Chapter 01

Introduction of Data Warehousing

1.1 Introduction

The different Government departments as well as different level of organization are understanding characteristics of implementing a data warehouse for their departments. The data warehouse gives a tremendous facility for transforming the groups of data into the desired software and getting different answers to their query into the making decision for the organization. Any data warehouse gives the facility for analysis of data into multidimensional analysis as compare to traditional process and reporting. The data warehouse also provides the result of different query in easier access for making decision to the executive level of the organization.

The data warehouse is a desired tools or application software for different purpose such as extracting, cleaning and delivering information into the multidimensional way and also provides the features to fire different query as well as analysis of the information for the making decision for the organization. The DW can store different types of row data for uploading into the system and finally taking the decision for the organization. Now a day we can store the different types of data into the data warehouse software such as text, picture, as well as movie for the business purpose.

“Now a day this simple concept becomes a multibillion-dollar industry, and both practitioners and academicians believe that data warehousing applications are up-to-the-minute weapons for decision-making initiatives” by Hisami (2000).
Kimball (1996) defined “A data warehouse is a copy of transaction data specifically structured for query and analysis”. According to Hwang (2004) “An application collects daily transaction-oriented enterprise data both internally and externally and then accumulates, categorizes, and stores huge historical data for further analysis, prediction and discovery of data pattern”. They added: “Those data are more related to statistics, non-modified and stored in warehouse in a long-term manner, also they are time-oriented, integrated and can be used effectively for analyzing and decision-making” Hwang et. al (2004).

I assume the definition of Inman (1996), who is known as the father of data warehousing, from his famous book “Building the data warehouse”; “A data warehouse is a subject-oriented, integrated, non-volatile and time-variant collection of data in support of management decisions”.

A closer look will be taken at each of the above-mentioned key features in Inman’s definition in the data warehousing concepts and characteristics section.

To taking appropriate decision for the any organization the data must be accurate. It is very decisive that the organization collects the good business data and stores it for taking flexible decision by top level management. The top level management can be used data warehousing tool for the enterprise to data analysis, decision making process etc from these data.

The data access point is incredibly portion of any data warehouse in products of DW and the managers must give some attention in bring to multidimensional data mode. The data access point can used for accepting data from the traditional database management system into the enterprise as graphical user interface to data analysis and decision making for the organization. This system or module analysis the desired data from the row data into the multidimensional data to at least 75 percent in their effort, expends of the project
1.1.1 Short History of Data Warehouse

In early 1980 we can say the origin of the data warehousing in the world, when the Relational Database Management System (RDBMS) is commercially used in the big organization. The easy use of Relational Database Management System and implementation of query in Structural Query Language (SQL) is the main cause of end user support and growing of this system. To help out of the end user into computing, extracting data of the organization from the traditional database and store into the new database and also help them to different query, reporting for the different level of the enterprise. The Relational Database Management System also provide the good performance of end user computing as well as different operation such as creating, updating, deleting, and produced reports for the different department of the organization.

In the beginning days the data warehouse used the subsets of operational data for extracting for these data. The end user uploaded all subsets in their system in regular basis such as weekly, monthly. The end users used subset for limited number for that for accessing different query and reporting tools.

The beginning days of data warehouses have different purpose and role in the organization which is already exist and still growing rapidly. When we compare early day’s data warehouse and current days of data warehouse they provide great support in decision making process. Now data warehouse is not a database system that supports end user for queering function and not subsets of operational system. Now data warehouse is a new collection of information which is used by the different department of organization and top level executive for data analysis and decision making. When you have plan of develop data warehouse you must applied group of modeling technique and also discussed with different department of the organization.

The data warehouse can be design as per the requirement of organization and also it is tool which provides the feature to the executive level manager for get accurate and timely information for decision making.
When you design the data warehouse it can involved all activities which make possible to the organization for implement all functions.

### 1.1.2 Multi Dimensional Modeling

Multidimensional Modeling gets its name from the idea of dimensions. Dimensions are categories of categories. The multidimensional modeling is the method by which using of modeling constructs in the world for develop multidimensional model.

![Figure: 1.1 Example of Star Schema](image)

The basically organization could be purchase the data warehouse because it is matter of amount and price of product. The classically dimension of data warehouse can be place of product, and time of product, as well as time of the product. The DBA can fire the queries to measure different values ranges and also used aggregate function for generate report such as total product in the particular month.

