
Data Warehousing: An Overview

By

Katherine Watier

December 3, 2001

MGMT 257-01

Marco Gonzalma

Data Warehousing: An Overview

Data warehouses were developed in the mid 1980s as a response to users' frustration with current IT applications and their inability to provide complete answers to business questions that involved separated business data (sales data and employee time data for instance). Questions like "Why were sales up in the Pacific Region but not in the Mountain Region?" or "Why are we over budget" were not easily answered, but with the added advantage of multi-dimensional querying provided by the data warehouse, managers can now arrive at intelligent answers to such questions. (Barquin, 1997) Additionally, managers can (by using data mining techniques) recognize patterns within data and predict future behaviors based on current characteristics, and the system can be developed to prompt the managers when trends occur that could impact business revenues.

In addition to frustrations over the limits of the technology, expanding markets also spurred the development of data warehousing. As businesses fragmented those audiences into micro-segments with special business needs and marketing appeals, the data flow became overwhelming. "The next state, and the next area for competitive differentiation, revolves around the intensification of analysis. Astute managers will shift their attention from *systems* to *information*." (Kelly, 1994:13) Competition was often cited as the main reason behind developing a data warehouse. "The common factor which triggers investment in data warehousing appears to be competitive intensity." (Kelly, 1994: 14) The data warehouse became the perfect solution to not only managing large amounts of data but also being able to access and use that historical data to solve current business problems and to maintain a competitive advantage.

Large corporations in retail, banking and telecommunications were among the first to build the first data warehouse (DW) in the US in the 1980s. These DW were created to

integrate data that had been fragmented across large complex organizations and the most common applications were used for marketing and sales analysis. The term Data Warehouses originated by W. H. Inmon who published the first papers on the subject and concurrently, IBM published the first vendor commentary about data warehouses, who was one of the first two vendors that entered the field were IBM (along with Teradata Corporation) (Moeller, 2001: 30). Today, data warehouses are becoming an essential part of running a successful business and many are following in the footsteps of Walmart who as the first retail corporation in 1998 (Westerman, 2001: 7) to establish a data warehouse has been able to use it to efficiently track and market to its core audience.

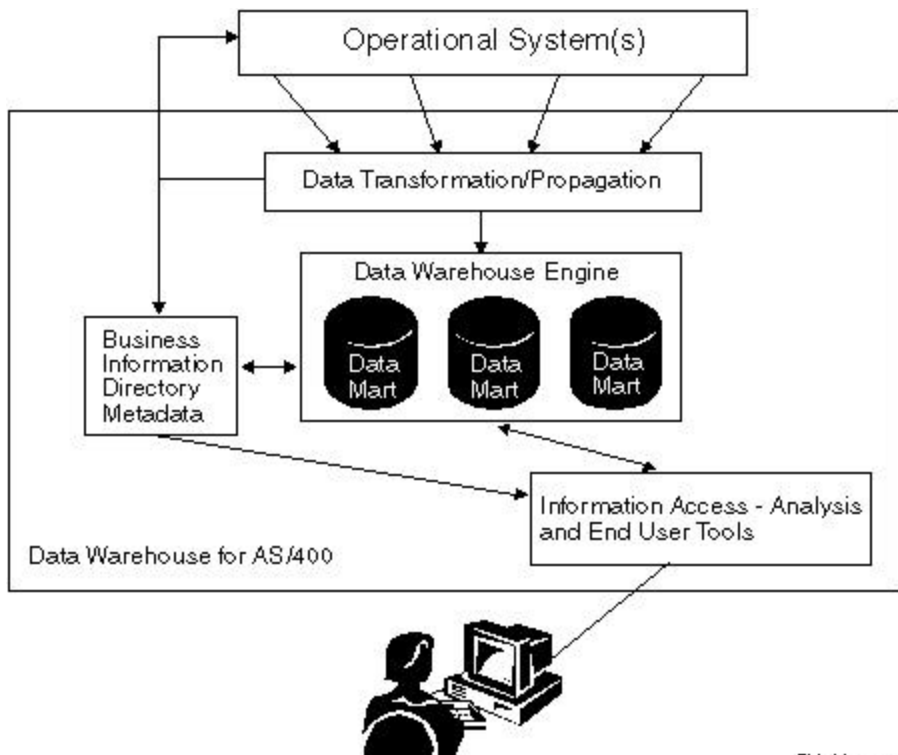
Data warehouses have three primary uses. They aid in the presentation of standard reports and graphs from multiple transaction systems. Secondly, they support dimensional analysis (a form of querying and reporting), which allows the user to look at data summaries from across a number of important data attributes (especially time periods). Thirdly, data warehousing supports data mining, which automatically recognizes patterns in the data to help the user describe existing data and predict the future behaviors of your customers and/or business trends (Barquin, 1997). Beyond simply storing historical data, data warehouses are made up of a variety of other tools that assist with the maintenance of the data and the querying process. Most data warehouses have:

