47689.42	

Cube □ OLTP+Cube ■ OLTP ▨ OLAP □

Fig 5.4: Query for Historical Data

If user is interested in going through details then the following screen (fig 5.5) will be displayed. It shows date wise sales data within selected month, on which transaction has been made.



Monthly Sales Cube

Product Keyboard Wrist Rest

Year 2008

DW LOAD DATE
29-DEC-2009

12-2008

Date	Quantity	Amount
07-DEC-08	12	131.74
17-DEC-08	78	851.03
30-DEC-08	6	65.54
	96	1048.31

Fig 5.5: Details of Query for Historical Data

5.2.4 Historical and Fresh Data Query

IT equipment sold in all months 2009 by all channels and to all customers. The processing is as follow:-

For this testing last upload date of DW is 29 Dec 2009.

5.2.4.1 Syntactic and semantic analyzer

It checks the syntax first and then checks the attributes and relation names from global metadata.

5.2.4.2 Query Recognizer

Perform following processes

- Checks that date exists in the user query

- From global metadata the data warehouse last load date is checked
- The date of the user query will be compared with load date
- The last load date in this case is between start and end dates of query.
- So both historical and fresh data is queried by user
- The data from DW and OLTP system is required
- Decomposition of query is required

5.2.4.3 Query Decomposer

It decomposes the query in two parts. The first query for OLAP system up to load date and the second is for the OLTP System after the load date. The decomposed query in two parts is shown below in Table 5.1.

Q 1 : OLAP Query	Q 2 : OLTP Query
Quantity of an item sold to all customers from 01 Jan 2009 to 29 Dec 2009	Quantity of an item sold to all customers from 30 Dec 2009 to 31 Dec 2009

Table 5.1: Decomposed query in two parts

Query 1 will be sent to the execution engine

Query 2 will be sent to the query converter

5.2.4.4 Query Converter

Query 2 will be converted to an OLTP query. Table 5.2 describes the details of mapping.

OLAP Query	Mappings
Quantity of an item sold to all customers from 30 Dec 2009 to 31 Dec 2009	• Quantity is mapped to sum (Qty_sold) • Product category is mapped to product name • Date attribute derives the Year

Table 5.2: Mapping Details

5.2.4.5 Execution Engine

OLAP and OLTP queries are received and executed by execution engine. The results of both queries will be merged and will be sent to user. Pederson [8] explained this process in detail.