#### 1.1.2.1 Features of a Multidimensional Model

There are following features of multidimensional model of data warehousing, such as:
a. Explicit separation of structure and contents: - It is basic need of model which makes the difference of schema. The schema shows the complete structure of data and their instance.

b. Explicit hierarchies in dimensions: - The multidimensional mode is basically design in hierarchy into the level for the grouping of data with dimension.

c. Multiple hierarchies in each dimension: - When we design the multidimensional model then we can combine more paths in one for accessing aggregate data from the system and make the relationship between different dimensions.

d. Measures sets: - In multidimensional model we can define structure of complex cell which is same as fact or cube.

e. Symmetrical treatment of dimensions and measures:- In multidimensional model we can measures in dimensions and vice versa. This is use full in measurement of concept and also group different facts so we can use it for register factual data into different direction.

1.1.3 Data Warehouse Architectures

The structure of data warehouse is consideration as per the requirement of organization, data security, data modeling as well as features of basic requirements. It can also be based on Meta management, area of warehouse staging, bandwidth and technology which is used in the design of data warehouse of the organization.

The selection of architecture of data warehouse should be based on location or place of reside of it and data marts as well as control resides. The organization can finalized about the data, it can be reside in central server which is manage centrally or it can be reside in locally which is managed by individually.
In this research I can suggest of data warehouse architecture can be independent, it should be global, and it should be interconnected to each other. I can also suggest regarding implementation of architecture of data warehouse, it can be top down, it can be bottom up, and some time we can combine of top down and bottom up approach. We can also combine the design and implementation of data warehouse architecture for the organization. For example, architecture could be logically distributed, managed independently, and implemented from the top down approach.

The architecture of other system and data warehouse system could be same but they do not have same features. There are different types of data warehouse architecture for the organization such as, single-layer architecture and N-layer architecture. We can differentiate with each other by number of middleware of OS and Analytical tools. We implement high level architecture as well as parts of architectures in the system/module.
1.1.3.1 Single-Layer Architecture

It is very simple and implementation of single layer architecture is also simple. In the single layer of architecture we can create physical data warehouse or any data mart between the database and tools. In single layer middleware could be used a virtual data warehouse those support the software layer. But it can not support database layer of the data warehouse. The single layer model store lots of data, and it is light weight, but we can separate between analytical and processing. In this layer analysis can be done directly on the operational data.

1.1.3.2 Three-Layer Architecture

The source layer, reconciled layer and data warehouse layer can be combining into the three layer architecture. In the source layer we can combine multiple source systems, in data warehouse layer we can combine data warehouse and data marts and reconciled layer sits between the source data and data warehouse. We can send the data from source system by ETL process and also store analyzed data in it for future process by ETL. We can clean our data in reconciled layer and combine it send to source system for further processing. The three layer architecture can use for big organization system. This architecture requires more data storage space for the further processing.

1.1.3.3 Multi-tier Architecture

The multi-tier infrastructure is the way to access and share information and data from any where of the data warehouse for the organization. The multi-tier infrastructure of the CWC Pune is shown in Figure 1.3 below.

Before explaining the multi-tier infrastructure in detail, let us show how users access the multi-tier infrastructure in CWC Pune.

1.1.3.3.1 How Do Users Access the System

It is based on web based system so if your want to access any information from system then they can access all information or data from web browser such as
Netscape, internet explorer simulation system is web-based, so users will access the system through their web browser, including Netscape and Internet Explorer with the help of well-known protocol such as Hyper Text Transfer Protocol (HTTP). Let us see the different steps of end user to access the information and data from the internet.

Figure 1.3. The multi-tier infrastructure of CWC Pune

1. The end user point of view they can inter the following address into the browser to access the home page:

   http://cwc.cse.in.edu:8000/nom/homepage.jsp.

2. If the user is a new end user, they have to registers with the current system and existing end users can skip this step.
3. They logs in the current existing system, provides environment variables with the help of Molecule Editor Wizard.

4. After that they invoke the system after finishing the wizard.

5. After they can show the reports page on the web browser specially according the current information exist for the organization.

6. The all information will be sent by data mining server according the end user requirement for the organization.

1.1.3.3.2 How Does the Infrastructure Work

The new model can be write with the help of core java with the help of Swarm library. All data and information stored in master database which is generated by this system. They provide different types and different level of reports for the end users. The more datasets can be summarized, extracted, transformed and reloaded into data warehouse for different data mining technique for the CWC organization and also this information can be available for all end user of the world.