- Online Transactional Processing OLTP/legacy systems which provide the data
- Transformation/propagation tools to populate the data warehouse
- Data warehouse database server
- Analysis/end-user tools (OLAP)
- Tools to manage information about the warehouse (metadata)

- Data marts for highly specified querying (usually for a department)

The following diagram illustrates the technical architecture that makes up a data warehouse. The Operational Systems depicted can contain legacy systems, OLTP systems, or a variety of external data sources.

Data warehousing Diagram



RV3M1086-1 (IBM online)

There are a variety of reasons for implementing a data warehouse: The corporation might want to conduct efficient analysis based on business processes and decision-making rather than transactions or events without impairing the performance of transactions. Often the useful data and metadata for such queries is fragmented across multiple transaction systems, on PCs and available from external data providers and the data warehouse creates an easy and efficient way to access that data. Executives within the company might want support for their strategic decision making by providing detail

and summary data that can be used in trend analysis, performance measurement comparison, statistical analysis, correlation among disparate facts and other similar requirements (Hammergren, 1998:85). Similarly, the corporation might want to promote a single source of authoritative, consistent, accurate and timely data that cuts across departmental applications. Many corporations want to empower the workforce by providing access to data for business user improving analysis capabilities and reducing the dependence of time-consuming specialized report development. Or perhaps the corporation has discovered problems with its data quality between disparate departmental data storage systems, and a data warehouse would allow for the elimination and reconciliation of that inconsistent data. Often, especially in the current business environment's focus on implementing knowledge management strategies, corporations might want to implement a data warehouse as a way to document organizational knowledge.

Data warehouse Vs. OLTP (Transactional) System

In order to understand the history and purpose for data warehouses, one must realize the limitations of traditional operational databases in assisting with business decisions. Operational processes and systems capture the day-to-day changes in data transactions for the company (like sales data), whereas analytical systems are systems that provide information for the analysis of business problems, situations and future projections. Analytical processes (those carried about by a data warehouse) are often done through comparing data (often from various legacy systems and previous disconnected databases) and looking for patterns and trends within that data.

There are a wide range of differences between transactional (OLTP) applications and data warehousing applications. Transactional Systems are organized around the transactions that they perform, such as entering orders, updating inventory, etc and data

warehouses are organized around a particular subject, such as the customer or product (Barquin, 1997). In order to cull and track all the information about a subject (for instance a customer) the data warehouse received data from applications such as: accounts receivable, customer service, credit authorization, sales management, archived files etc. The following table outlines in detail the differences between the two structures:

| Online Transactional Processing System | Data warehouse |
|---|--|
| System that tracks real-time data | System that monitors trends and other patterns |
| Large queries bog down performance of system | Available for large queries without impairing day to day functionality of transactional systems |
| Separate from other data systems | Integrated |
| Fast (often near instant) queries | Queries can take hours to days to complete |
| Hundreds of users access the system at one time | Limited number of users |
| Time stamped with time of event | Time-stamped with period of time |
| Event oriented | Subject oriented |
| Volatile | Non-volatile |
| Limited size usually within one hardware system | Large Systems that often require the establishment of parallel hardware structures including historical data stores and data marts |
| Usually requires custom training to use | Easily accessible to users with limited knowledge of data structures or computer systems |
| Difficult to scale & add features | Easy to scale & add features |
| Current (Real-Time) Data | Data is usually 5-10 years old |
| Data often created from one source | Data input from multiple sources in a variety of formats. |
| Highly normalized Data Structure | Often de-normalized |
| Due to stable transaction definitions, database design is relatively stable | Constantly changing business requirements create a design that is under pressure to be adaptive |