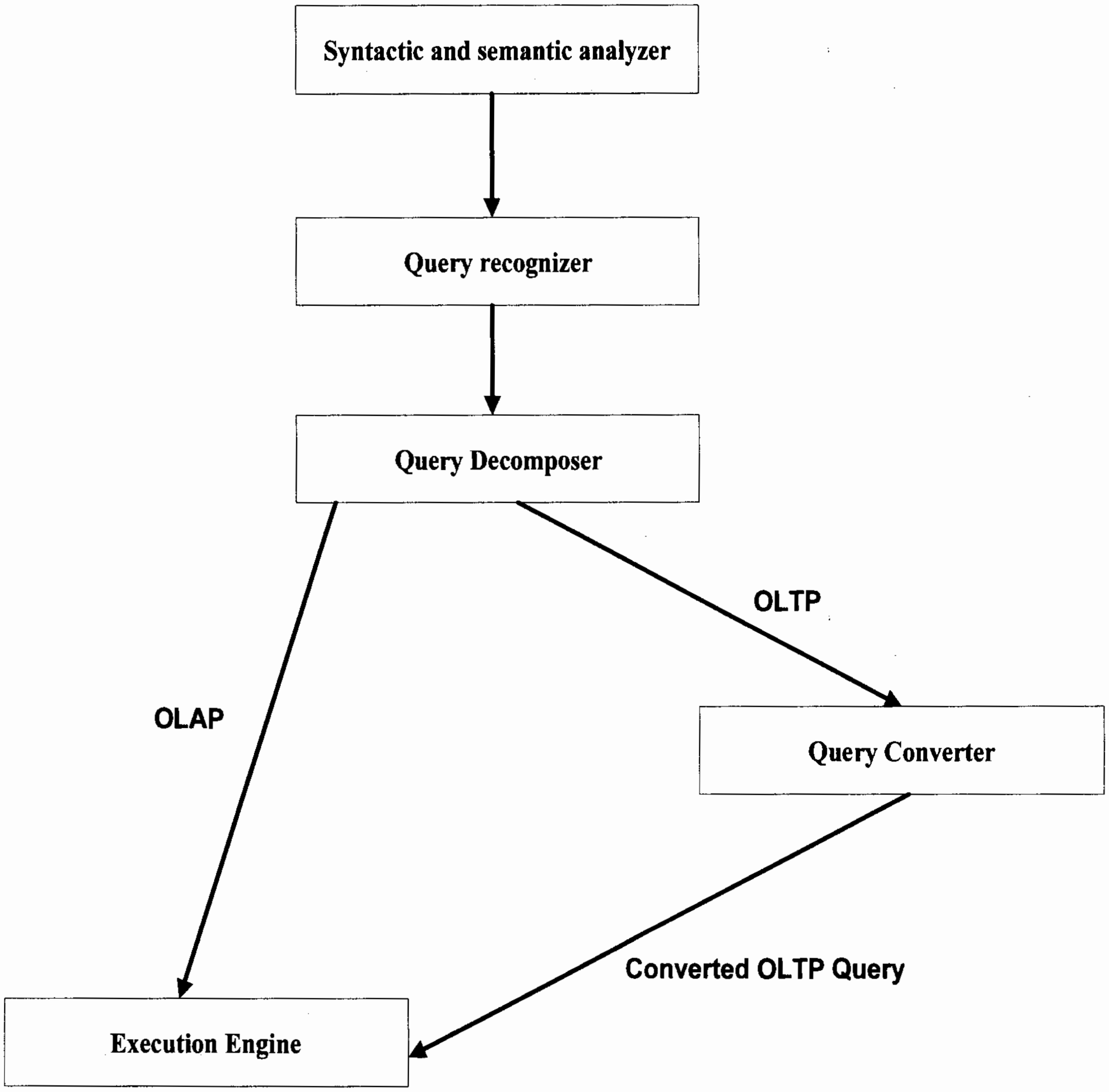
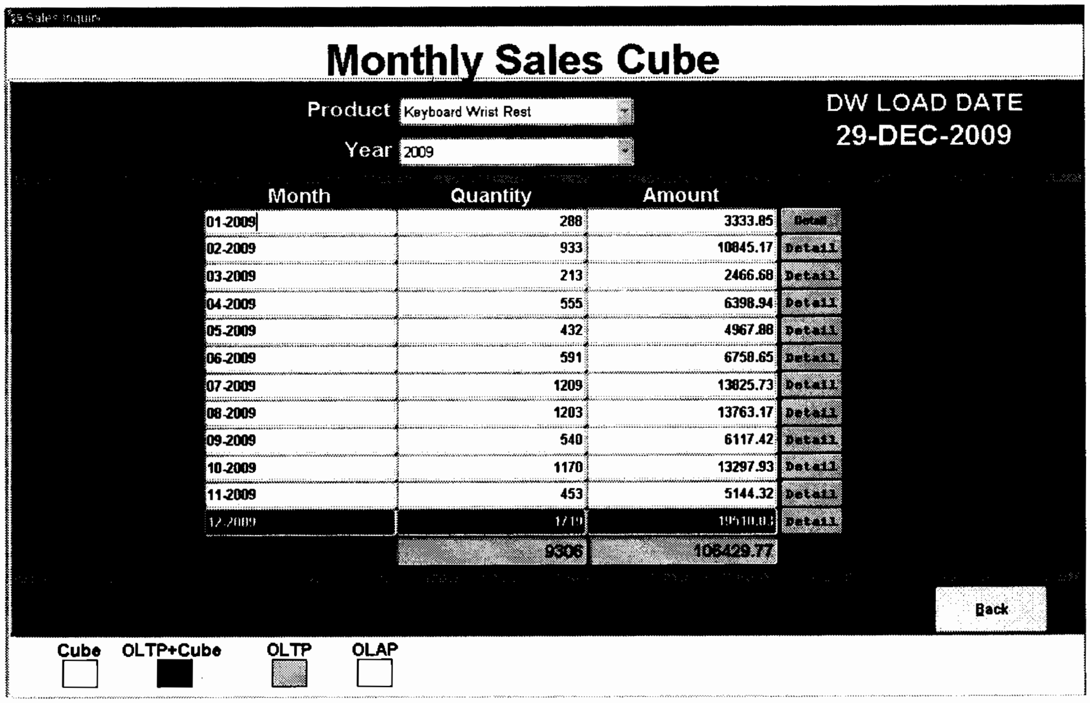


Fig 5.6: Flow of User Query for Historic and Fresh Data

The resultant GUI, after running this query is shown in figure 5.7.