Application Server

The Java 2 Platform, Enterprise Edition (J2EE) is the best standard for develop and implement multi-tier enterprise for any CWC organization. They also supports Enterprise Java Beans (EJB), Java Servlets, Java Server Pages (JSP) and XML technology for develop user friendly application such as window based application for the organization. The J2EE platform can be used for publish JSP pages for client server technology and call in session of Java Servlets on web for the end users. We select OC4J (Oracle9iAS container for J2EE) for access information from Oracle and make a link between java and oracle and also provide all information to the end user. They also support to develop web interface with the help of JSP and Java Servlets. The end users can download OC4J Server from http://www.oracle.com.

Currently, CWC Pune have no network but in feature they can install two Linux or windows based machines, cwc.cse.in.edu and cwcadm.cse.in.edu, which is used for access all information from the application servers.
The new system can run either on cwc or on cwcadm, they can invoke by Java Servlets on cwc and cwcadm respectively. When data or information become very large then they can stored all information and data into Oracle databases.

Our system can be uses the Swarm library and this is installed on both cwc and cwcadm system. Swarm can be freely downloaded from http://www.swarm.org. We are using Oracle to store dataset which is generated by the current system; we can also use Oracle Database Server and Data Mining Server for the same. When the system will implemented the data is inserted into the Oracle using Java Database Connection (JDBC).

To store and extract information and data while system is running, we have to create Online Transactional Processing (OLTP) databases for the organization. Since our system will generate a large set of data and information, we have designed database for fast insertion and storage for the end users. We have also create a data warehouse which can stores row data, summarized, transformed and aggregated data from those OLTP databases and also provide all information should be available any time for the end users. Our basic aim of the data warehouse system to access information very fast, accurate and also provide all information to the all users. We will develop new model or system for the CWC Pune chapter.

Currently, we don’t have any database system specially oracle in CWC Pune, they have stored all information in Excel application according to requirement of file and data. These information we can upload into the oracle based system. The oracle supports all library of java API so we can design and develop data warehouse according to end user requirement. In the chapter 06 we discuss more about new system or model for the CWC Pune.

*Oracle Reports Server*

When we develop the new system for the CWC Pune the all department of organization is able to view all types of information and data and also other user
which is outside of CWC, they can also view the all information, reports and other data for there used. We can create different types of report as well as different level such as Top level, middle level and lower level for the end users. The all information will be access by web browser. These types of reports generated by the Oracle Reports tool in JSP format. The query of end user for report generation can be handling by oracle databases with the help of Transparent Network Substrate (TNS) protocol. The reports server is combined with an OC4J server as a Java Servlets.

1.1.3.3.3 Load Balance and Fail Over

As shown in Figure 1.3 above shown the different application servers on which is used for invoked and run the current system. We also have different database servers which can store extracted and row data for the organization. To effeteely used the new model we can used web interface for distribute and all information in different application server, so end user will be access all information and data from any where of the world. We also add code of oracle program such that new system can stored and extract all data form the system.

1.1.3.3.4 Load Balance the Application Servers

The idea of used oracle 9i application servers is straight forward: we don't want our users to use Excel for store and extract information for the server or database. So we use the round robin method to load and extract information or data among the two application servers as in the following JSP code:

```jsp
<%! private static int access=0; %>
<%
    if ((++access)%2==0)
    {
        <a href="http://cwc.cse.in.edu:8000/nom/runnom?user_id=10">Invoke</a>
    }
    <%
}
Else
{

```
11
In the above JSP code, we declare a static variable `access` because in every load of the web page will be increment our variable by 1. If access is even, the system will run on cwc, otherwise it will run on cwcadm. In this way, if multiple systems are invoked, half of them will run on cwc and the other half on cwcadm.

### 1.1.3.5 Load Balance the Database Servers

The basic aim of load balancing database servers is the same as balance the application servers for accessing information from the different types of servers. The following Java code which is shown below from our system program shows us how database servers are load balanced, where sessionid is the identifier for the current system invoked by a user.

```java
int sid=(sessionid)%3+1;
String url=null;
if(sid==1)
    url="jdbc:oracle:thin:@foyt.cwc.in.edu:1521:simu2";
else if (sid==2)
    url="jdbc:oracle:thin:@symphony.cselab.nd.edu:1521:etech";
else
    url="jdbc:oracle:thin:@bigband.cselab.nd.edu:1521:mynom";
DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
Connection conn=DriverManager.getConnection(url, "username", "password");
```

When the end user execute the following statement one problem will arises because the reports pages will be pointed after invoking the system and also the
reports pages require to know which database will be accessed to get the information and data for generate the different reports.

To manage this problem we can create the table called nom sessions with different attributes such as session id, user id, sid and status. In this table session id is the identifier for the access the session for the system user id is the identifier of the end users and sid is used for database for states all information for the particular sessions and status specify the status of current session.

