INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

(Volume1, Issue 2)

Implementation of Data Marts in Data ware house

Neha Malhotra

nehamalhotra588@gmail.com

Abstract: A data mart is a persistent physical store of operational and aggregated data statistically processed data that supports businesspeople in making decisions based primarily on analyses of past activities and results. A data mart contains a predefined subset of enterprise data organized for rapid analysis and reporting. Data warehousing has come into being because the file structure of the large mainframe core business systems is inimical to information retrieval. The purpose of the data warehouse is to combine core business and data from other sources in a format that facilitates reporting and decision support. In just a few years, data warehouses have evolved from large, centralized data repositories to subject specific, but independent, data marts and now to dependent marts that load data from a central repository of Data Staging files that has previously extracted data from the institution's operational business systems (e.g., student record, finance and human resource systems, etc.).

I. INTRODUCTION

To understand what a data mart is, we must first know a little about data warehouses. The concept of data warehousing first appeared in the late 1980s in articles published by Bill Inmon and others. Essentially, a data warehouse involves a systematic approach to gathering, organizing, and storing data–generally from internal production sources but also from external providers–to create a definitive source of information for analysis and decision support [1]. A data mart is basically a condensed and more focused version of a data warehouse that reflects the regulations and process specifications of each business unit within an organization. Each data mart is dedicated to a specific business function or region. This subset of data may span across many or all of an enterprise's functional subject areas [2].

II. DEPENDENT AND INDEPENDENT DATAMARTS

There are two basic types of data marts: dependent and independent. The categorization is based primarily on the data source that feeds the data mart [3]. Dependent data marts draw data from a central data warehouse that has already been created. Independent data marts, in contrast, are standalone systems built by drawing data directly from operational or external sources of data, or both.

The main difference between independent and dependent data marts is how you populate the data mart; that is, how you get data out of the sources and into the data mart [4]. This step, called the Extraction-Transformation-and Loading (ETL) process, involves moving data from operational systems, filtering it, and loading it into the data mart.

With dependent data marts, this process is somewhat simplified because formatted and summarized (clean) data has already been loaded into the central data warehouse [5]. The ETL process for dependent data marts is mostly a

process of identifying the right subset of data relevant to the chosen data mart subject and moving a copy of it, perhaps in a summarized form.

With independent data marts, however, you must deal with all aspects of the ETL process, much as you do with a central data warehouse. The number of sources is likely to be fewer and the amount of data associated with the data mart is less than the warehouse, given your focus on a single subject [3].

The motivations behind the creation of these two types of data marts are also typically different. Dependent data marts are usually built to achieve improved performance and availability, better control, and lower telecommunication costs resulting from local access of data relevant to a specific department. The creation of independent data marts is often driven by the need to have a solution within a shorter time [6].

III. DIFFERENCE BETWEEN DATA WAREHOUSING AND DATA MART

It is important to note that there are huge differences between these two tools though they may serve same purpose. Firstly, data mart contains programs, data, software and hardware of a specific department of a company. There can be separate data marts for finance, sales, production or marketing. All these data marts are different but they can be coordinated. Data mart of one department is different from data mart of another department, and though indexed, this system is not suitable for a huge data base as it is designed to meet the requirements of a particular department. Data Warehousing is not limited to a particular department and it represents the database of a complete organization. The data stored in data warehouse is more detailed though indexing is light as it has to store huge amounts of information. It is also difficult to manage and takes a long time to process. It implies then that data marts are quick and easy to use, as they make use of small amounts of data. Data warehousing is also more expensive because of the same reason.

IV. FUNCTIONAL OVERVIEW

While it is recognized that Data Warehouse systems each have their own unique characteristics, there are certain generic characteristics shared by the family of systems known as Data Warehouses. These include:

1. The prime purpose of a Data Warehouse is to store, in one system, data and information that originates from multiple applications within, or across, organizations. The data may be stored 'as received' from the source application, or it may be processed upon input to validate, translate, aggregate or derive new data/information.
2. Most of the data load functions are processed in batch. There are few on-line data maintenance functions. The on-line functions that do exist tend to update the reference files and data translation tables.
3. A database alone does not constitute a Data Warehouse system. At a minimum, a Data Warehouse system must include the database and corresponding load functions. Data reporting functions are optional. They may or may not be an integral part of the Data Warehouse system.
4. The prime purpose of storing the data is to support the information reporting requirements of an organization, i.e. multiple users and multiple applications.
5. The Data Warehouse may or may not provide the required reporting functions. In some cases, external applications access the Data Warehouse files to generate their own reports and queries.
6. Data Warehouse functions are often based upon packaged software. An example is the Business Objects product.
7. Where the data within the Warehouse supports the reporting requirements of multiple applications and users, the data may be physically stored based upon the user requirements. Separate database segments may store the 'user views' for a particular user. This results in the physical storage of a considerable amount of redundant data. The data storage approach is designed to optimize data/information retrieval.