Monthly Sales Cube

Product: Keyboard Wrist Rest

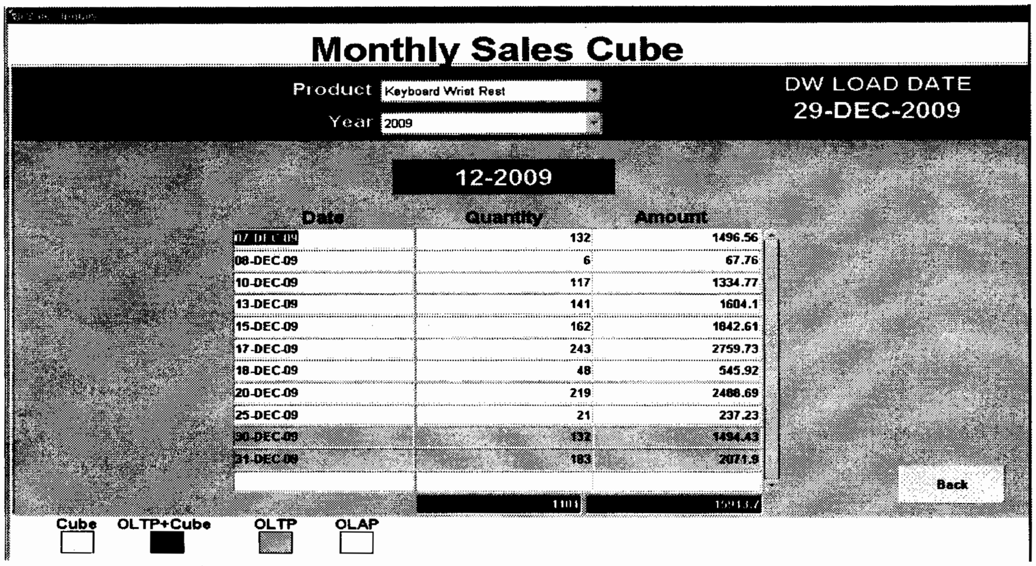
Year: 2009

DW LOAD DATE
29-DEC-2009

Month	Quantity	Amount	
01-2009	288	3333.85	Detail
02-2009	933	10845.17	Detail
03-2009	213	2466.68	Detail
04-2009	555	6398.94	Detail
05-2009	432	4967.88	Detail
06-2009	591	6758.65	Detail
07-2009	1209	13825.73	Detail
08-2009	1203	13763.17	Detail
09-2009	540	6117.42	Detail
10-2009	1170	13297.93	Detail
11-2009	453	5144.32	Detail
12-2009	1719	19510.83	Detail
	9306	108429.77	

Fig 5.7: Query for Historical and Fresh Data

If user is interested in going through details then the following screen (fig 5.8) will be displayed. Where each day within month, on which transaction has been made, will be displayed.



Monthly Sales Cube

Product: Keyboard Wrist Rest

Year: 2009

DW LOAD DATE
29-DEC-2009

12-2009

Date	Quantity	Amount
07-DEC-09	132	1496.56
08-DEC-09	6	67.76
10-DEC-09	117	1334.77
13-DEC-09	141	1604.1
15-DEC-09	162	1842.61
17-DEC-09	243	2759.73
18-DEC-09	48	545.92
20-DEC-09	219	2488.69
25-DEC-09	21	237.23
30-DEC-09	132	1494.43
31-DEC-09	183	2071.9
	1404	15943.7

Fig 5.8: Details of Query for Historical and Fresh Data

5.2.5 Fresh Data Query

IT equipment sold in all months 2010 by all channels and to all customers.

For this testing last upload date of DW is 29 Dec 2009

5.2.5.1 Syntactic and semantic analyzer

It checks the syntax first and then checks the attributes and relation names from global metadata.

5.2.5.2 Query Recognizer

Perform following processes

- Checks that date exists in the user query
- From global metadata the data warehouse last load date is checked
- The date of the user query will be compared with load date
- The last load date in this case is less
- So only fresh data is queried
- The data from OLTP system is required
- Conversion of query is required

5.2.5.3 Query Converter

The query will be converted to an OLTP query by applying mappings.