There are three different results or values of the status such as executing, terminating, and terminated. First time when the system is executed the status is executing, when end user select the terminate session the user will get the reports with some summary of report such as page, status to changes.

The system will check the status at the beginning of every step of extracting or querying on database. When the system find the status is terminating the program will change the status to terminate. The process will checked by Java Servlet with communication of process. The Java does not support to kill the process so it very hard to kill the process with help of kill command in java. Java is platform independent language and they do not support the features of pid in some platforms. This process is similar to capability proceed by Java Message Services (JMS).

After executing the system, the report will generate with the help of two parameters such as user id and session sid. The end users can generate the report in every session when they invoked it.

The table 1.3 below show different sessions of the centralized database which is different from the databases which is store in the master database. The reports page required the filed _rst to retrieve information of data from session table and also specify the database stores in the system.
The following Core Java code shows the procedure to load balance the database servers.

```java
Connection conn=DriverManager.getConnection(url, user, passwd);
Statement stmt=conn.createStatement();
ResultSet rset=stmt.executeQuery("select session_id.nextval from dual");
while(rset.next())
{
    sessionid=rset.getInt(1);
    int sid=(sessionid%3)+1;
    stmt.executeUpdate("insert into nom_sessions values ("+
                          sessionid+","+StartMolecule.userId+","+sid+")");
    String nom_url=null;
    switch(sid)
    {
        case 1:
            nom_url="jdbc:oracle:thin:@foyt.cse.nd.edu:1521:simu 2";
            break;
        case 2:
            nom_url="jdbc:oracle:thin:@bigband.cselab.nd.edu:1521:mynom";
            break;
    }
}
```
break;
case 3:
    nom_url="jdbc:oracle:thin:@symphony.cselab.nd.edu:1521:ete
    ch";
    break;
    default:
    break;
}
}

In this statement the three different database servers are used in a round robin method. According to the condition they pass the value such as if value of sid is 1, data will be stored on foyt; if value of sid is 2, data will be stored on bigband; and if value of sid is 3, data will be stored on other database. In this statement you can see the multiple function is executed and database servers will be load balanced.

1.1.3.3.6 Fail Over

When end users user multiple application servers and database servers then they want to reduce the down time for the new developed model. The basic aim of fail over is that if system finds some node which is evolved in system down then it send information to another peer node with failed over message. For example, if the system is planned to run on joy and store all information and data on foyt, but if there is error encountered, then our program will select another database to provide the information or data, until a connection is successful.

This feature really increases the availability of the whole function of developed system. We can do some quantitative measure of the improvement of the system; let's assume that each system has a probability $p$ of down time. If fail over is not utilized, the possibility that a system can not run easily is $1−(1−p)^2=2p−p^2$. If fail over is utilized, the probability that a system will not run is reduced to $1−(1−p^2)(1−p^3)=p^2+p^3−p^5$. 

15
For example, let's assume that $p=0.01$, then the down probability is reduced from 0.0199 to 0.0001.

You can have different way to implement the fail over to another database. When you try to make a connection with database and you are notable to connect it then you will get runtime exception which is thrown by the system. We just catch the system generated value through catch statement in system and try to connect to another database, until we succeed.

The fail over to another application server which can be implemented by web JSP page which is provide the facility to make a links to java Servlet in the program. When system invoked by Servlet successfully a success message will be sent by the JSP and also JSP will redirect to java Servlet to another server.

### 1.1.4 Design of Data Warehouse

As we have shown in Figure 1.2 above, in some possible architecture of Data Warehouse, a Data Warehouse can be implemented by SQL or by OLAP.

There are two different approach of relational data warehouse such as dimensional modeling techniques and other is materialized views. Both can be used in design and implementation in large organization.

We can use cube structure for data representation in dimensional model which support compatible logical data representation in OLAP management. There are different objectives of dimensional modeling, such as:

a) To develop structure of database this is used by users and implements any query command.

b) To maximization of effective used of SQL queries by minimization of no of table as well as relationship between them.
When we used dimensional modeling in data warehouse out concentration always with facts, dimensions and measures. The fact can be defining as a collection of related data items which is combination of measures and context data. We can define dimension is nothing it is collection of data which always show one business dimension. And it determines the contextual background for the facts. In measure we can always think about numeric attributes of fact, different behavior of dimension etc.

When you use relational data warehouse then you can considering basic models which is used in dimensional modeling such as star model and snowflake model. In the star model we can used basic structure for dimensional model where one large central table which is called fact table and also set of smaller table which is called dimension and both are arrange in radial pattern. (We can show an example in Figure 1.3 below).