V. CONTRASTING OLTP AND DATA WAREHOUSING ENVIRONMENTS

Figure 1-1 illustrates key differences between an OLTP system and a data warehouse. One major difference between the types of system is that data warehouses are not usually in third normal form (3NF), a type of data normalization common in OLTP environments. Data warehouses and OLTP systems have very different requirements. Here are some examples of differences between typical data warehouses and OLTP systems:

- Workload

  Data warehouses are designed to accommodate ad hoc queries and data analysis. You might not know the workload of your data warehouse in advance, so a data warehouse should be optimized to perform well for a wide variety of possible query and analytical operations.

  OLTP systems support only predefined operations. Your applications might be specifically tuned or designed to support only these operations.

- Data modifications

  A data warehouse is updated on a regular basis by the ETL process (run nightly or weekly) using bulk data modification techniques. The end users of a data warehouse do not directly update the data warehouse except when using analytical tools, such as data mining, to make predictions with associated probabilities, assign customers to market segments, and develop customer profiles [7].

  In OLTP systems, end users routinely issue individual data modification statements to the database. The OLTP database is always up to date, and reflects the current state of each business transaction.

- Schema design

  Data warehouses often use denormalized or partially denormalized schemas (such as a star schema) to optimize query and analytical performance..

- Typical operations [5]

  A typical data warehouse query scans thousands or millions of rows. For example, "Find the total sales for all customers last month."

  A typical OLTP operation accesses only a handful of records. For example, "Retrieve the current order for this customer."

- Historical data

  Data warehouses usually store many months or years of data. This is to support historical analysis and reporting.

  OLTP systems usually store data from only a few weeks or months. The OLTP system stores only historical data as needed to successfully meet the requirements of the current transaction.

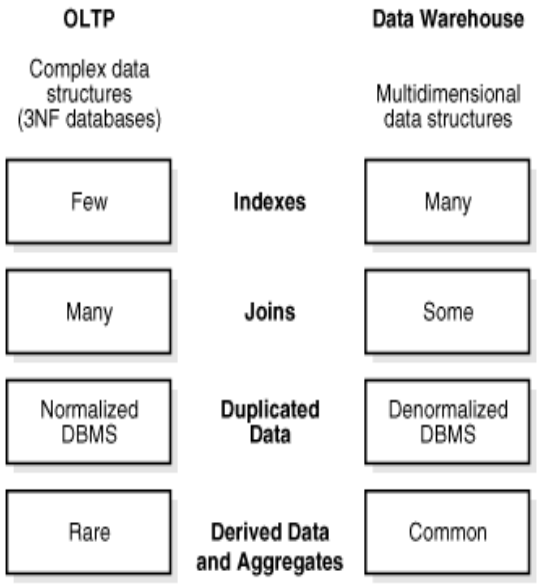

Figure 1-1 Contrasting OLTP and Data Warehousing Environments

VI. DATA WAREHOUSE ARCHITECTURES

Data warehouses and their architectures vary depending upon the specifics of an organization's situation. Three common architectures are [3]:

- Data Warehouse Architecture: Basic
- Data Warehouse Architecture: with a Staging Area
- Data Warehouse Architecture: with a Staging Area and Data Marts

Data Warehouse Architecture: Basic

Figure 1-2 shows a simple architecture for a data warehouse. End users directly access data derived from several source systems through the data warehouse.

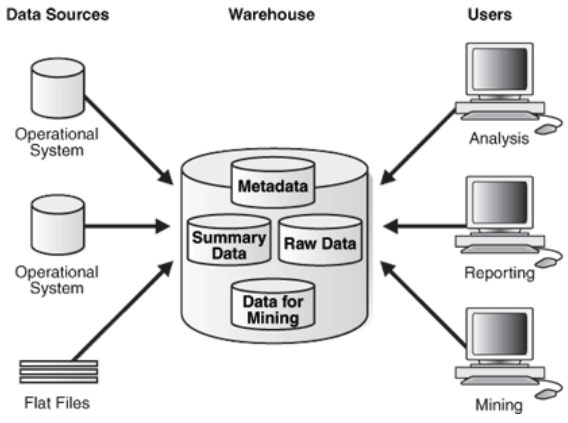

Figure 1-2 Architecture of a Data Warehouse

In Figure 1-2, the metadata and raw data of a traditional OLTP system is present, as is an additional type of data, summary data. Summaries are very valuable in data warehouses because they pre-compute long operations in advance. For example, a typical data warehouse query is to retrieve something such as August sales. A summary in an Oracle database is called a materialized view [8].

Data Warehouse Architecture: with a Staging Area

You need to clean and process your operational data before putting it into the warehouse, as shown in Figure 1-2. You can do this programmatically, although most data warehouses use a staging area instead. A staging area simplifies building summaries and general warehouse management. Figure 1-3 illustrates this typical architecture.