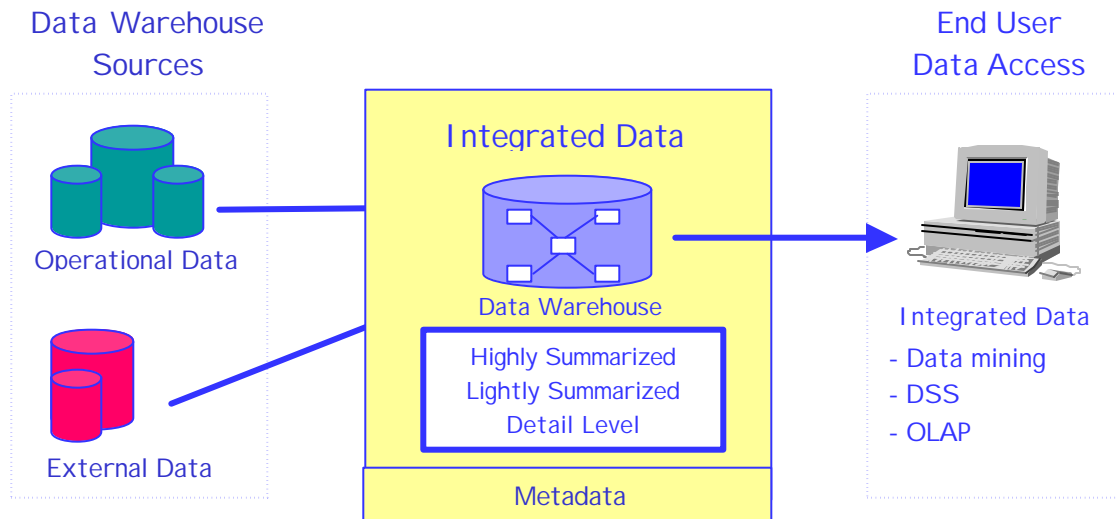
Due to the fewer number of users accessing the DW, the queries can be more resource intensive and do not need to be returned in the same type of turn around as the transactional system. Data warehouse queries are more resource intensive due to the need to look at potentially billions of rows often by joining five or more tables to get an answer. OLTP are designed differently. OLTP structures are highly normalized with only a single reference to an attribute in a single table. For example, a table that had sister1 and sister2 would require the creation of a separate sibling table to store the sister's

names with the foreign key for that table stored within the original table. Due to the amount of data that is stored within the data warehouse and in combined with the need to increase performance, data warehouses do not follow the same normalization standards. The historical nature of the data fed into the data warehouse implies that time plays an important role in the aggregation and comparison of values and the prediction of trends (Barquin, 1997). Data warehouses also receive new data less frequently than OLTP systems.

Unlike a relational database, often the database structure within the database is denormalized to increase performance and allow for multi-dimensional queries.

Many of the unique modeling transformations within a data warehouse include the denormalization of entities such as dimensions. My experience has been that de-normalization of the operational table provide useful data warehouse objects without requiring a lot of manipulation from the client application. While de-normalization can be at times controversial with database purists, the price of disk is quickly becoming an insignificant cost, and the results of de-normalization include a more useful product for the user. Not to mention the performance savings you will experience by avoiding dynamic or user of defined joins!
(Hammergren, 1998: 68)

Data analyzed within the data warehouse are "snapshots" of static data, versus the constantly updated data that is created through operational systems. In order to compare data, the data must be static, and intensive data analysis is often resource and time intensive and that sort of analysis would slow down operational systems if the data were not separated for analysis (Kelly, 1997). The purpose of a data warehouse is to combine the data, transform it to insure accuracy and consistency, and organize it for easy access and efficient querying (IBM online).



A Diagram of the Types of Data and Data Flow within the Data Warehouse

Types of Data within the Warehouse

The bulk of the data within the data warehouse often originates in various transactional systems (examples of OLTP systems include a reservation system, a online registration system, an accounting application or a order entry application), but other data can come from market research companies, departmental PCs, and a variety of other sources. Data warehouse data is subject oriented and organized in relation to the major entities of the corporation such as: customers, products, vendors, transaction, orders, policies, accounts, and shipments. The subject orientation allows for the data use to change over time without fundamentally affecting its organization or structure (Barquin, 1997). It is easiest to think of the data in the data warehouse as nothing but snapshot historical records. There are 2 popular models for recording history, State and Event. A record using a State Model has a "from" and a "to" date that denotes the time that the record was accurate. (i.e. Customer ID, From Data, To Data). The event model reflects the

moment in time the record was accurate and often has an Order ID, Part Number and Order Data. (Inmon, 2001:95) The data warehouse is able to store summary as well as detailed data to meet the needs of the various users. There are various examples of summary data - for example, a profile record (customer record and record for each use of the telephone during the statement cycle.) or a transaction summary record. (Inmon, 2001:97)