![Figure: 1.4 Star Models](image)

In the snowflake model we can decompose result in one of more than one of the dimensions. It support many-to-one relationship between sets of different attributes of the dimension and can be separated into the new table which is in hierarchy. (Figure 1.4 below shows an example).
The basic features of structure of snowflakes are to visualize the hierarchical structure of dimension but other models can be used alternatives design such as flat, terraced, and star cluster etc.

We can used use star schema approach in design techniques and methods as a practically. The executive level of staff can take the decision about present schema for the big organization, they can use pattern oriented approach for use in corporate data model in the data warehouse or they can also used traditional entity relationship model of data warehouse for the organization.

The top level manager also considers the approach of materialized view in the data warehouse design. The will be facing the following problem when he designs the data warehouse such as:

a) Total space is available for materialization.
b) The able to provide the answer of group of query for display records.

c) The cost of use nested query evaluation and maintenance of database.

The basically when the executive level of management want to design data warehousing application software they consider only on techniques of specific sub models and design style of desired domain areas. But for this purpose they have to valuable knowledge in data warehouse design and their practical implementation into the organization.

The present work intends to abstract and structure Data Warehouse design techniques and strategies in the schema transformation primitives.

1.1.5 Implementation of Data Warehouse

We can develop a prototype of a Data Warehouse Design tool, called DWDesigner that can be combined with other components conforming a CASE tool.

There are following components of data warehouse, such as: (1) CMDM (Conceptual Multidimensional Data Model) (2) A Repository Manager for a CASE tool and (3) From the conceptual schema to the logic schema.

Figure 1.5 below presents an overview of the architecture of the whole CASE environment.
Figure 1.5 Over all View of Architecture of Data warehouse

DW Designer is a tool that implements the principal ideas of this thesis (Data Warehouse logical design): transformation primitives, transformation trace, schema invariants and consistency rules. Implementation of Data Warehouse evolution, whose ideas were presented in this thesis.

1.1.5.1 The Prototype

The prototype is a Data Warehouse Design tool with the following main characteristics:

- It allows the designer to design a Data Warehouse schema starting form a source schema by means of Primitive applications.
- It generates a trace of the transformation applied.
- It provides invariants checking.
- It includes consistency rules triggering.
- It has a graphical user interface.
• It is extensible. Its modular design allows adding and removing primitives without modifying or recompiling the existing code.

1.1.5.2 Functional Features

The tool is intended to be a Data Warehouse design graphical environment. It should be useful to a Data Warehouse designer who has a source database, some Data Warehouse requirements, and wants to generate a Data Warehouse schema that satisfies these two elements. With this tool the designer can apply different Data Warehouse design criteria and techniques in order to obtain the target Data Warehouse schema. Invariants checking can be invoked at any moment in the design process. It allows checking the consistency of the schema that is being generated. Consistency rules are triggered by the application of certain primitives to certain elements; when these applications put in danger the consistency of the schema that is being generated. The graphical interface facilitates interaction with the source database elements, application of primitives and parameters selection, and visualization of the design trace.

1.1.5.3 Conceptual Design

For the design of the tool we applied object-oriented techniques. The design language we used was UML. The core of the tool is the Virtual Machine. This is the most important layer of the system. The main components of the Virtual Machine are the following:

• Tools for schema transformation: This is the representation of the transformation primitives, and two sets of schema elements (called Repository and Data warehouse) that are the containers used during the transformation process.

• Database concepts representation: All schema elements, relations, attributes, keys, etc. must be represented in order to be manipulated by the user and the primitives.

• Trace: Each applied primitive, with its input and output relations, is recorded in the trace.
• **Invariants:** The invariants are properties that must be satisfied by the schema. Therefore we represented the invariants as a part of the schema representation.

• **Consistency Rules:** We represented the Rules as an independent entity, which is referenced and invoked from different places.

### 1.1.5.4 Implementation

The prototype was implemented in JAVA language, using the Java Development Kit version 1.2.2 and Borland’s JBuilder version 2 as the development environment.

Implementation details of the tool, excepting the parts of Invariants and Rules, can be used in data warehouse. We can implement by different Invariants Checking and Rules Triggering.

Invariants checking are an option of the tool’s main menu that allows checking the consistency of the existing Data Warehouse schema. The user has the possibility of choosing between lists of invariants. We implemented the procedures that perform the Invariants Checking as methods of the *General Schema* class. When the user press *OK* button, we call the methods of the *General Schema* class that correspond to the invariants selected by the user.