5.2.5.4 Execution Engine

The query will be sent to Data Integrator for execution on OLTP.

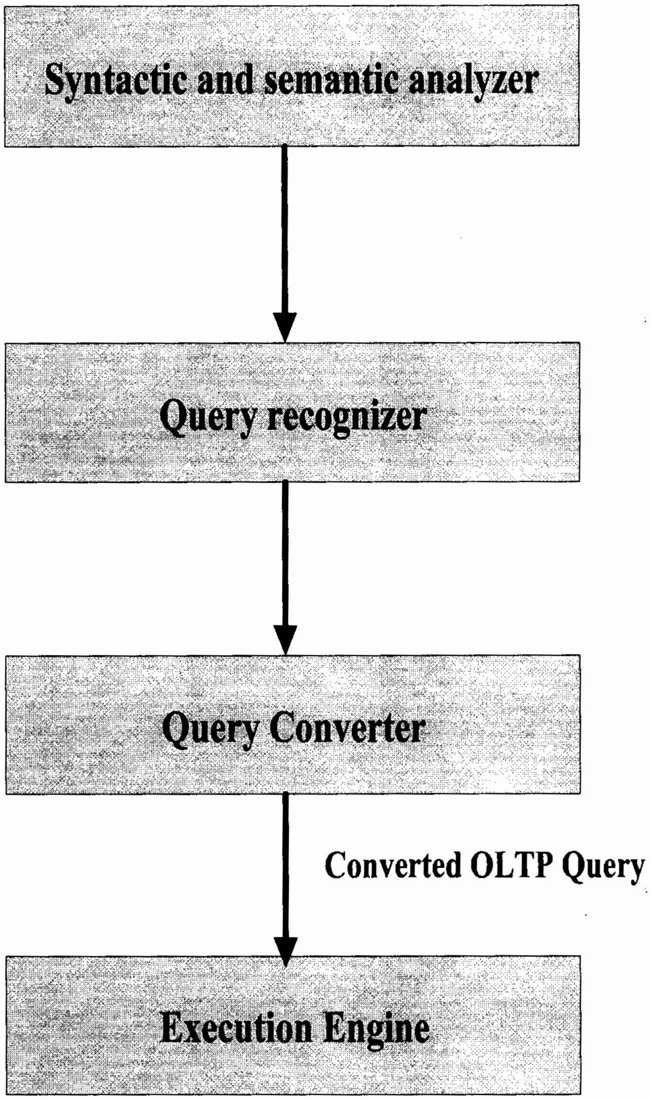
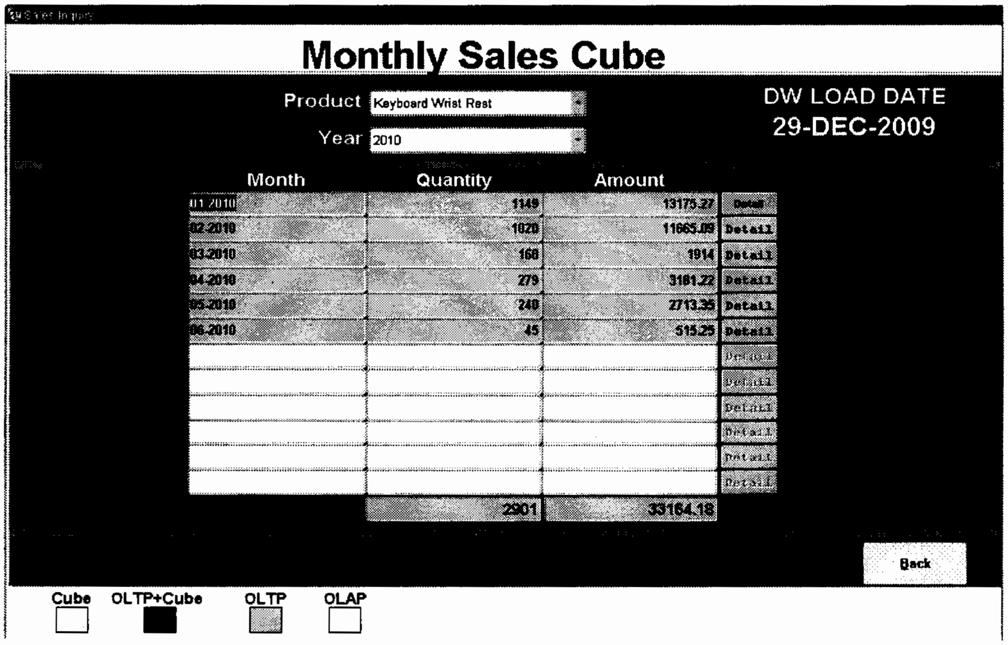