Changing the Data for the Data Warehouse Environment

Data that is found in OLTP, legacy or other systems must be transformed and cleaned before being available for analysis within the data warehouse. Getting clean data into the DW presents a variety of challenges. The naming structures of independent (often departmentally oriented databases) are different, and often the names of customers (for instance) occur in multiple places and accuracy problems can create significant problems (Barquin, 1997). It is therefore imperative that data acquisition, cleaning and transformation tools are in place to receive the data from the transactional systems and prepare it for the data warehouse. There are three principle types of these tools:

1. Data copying/extraction/ replication tools that move data from relation databases to relation databases and allow for the efficient transfer of data from non-relational sources into relational databases. (Barquin, 1997)
2. Data transformation tools extract data from relational and non-relational sources and they convert codes and calculate derived values. They often create new programs that regularly extract and change the data.
3. Data cleaning tools allow for the compilation of similar data from multiple sources about customers and suppliers when the data entry clerks used different spellings for the same subject.

These tools are found in the Information and Transformation (I&T) Layer where transaction data is gathered directly, either from the end user or directly from the consumer. As raw data is passed through the I&T layer, a fundamental alteration is done to the data to achieve an integrated foundation for the DW. As the data is passed, it is integrated through a complex process which include:

- Key resolution
- Resequencing of data
- Merging of data
- Restructuring of data layers
- Aggregation of data
- Summarization of data
- Outputs logic of transformation that is placed in the metadata repository.

The I & T layer converts the data into a universal format with universal naming conventions, creates new fields that are derived from existing operational data, summarizes data to the most appropriate level needed for analysis, and denormalizes the data for performance purposes. (Inmon, 2001: 94)

Metadata

Just as important as cleaning and populating the data warehouse with data, it is imperative to develop and maintain (and store) information about the data to aide business users in understanding their analysis and the assist the administrator in maintaining the data warehouse structure and integrity. The **technical metadata** (or administrative data) contains a description of the operational database and a description of the data warehouse. This data contains descriptions of the source databases and their contents, the objects within the data warehouse and the transformations conducted when the data is moved into the data warehouse (Barquin, 1997). Metadata can also contain information about the origins of the source data, what data is scheduled to be purged, who is using the data warehouse and how they are using the data, etc. This data helps the data warehouse administrators maintain the data warehouse, know where all of the data is coming from.

The **business metadata** (or end-user data) helps users find information in the data warehouse without knowing the underlying implementation of the database. This information is presented in business terms, instead of the terms used by the

programmers when the database was built. The business data also gives the user information about:

- When the data was moved into the warehouse (how current it is)
- Where the data came from (which operational database)
- Other information that lets the user know how reliable the data is

Metadata assists the business users create and interpret their queries. Metadata is usually stored in a metadata repository and is managed by metadata repository management software that typically runs on a workstation and enables user to specify how the data should be transformed (Berson, 1997:119).

Querying the Data

One of the most unique features of the data warehouse is the ability to conduct queries that are multi-dimensional. Multidimensional Analysis is necessary function for a data warehouse. It allows the business user to view data entries such as products, geographies, time periods, etc. that may represent different dimensions. Relational databases only store data in a two dimensional format: tables of data represented by rows and columns. With a multi-dimensional solution it is possible to quickly analyze potentially large amounts of data, drill down or roll up through various dimensions as defined by the data structure, and quickly identify trends or problem areas that would otherwise be missed (IBM, online source). In order to take advantage of the features offered by both database designs, many companies create both and only store specialized data within the MDD.