Rules are implemented as an abstract class *Rule* and a sub-class *RuleXX* for each existing rule (analogous to the primitives). We defined the *Rule Directory*, which is a sequence containing the existing rules and is initialized at start. Also at this moment, some primitives initialize the *rules* attribute that is a set of rules. The rules that belong to a primitive’s set of rules are the ones that must be triggered after the primitive application. When the user applies certain primitives, a dialog box appears showing him what rules should be applied in the form of check boxes. The rules that are checked by the user are applied automatically. The transformations that are made by the rules, are applications of primitives, therefore they have the same behavior as any primitive application.
DW Designer is a prototype of a Data Warehouse Design Tool. Invariants checking and Rule Triggering functionalities. The developed tool implements the principal ideas of this thesis (Data Warehouse logical design) transformation primitive, transformation trace, schema invariants and consistency rules. This tool offers a graphical user interface that allows the designer to apply primitives to a source schema, constructing a new schema, visualize the generated transformation trace, check schema invariants and apply consistency rules.

The tool can be connected with other modules, complementing each other. Altogether, they constitute a CASE tool for designing a DW that covers the stages of: conceptual modeling, derivation of a logical model, management of the logical model, and persistency of the design.

1.2 Online Analytical Processing

An OLAP application is a tool for getting not only for raw data but also information by the end user. It helps out to the end user to identify the style or trends of desired data in very quickly with accessing the lots of raw data in sequentially. Basically the online analytical processing provide the facility to end user to get the different answer for them such as, “how are our communication is done by month of June 2012 in north Maharashtra?”. We can also use OLAP in different field such as forecasting of sales and marketing, data mining.

We can difference ate our OLAP application and OLTP application in different methods such as store data, analysis stored data, as well as way to display reports to the end user. The basic difference of OALP application is that it provides very sophisticated and complicated business oriented question for the small and big organization.

The OLAP always store their data in such as way that more than one user can work with data in parallel. It avoids storing data such as traditional relational model. It is always used to work with aggregate data with help of aggregate function for access quickly data.
1.2.1 Storing the Data in OLAP

The OLAP is a tool to provide the facility to more than one end user to work individual at a time. The OLAP used three methods to store the data in cube format, such as ROLAP, MOLAP, and HOLAP. All are used cube methods to store physical data into them.

1.2.2 ROLAP

The ROLAP stands for "Relational OLAP". In this type of OLAP they store all the data in cube format and both base and relational database. It hides the table or database and shows all data only in cube format. They used SQL query statements for combinations of base data and aggregate data with each other to provide the appropriate result and also maintaining tables and database in proper order for generate reports of different level of the organization.

1.2.3 MOLAP

The MOLAP stands for “Multidimensional OLAP”. In this method we can store all cube data in two ways namely base and high level with help of multidimensional data files. It provides the facility to extract the base data from the desired table into the particular multidimensional format and evaluates it. It also gives the quick access the data from the multidimensional data and always fit into RAM. It used large disk space for store hues data in it. But it also store duplicate of main data and also copy into the cube.

1.2.4 HOLAP

The HOLAP stands for “Hybrid OLAP”. It can be used to store high level of data with proper priority into multidimensional database. But it can not change in original data of base database. It removes the duplicate data from the base database and also save the time and space of the secondary storage device. But it require robust link between the multidimensional database and relational database.
1.3 Introduction of Cube

In the all OLAP application the cube can be used as conceptual design for different purpose such as data store, data retrieval etc. But we can store data may be in different methods because cube can be store all data into the logical design. We can understand the storage method of cube by the help of the table given below.

Table 1.1 Sales Records from Transactional Database

<table>
<thead>
<tr>
<th>Description</th>
<th>Location</th>
<th>Quantities</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Down Side</td>
<td>25</td>
</tr>
<tr>
<td>CPU</td>
<td>Middle Side</td>
<td>27</td>
</tr>
<tr>
<td>CPU</td>
<td>Upper Side</td>
<td>32</td>
</tr>
<tr>
<td>Mother Board</td>
<td>Down Side</td>
<td>31</td>
</tr>
<tr>
<td>Mother Board</td>
<td>Middle Side</td>
<td>25</td>
</tr>
<tr>
<td>Mother Board</td>
<td>Upper Side</td>
<td>45</td>
</tr>
<tr>
<td>CD</td>
<td>Down Side</td>
<td>152</td>
</tr>
<tr>
<td>CD</td>
<td>Middle Side</td>
<td>215</td>
</tr>
<tr>
<td>CD</td>
<td>Upper Side</td>
<td>157</td>
</tr>
</tbody>
</table>

Table 1.1 above shows sales records from three electrical stores which is displayed into the transactional table. There are two field columns “Description” and “Location” that contain information about record and a third value column “Quantities”. The above table is also called fact table of the database.