Fig 5.9: Flow of User Query for Fresh Data

The resultant GUI, after running this query is shown in figure 5.10.



Monthly Sales Cube

Product: Keyboard Wrist Rest

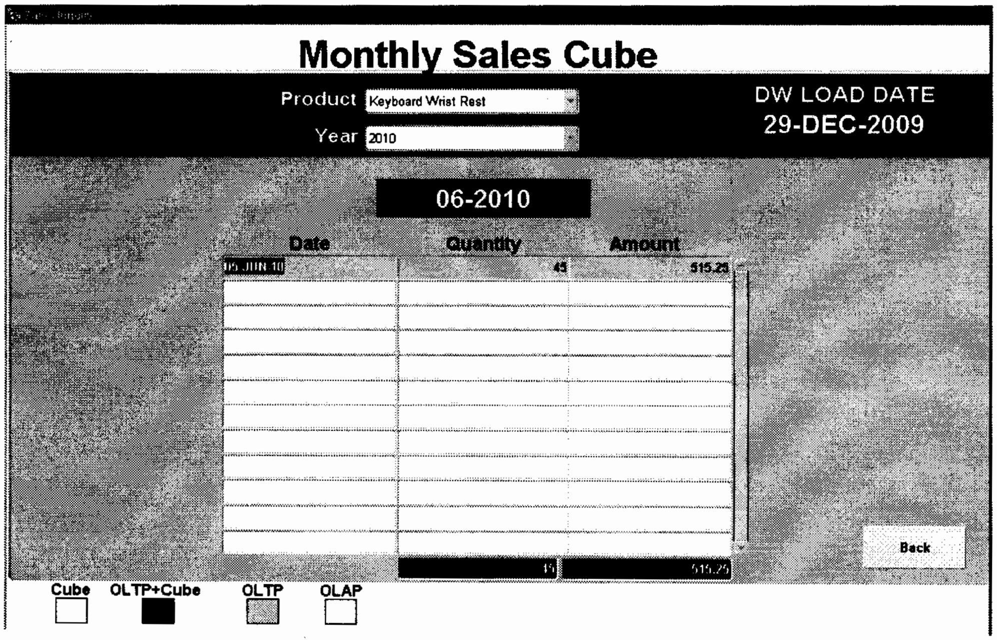
Year: 2010

DW LOAD DATE
29-DEC-2009

Month	Quantity	Amount	
01-2010	1149	13175.27	Detail
02-2010	1020	11665.09	Detail
03-2010	168	1914	Detail
04-2010	279	3181.22	Detail
05-2010	240	2713.35	Detail
06-2010	45	515.25	Detail
			Detail
			Detail
			Detail
			Detail
			Detail
			Detail
	2901	33164.18	

Fig 5.10: Query for Fresh Data

If user is interested in going through details then the following screen (fig 5.11) will be displayed. Where each day within selected month, on which transaction has been made, will be displayed.