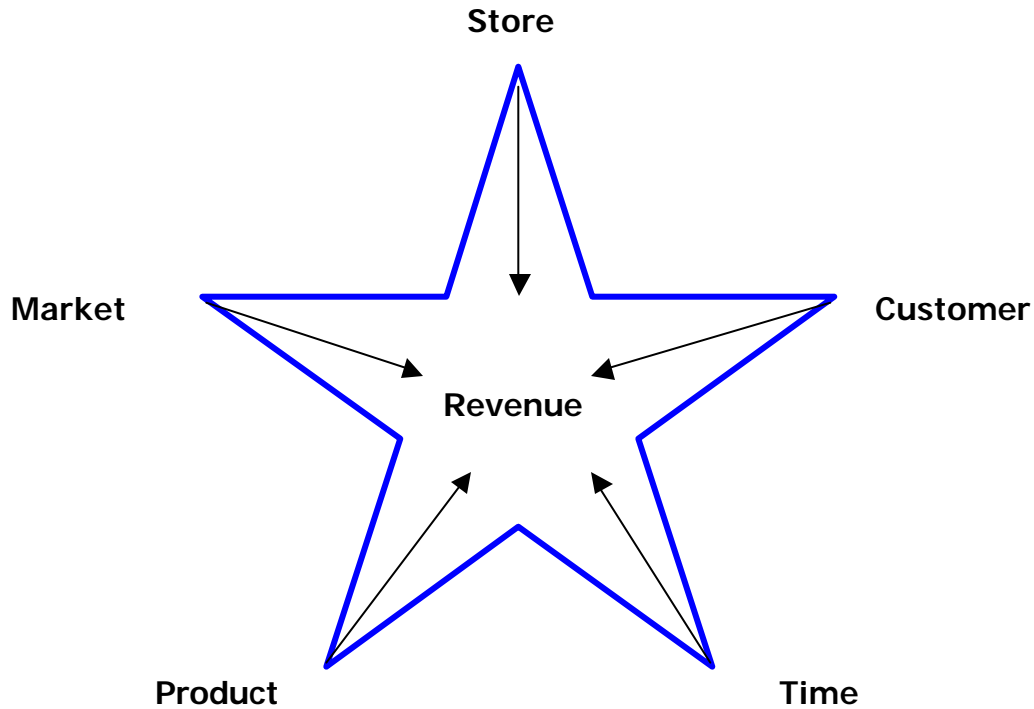
Database structures for Multi-Dimensional Queries - Star Schema

Frequently, in order to support multi-dimensional analysis, software on the client's workstation may require the data to be structured as a "star schema" to accommodate the software operation and simplify the user's view of the data. (Barquin, 1997) A "star schema" is a specific type of database design used to support analytical processing. Pioneered by Ralph Kimball, it is a mechanism by which the relational database can be configured to optimize queries that are naturally dimensional. (Kelly, 1994:121) It has a specific set of normalized tables: fact tables and dimension tables. Fact tables (also called major tables) contain the quantitative or factual data about a business. Often they have numerical measurements and can consist of many columns and millions of rows. These tables are used for types of data that occur in great volume (Berson, 1997:115) Dimensional tables, sometimes called minor tables, are smaller and hold descriptive data that reflects the dimension of a business. Dimension tables are smaller tables are joined to the data in the fact table by means of a foreign key relationship. Selected information is returned by using SQL queries that are made up of predefined and user-defined join paths between fact and dimension tables that have data constraints. (Berson, 1997:33) The easiest way to understand the type of data that is stored within each table is to think of how a sales person analyzes revenue. They look at the revenue *by* customer, product, market and time period. The revenue belongs in the fact table, and all of the "by" comparisons belong in the fact tables. (Poe, 1996: 123) "Whenever one hears the 'by' word, one knows that the query has many dimensions. For example, the query 'I want to know the breakdown of customers by region, by demographics, by value and by segment' is a classic multi-dimensional query. The critical issue for the database designer relations to the data partitioning strategy that is pursued." (Berson, 1997:125)

The star schema gets its name from the pattern the tables make when joined together.

A very large fact table containing the detailed transaction data is the center of the star with much smaller tables containing dimensional data forming the points of the star.

The following diagram illustrates the design of the star schema:



“Each fact table also includes a multipart index that contains as foreign keys the primary keys of related dimension tables, which contain the attributes of the fact records. Fact tables should not contain descriptive information or any data other than the numerical measurement fields and the index fields that relate the facts to corresponding entries in the dimension tables. ” (IBM online) The central table normally contains business data that can be expressed in units, and the tables that create the points of the star normally contain textual values that equate to constraints. For example, a query that “selects all women under age 25” the “under age 25” is a constraint. Fact tables are fully normalized while dimensional tables tend to be denormalized into a more flat structure (Kelly,

1994:122). This type of structure limits the number of table joins that need to occur in order to answer complex query.