In the cube table we can define the identifiers from the field into the table. In the cube each axis is show the dimension, the horizontal dimension is description and vertical dimension shows location and it is called location dimension.
1.4 Introduction of Data Mart

When the top level management has decide to develop data warehouse for the organization they consider data mart as physical store of group data and operational data for making decisions for the organization with the help of data analysis, previous work and results. We can combine the specific subset of enterprise data for the analysis and generate different level of reports in the data marts. The data mart provide the facility to simplicity and rapid development for the decision making as well as quickly returns of total investment for the organization.

We can define data mart is a subset of data warehouse which is used for specific business or organization as well as institution. For the storage and function point of view different department of the organization considered that they have owner of data mart such as hardware, software as well as raw data of the organization. The data mart we help to the department of organization for use, management of data and development of data without changing raw data into inside of other data mart and also into data warehouse.

Basically data mart used for highly focused on single subject area instead of whole organization with the help of data warehouse. The data mart also provides the facility to change everything into the raw data and information by controlling of all process such as time for development of data mart and also money for development. It reduces the development cost and time for development for the very large organization. The data mart also reduces the risk in delivers information on time for large scale data warehouse.

The main function of data mart is basically used in business intelligence application (BI). We can used BI is for collection, storage, extracting, and analyze of raw data of the organization. We can also used BI in to the small to very large scale of the organization. As compare to data warehouse the data mart is very less expensive which is making the cost effective for the different level of the organization. The Table 1.2 below summarizes the key differences between data warehouses and data marts.
Table 1.2 Differences between Data Warehouse and Data Mart

<table>
<thead>
<tr>
<th>Data Warehouse (DW)</th>
<th>Data Mart (DM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>• We can include all subject area in it.</td>
<td>• We can include only Single subject area in it.</td>
</tr>
<tr>
<td>• The WD is used in broad scope for the different level of Organization.</td>
<td>• The DM used for focused on scope for the different level of Organization.</td>
</tr>
<tr>
<td>• In DW we can analysis of performance in Yearly.</td>
<td>• The DM performance can be analysis in monthly.</td>
</tr>
<tr>
<td>• We expense millions in development of DW.</td>
<td>• We expense thousands in development of DM.</td>
</tr>
</tbody>
</table>

The data mart can also be set up within less than 90 days for any organization which is very less time compare to data warehouse that take more time to set up into the enterprise. In real life all small organization used BI application because of it low cost as well as less time for setup of the data mart which provide the facility to suitable method or technique of data storage for the organization.

Finally we can take the advantage of data mart such as mitigate the risk, less expense and time for data management for the data warehouse functionality. For the scalable of data mart it can be used for all level of organization and complexity of the enterprise. And it is easy to understand for any employee of the organization and it support mechanism for fast delivering, and capable of decision making for the all level of the organization.

1.4.1 Characteristics of Data Mart

For the small scale and very large scale of organization there are following characteristics of data mart.
• It provides the facility of access data in very frequently when the different department of organization needs it.

• It can be used group of end users for collective view of report of the enterprise.

• The data mart can improves the response time for the different end users.

• The cost of development of data mart as well as implementation of it is very low compared to data warehouse.

• The data mart can be defined very clearly by the users compared to full data warehouse.

1.4.2 Creating Data Mart

The organization used data mart for access all information in easy way, they can used star schema for their design which provide the facility in relational database to do analytical function with the multidimensional database for the any scale of organization.

1.4.2.1 Define Data Mart Requirement

Before creating the desired data mart the different level of management has to define the basic requirement of data mart. The table 1.3 below defines the requirement of data mart for the different users when they are make a plan for design the data mart, such as:
Table 1.3 Basic Requirements of Data Mart

<table>
<thead>
<tr>
<th>End User</th>
<th>Requirements of Data Mart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Analyst/Project Data Analyst</td>
<td>(A) They can identify the basic requirement of the organization for data mart. They can also define the requirement specification of data mart, such as:</td>
</tr>
<tr>
<td></td>
<td>1. Types of users;</td>
</tr>
<tr>
<td></td>
<td>2. Location of users specifically Geographic;</td>
</tr>
<tr>
<td></td>
<td>3. Requirement of data for the user;</td>
</tr>
<tr>
<td></td>
<td>4. Requirement of data security.</td>
</tr>
<tr>
<td>Project Data Analyst</td>
<td>(B) They can identify and design the model as well as metadata of the source data. He can include an Enterprise Data Model (EDM) in this system.</td>
</tr>
<tr>
<td>Project Manager, Business Owner</td>
<td>(C) They can do the review on different document which is identify for the data requirement.</td>
</tr>
<tr>
<td></td>
<td>(D) Finally after all activities are over they can sing off of requirement of data document.</td>
</tr>
</tbody>
</table>

1.4.2.2 Create and Test of Developed Data Mart

After define the basic requirement of data mart the different level of manager and staff of organization has to create and test the developed data mart on that organization.