Monthly Sales Cube

Product: Keyboard Wrist Rest

Year: 2010

DW LOAD DATE
29-DEC-2009

06-2010

Date	Quantity	Amount
05 JUN 10	45	515.25
	45	515.25

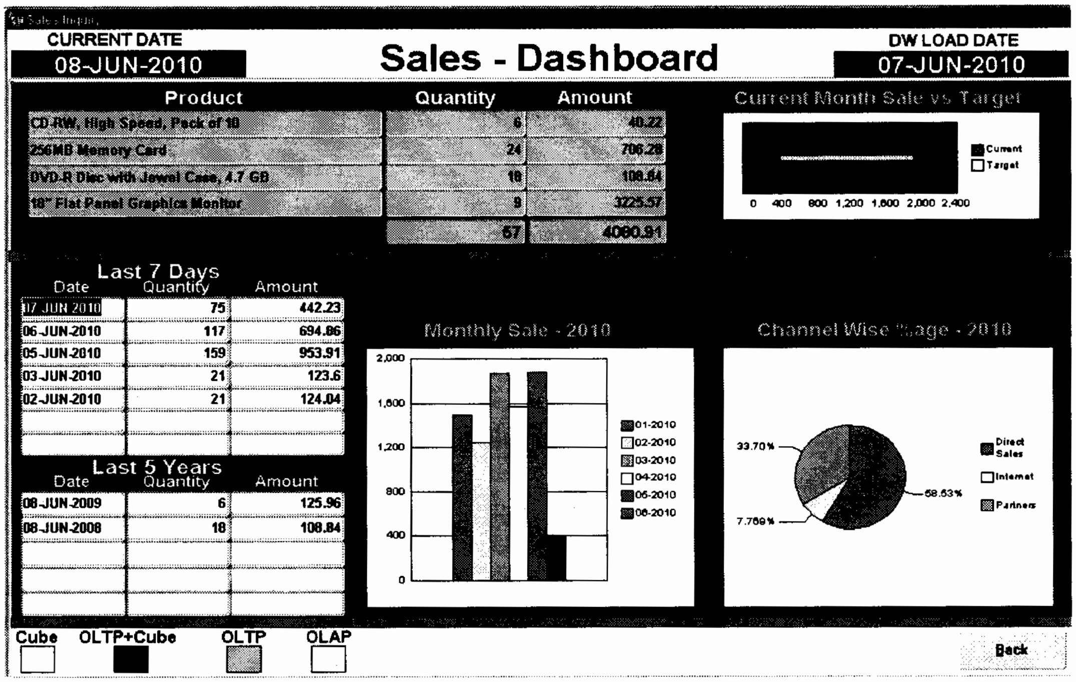
Fig 5.11: Details of Query for Fresh Data

5.3 Sales-Dashboard

Just to provide user more facilities to inquire and inspect different views generated against same query, Dashboard application is provided. This application has multiple plots on same screen, so that user gets familiar with different aspects of data against same products at same time. The important plots are as follow

- Transaction against each product along with quantity & amount details.
- Current month sale vs. Target
- Last 7 days transaction
- Monthly sale per annum
- Channel wise percentage on annual basis
- Last 5 years transaction on same date

Just for sake of understanding we have quoted two product's dashboard view. Figure 5.12 is the Sales dashboard snapshot against DVD-R disc with jewel case.



Sales - Dashboard

CURRENT DATE
08-JUN-2010

DW LOAD DATE
07-JUN-2010

Product	Quantity	Amount
CD-RW, High Speed, Pack of 10	6	40.22
256MB Memory Card	24	706.28
DVD-R Disc with Jewel Case, 4.7 GB	18	108.84
18" Flat Panel Graphics Monitor	9	3225.57
	57	4080.91

Last 7 Days

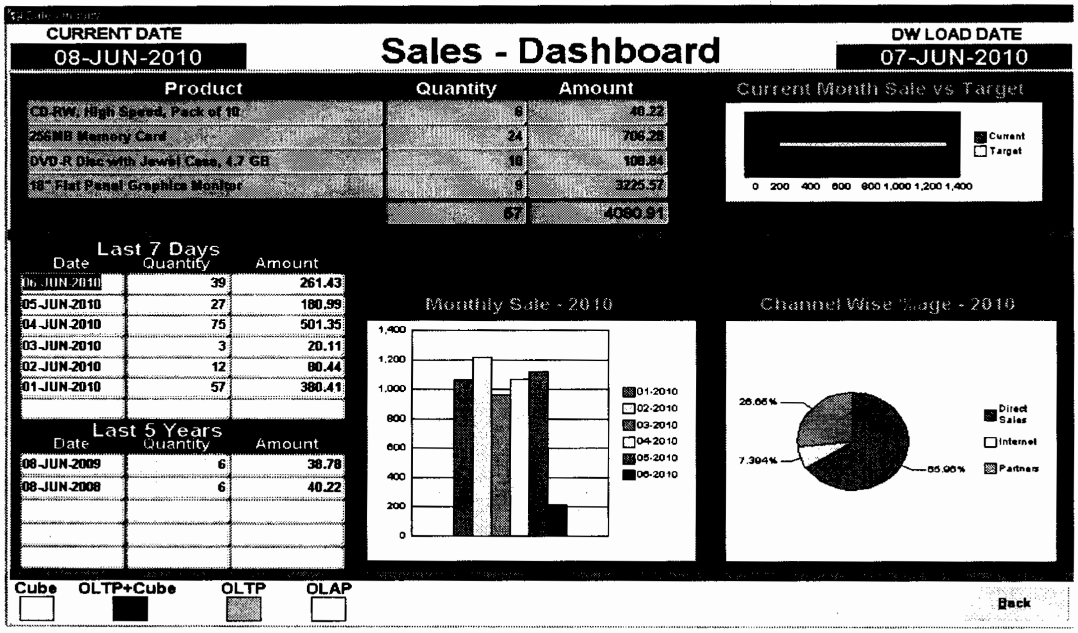
Date	Quantity	Amount
07-JUN-2010	75	442.23
06-JUN-2010	117	694.86
05-JUN-2010	159	953.91
03-JUN-2010	21	123.6
02-JUN-2010	21	124.04