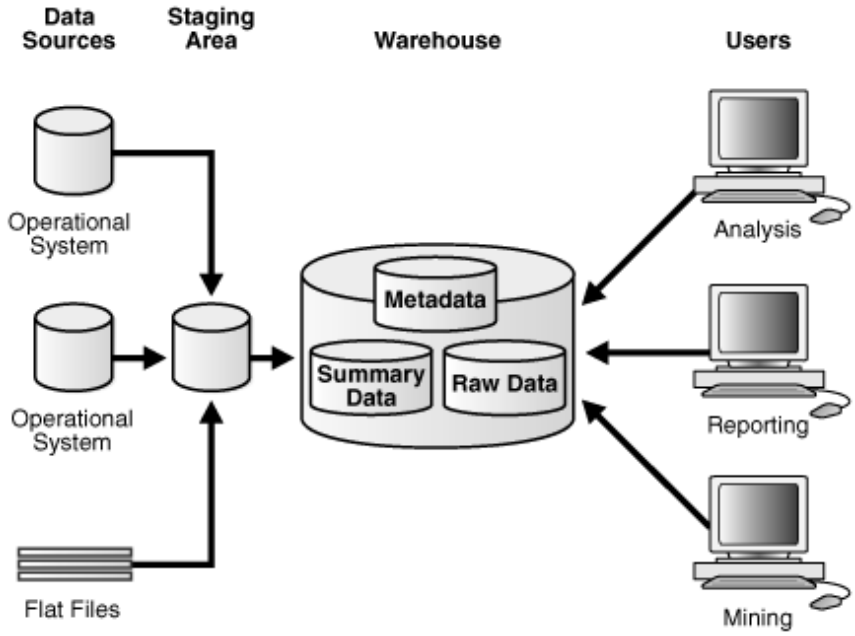

Figure 1-3 Architecture of a Data Warehouse with a Staging Area

Data Warehouse Architecture: with a Staging Area and Data Marts

Although the architecture in Figure 1-3 is quite common, you may want to customize your warehouse's architecture for different groups within your organization. You can do this by adding data marts, which are systems designed for a particular line of business. Figure 1-4 illustrates an example where purchasing, sales, and inventories are separated [7]. In this example, a financial analyst might want to analyze historical data for purchases and sales or mine historical data to make predictions about customer behavior.

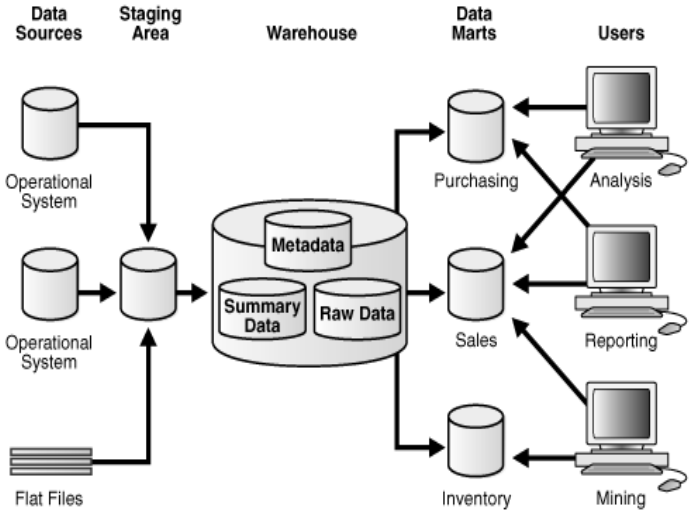

Figure 1-4 Architecture of a Data Warehouse with a Staging Area and Data Marts

VII. CONCLUSION

Data mart and data warehousing are tools to assist management to come up with relevant information about the organization at any point of time. While data marts are limited for use of a department only, data warehousing applies to an entire organization. Data marts are easy to design and use while data warehousing is complex and difficult to manage. Data warehousing is more useful as it can come up with information from any department

VIII. REFRENCES

1. Berry, M., J., A., & Linoff, G., S., (2000). Mastering data mining. New York: Wiley.
2. Edelstein, H., A. (1999). Introduction to data mining and knowledge discovery (3rd ed). Potomac, MD: Two Crows Corp.
3. Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1996). Advances in knowledge discovery & data mining. Cambridge, MA: MIT Press.
4. Han, J., Kamber, M. (2000). Data mining: Concepts and Techniques. New York: Morgan-Kaufman.
5. Hastie, T., Tibshirani, R., & Friedman, J. H. (2001). The elements of statistical learning: Data mining, inference, and prediction. New York: Springer.
6. Pregibon, D. (1997). Data Mining. Statistical Computing and Graphics, 7, 8.
7. Weiss, S. M., & Indurkhya, N. (1997). Predictive data mining: A practical guide. New York: Morgan-Kaufman.
8. Westphal, C., Blaxton, T. (1998). Data mining solutions. New York: Wiley.
9. Witten, I. H., & Frank, E. (2000). Data mining. New York: Morgan-Kaufmann.
10. http://www.kron.com/nc4/contact4/stories/computer_privacy.html
11. http://www.privacyrights.org
12. http://www.cfp.org