Table 1.4 Basic Activities of Data Mart

<table>
<thead>
<tr>
<th>End Users</th>
<th>Activities of Data Mart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Administrator</td>
<td>(A) The DBA can use DDL for data mart with help of physical data model.</td>
</tr>
<tr>
<td></td>
<td>(B) They can implement dimensional data mart in to the DBMS and also inform to top level manager.</td>
</tr>
<tr>
<td>Project Data Analyst</td>
<td>(C) He can work with subject experts to develop logical view of multidimensional model.</td>
</tr>
</tbody>
</table>
1.3 Problem on Hand

Before development of data warehouse for any organization there are many issues adjoining of data warehouse that can be remember by designer of particular data warehouse for the organization. If designer of data warehouse is neglect the key issued of data warehouse in the beginning that is cause of data warehouse project are unsuccessful for any organization.

The organization may use data warehouse for database management, decision making and report generation of different department. But there any problem that organization can face in its implementation and utilization.

- Every organization think about the data warehouse and make a concept about it; it is very big thing for the organization without study of basic requirement of the organization where they use the DW.

- Without thinking about core business tool for the organization they make the decision about data warehouse. The value of DW only is politics where they may implement within organization. The employee does not understand of use of data warehouse because of its complexity and they do not have training on data warehouse software. It is the most important challenge for any level of organization.

- Before taking any effort to analysis the data warehouse feature for the organization before buy of it. They do not study the basic requirement of organization specially requirement of different department. So organization will waste their money, employee development and also time of the organization and employee.

- They organization also do the blind faith on software vendors that they promise to develop the easy software and they can work on it very easily. Basically the organizations face the problem in implementation of it.
1.4 Importance and Scope of the Project

The basic goal of this research work is to provide a tool that allows designing a data warehouse which is started from source database and propagating source schema evolution to the data warehouse.

*It is proposed that we are to explore the following areas in detail.*

- To measure the organization performance through multidimensional attributes. Each dimension and its associated factors are designed to predict the employee’s behavior.
- To predict Central Warehouse Corporation preference in certain fields of study.
- To study the understanding, prediction and prevention of the import export failure of the Central Warehouse Corporation.
- We propose a model which makes prediction about increase or decrees ratio of import and export based on quality of performance as well as system inform to the Top Level Management about the ratio of import and export. The proposed model also deals with entrance ratio of goods in a particular department as well as sub-department and exit ratio after successful completion of quality test.

In future we can take into consideration varied segments of infrastructures facility across various departments and try to find unidentified pattern in their performances using Data warehousing models which can help to predict unknown outcomes. The reports which will be generated in future will serve mainly by management and compare with different department over time in performances as may be affected by the different predictors that are available plus other well chosen variables.

We concentrate on the data warehouse design problem through a schema transformation approach. We propose more than one schema transformation primitives, which is used in high-level operations that transform relational sub-schemas into other relational sub-schema.
We also provide some tools that help in data warehouse design process. The first is the design trace, which is generated when a data warehouse schema is constructed by application of primitives. The second is a set of schema invariants. It is useful to check data warehouse schema consistency. By the schema invariants we make available a set of rules that specify how to correct schema-inconsistency situations which is generated by applications of primitives.

Finally, we make available some strategies for designing the data warehouse by application of primitives for solving some common data warehouse design problem.

We can divide propagating source schema evolutions problem of data warehouse schema into two phases:-

- Determination of the change and trace that must be done to the data warehouse schema.
- Basic application of evolution to the data warehouse.

For solving phase one we use the transformation trace that was generated in the design. This trace allows us to obtain the path that was followed by each schema element and then decide how to propagate the changes occurred on the source schema. In some cases it is not necessary to modify the data warehouse schema, but we always have to modify the trace in order to maintain the connection between source and data warehouse schema elements. We provide a set of propagation rules that state which changes have to be done to the data warehouse and to the trace, depending on each case of source schema change and dependency between source and data warehouse schema elements.

In order to solve phase two we analyze the applicability of existing schema evolution models and techniques to data warehouse schema evolution. We consider data warehouse features that affect the treatment of evolution.