Last 5 Years

Date	Quantity	Amount
08-JUN-2009	6	125.96
08-JUN-2008	18	108.84

Fig 5.12: Dashboard of DVD-R disc with jewel case

To have more emphasis we have run the same application against CD-RW High speed pack of 10 which is depicted in figure 5.13.



Sales - Dashboard

CURRENT DATE
08-JUN-2010

DW LOAD DATE
07-JUN-2010

Product	Quantity	Amount
CD-RW, High Speed, Pack of 10	6	40.22
256MB Memory Card	24	706.28
DVD-R Disc with Jewel Case, 4.7 GB	18	108.84
18" Flat Panel Graphics Monitor	9	3225.57
	57	4080.91

Last 7 Days

Date	Quantity	Amount
06-JUN-2010	39	261.43
05-JUN-2010	27	180.99
04-JUN-2010	75	501.35
03-JUN-2010	3	20.11
02-JUN-2010	12	80.44
01-JUN-2010	57	380.41

Last 5 Years

Date	Quantity	Amount
08-JUN-2009	6	38.78
08-JUN-2008	6	40.22

Fig 5.13: Dashboard of CD-RW High speed pack of 10

5.4 Result & Analysis

The proposed architecture has been designed & developed. System has been regressively tested under extreme conditions. The results were run against three criteria that are to test for historical, fresh & fresh and historical data. All the tests produced the expected results. Not a single test failed or goes unexpected. The high accuracy of results indicates the assurance of system precision against all possible combination of data within query.

5.5 **Summary**

This chapter provides the results of proposed technique when it is executed on against different datasets. The obtained results clearly show that adopted approach has been proved to be highly successful & effective.

Chapter 6

CONCLUSIONS

The data warehouse has already become a useful tool for analysis of business processes in different organizations. Its importance is already shown for helping in strategic level of decision making.

The new architecture proposed in this research is originated from the concept of three decision levels. This is an important concept but unfortunately the research work in this area is not much. The focus is on analysis of historical data. The fresh data is often required at lower level management to support decision making. The proposed architecture incorporates fresh data from OLTP system with the historical data, which improves the decision making.

Managers at tactical level can compare current progress with the historical data and target value of current operation which is decided at strategic level. If any task is not progressed well, the mangers can take better decisions in accordance with strategic decisions.

The tested components of different architectures are assembled to offers unique functionality which provide access to OLTP systems of an organization.

The decision capability to all decision levels of an organization increased. All the components are already tested which is an indication to produce the desired and valid results.

Chapter 7

FUTURE WORK

The incorporation of fresh data at real time is a new concept so the proposed architecture can be improved by future work. The important aspect of data warehouse solutions is performance. So the future work includes the improvement in query performance by development of query optimization techniques.

To make the proposed architecture more valuable, there should be real time data incorporation from different operational systems and other data warehouses.

The other important future work for proposed architecture is the implementation.

REFERENCES

[1] S. Chaudhuri, U. Dayal, "*An Overview of Data Warehousing and OLAP Technology*", ACM Sigmod Record, vol 26, 1997.

[2] W. H. Inmon, "*Building the Data warehouse*", 3rd Ed, Wiley 2002.

[3] R. Kimball, M. Ross, "*The Data Warehouse Toolkit*", 2nd Ed, Wiley 2002.

[4] P. Ponnaiah, "*Data warehousing fundamentals*", 3rd Ed, Wiley 2002.

[5] A. Agarwal, "*Data Warehousing*", Retrieved Feb 02, 2009, from the World Wide Web: http://toostep.com/insight/data-warehousing

[6] S. Terr, "*Real*-Time Data Warehousing: Real-Time Data Warehousing 101", Information Management Online. Retrieved Feb 02, 2009, from the World Wide Web: http://www.information-management.com/infodirect/20040423/1002228-1.html 2003.

[7] M. Golfarelli, S. Rizzi, I. Cella,"*Beyond Data Warehousing: What's Next in Business Intelligence?*", DOLAP'04, Washington, DC, USA, 2004.

[8] D. Pedersen, K. Riis, T. B. Pedersen, "*Query Optimization for OLAP-XML Federations*", DOLAP '02, McLean, VA, 2002.

[9] S. S. Conn, "*OLTP and OLAP Data Integration: A Review of Feasible Implementation Methods and Architectures for Real Time Data Analysis*", IEEE SoutheastCon 2005, Ft. Lauderdale, Florida, 2005.

[10] C. Bontempo, G. Zagelow, "The *IBM Data Warehouse Architecture*", Communications of the ACM, vol. 41, 1998.

[11] P. Gray, and H. J. Watson, "*Present and Future Directions in Data Warehousing*", ACM SIGMIS Database, vol. 29, 1998.

[12] K. Jones, "*An Introduction to Data Warehousing: What Are the Implications for the Network?*", International Journal of Network Management, vol. 8, 1998.

[13] M.K.Shahzad, "*Version-Manager: for Vital Simulating advantages in Data Warehouse*", Computing and Information System Journal, vol. 11, 2007.

[14] A. Eisenberg, and J. Melton, "*SQL Standardization: The Next Steps*", ACM SIGMOD Record, vol. 29, 2000.

[15] E. Riedel, C. Faloutsos, G. R. Ganger, and D. F. Nagle, *"Data Mining on an OLTP System (Nearly) for Free"*, ACM SIGMOD, International Conference on Management of Data, Dallas, Texas, USA, 2000.

[16] Davenport, T. H., and L. Prusak, "*Working Knowledge: How Organizations Manage What They Know"*. Boston, MA: Harvard Business School Press, 1998.

[17] H. Thomas, and A. Datta, *"A Conceptual Model and Algebra for On-Line Analytical Processing in Decision Support Databases*", Information Systems Research, vol. 12, 2001.

[18] N. Roussolpoulos, "*Materialized Views and Data Warehouses*", ACM SIGMOD Record, vol. 27, 1998.

[19] E. Schallehn, K. U. Sattler, and G. Saake, "*Advanced Grouping and Aggregation for Data Integration*", CIKM 2001, Atlanta, Georgia, 2001.

[20] J. W. Buzydlowski, I. Y. Song, and L. Hassell, "*A Framework for Object-Oriented On-Line Analytic Processing*", DOLAP '98, Washington DC, USA, 1998.

[21] M.T. Ozsu, P. Valduriez, "*Principles of Distributed Database Systems*", 2nd Ed, Prentice Hall, 1999.

[22] A. Vaduva, K.R. Dittrich, "Metadata Management for Data Warehousing: Between Vision and Reality", IDEAS 2001, Grenoble, France, 2001.

[23] M.P. Reddy, B.E. Prasad, P.G. Reddy, A.Gupta, "*A Methodology for Integration of Heterogeneous Databases*", IEEE Transactions on Knowledge and Data Engineering, vol 6, 1994.

[24] D.I. Sandu, *"Operational and real-time Business Intelligence"*, Informatica Economica Journal, vol 12, 2008.

[25] C. Thomsen, T. B. Pedersen, W. Lehner, *"RiTE: Providing On-Demand Data for Right-Time Data Warehousing"*, ICDE 2008, Cancun, Mexico, 2008.

[26] U. Dayal, M. Castellanos, A. Simitsis, K. Wilkinson, *"Data Integration Flows for Business Intelligence",* EDBT 2009, Saint Petersburg, Russia, 2009.

[27] S. Radeschütz, F. Niedermann, W. Bischoff, *"BIAEditor - Matching Process and Operational Data for a Business Impact Analysis",* EDBT 2010, Lausanne, Switzerland, 2010.