Tools to Help Users Analyze Data

Users access data in the data warehouse in a variety of ways, though the two most prevalent are ad hoc queries and packaged queries and reports (Barquin, 1997). The latter is more frequently used and the systems that customize these reports are often called executive support systems or decision support systems. **Business intelligence software** is a fairly new term referring to the tools that are used to analyze the data.

This software can consist of:

- **Decision support systems** (DSS) tools, which allow you to build ad hoc queries and generate reports
- **Executive information systems** (EIS), which combine decision support with extended analysis capabilities and access to outside resources (such as Dow Jones News services)
- **Data mining** tools, which allow automation of the analysis of your data to find patterns or rules that you can use to tailor business operations (IBM online)

Another feature often used by business users is exception reporting or alerts. Alert systems are developed to monitor specific indicator or combinations of indicators and send out emails or other notices whenever the values exceed the acceptable ranges (Barquin, 1997).

There are a variety of tools that are used to provide easy user access to the Data warehouse. Online Analytical Processing (OLAP) allows the users to look at a summary of data and then “drill down” through the data to see yet more details. For example, a

“manager may first ask for the sales data summarized by week for the quarter in question, then drill down to the days of the week to find that the unusual sales amount is a one-time event. Having discovered the time of the event, the manager then explores the sales by product and finds that a large sale of a specific product was made on that day. Based on these answers, the manager then asks for the sales by region and drills down to find that a particular store filled a large order of a specific product on a specific day.” (IBM online) This type of querying is supported by multidimensional data structures (“cubes”) that are predefined and created to organize and summarize data warehouse data in such a way that often asked explorative analysis questions can be answered with little or no querying of the relational database.

In contrast to OLAP queries, data mining automatically analyzes the data and identifies interesting nuggets of information such as groupings of data for the analyst or manager to examine. “Data mining can also create decision trees that can be used to predict future data based on attributes of existing data elements.” (IBM online) Corporations use data mining to discover the hidden, previously undetected facts about customers, retailers, suppliers, business trends and other significant factors. With data mining, the system researches the data and determines patterns, classifications, and associations, while the analyst determines what to do with the results.

Data mining has been quite useful in the retail industry to analyze consumer buying patterns and form marketing programs to take advantage of the analysis results. For instance, data mining can find patterns in your data to answer questions like:

- What item purchased in a given transaction triggers the purchase of additional related items?
- How do purchasing patterns change with store location?
- What items tend to be purchased using credit cards, cash, or check?

- How would the typical customer likely to purchase these items be described?
- Did the same customer purchase related items at another time?

Often querying the entire data warehouse is not efficient and a user is only interested in a specialized section of the data. In these instances, storing and analyzing data within a smaller (or more focus) data warehouse (called data marts) is more effective. Datamarts are specialized sections that address specific areas of the organization. They could be considered separate data warehouses due to their size –often growing larger than the data warehouse that spawned them- and querying ability. They are designed to meet the special focused needs of a particular aspect of a business and often the data structures and summaries are even designed for that department in mind. Due to the time it takes to build a data warehouse, companies often build data marts first so that they can benefit from the querying functionality (Barquin, 1997). Data marts can be created before the data warehouse is constructed or afterwards, but they must adhere to the central design specifications in order to produce reports that are consistent even though the data resides in different places.

Maintaining the Data Warehouse

Once a data warehouse is built, there is often additional work to be done to enhance and expand its functionality. Additional data from other data sources need to be added, often front-end access applications need to be enhanced, and there is often ongoing training. Help desks for the data warehouse are established and on-going internal marketing plans are put into place. Often data warehousing key groups are established to identify strategic issues and what the future data warehousing issues may be.

As a rule, the data warehouse administrator manages the data warehouse. The Data Warehouse Administrator organization is responsible for: building the data warehouse, ongoing monitoring and maintenance of the data warehouse, coordinating usage,

recording management feedback –success and failures-, securing the resources for the data warehouse, selection of hardware and software components. (Imnon, 2001:102)

The data warehouse administrator monitors the amount of data being added to the data warehouse, the quality of data, and creates a data content card catalog that else what the content of the data warehouse is and projects future growth, prepares for data recover, removes data from the data warehouse, and creates indexes. (Inmon, 2001:103-104) There are many automated systems that can assist the data warehouse Administrator in managing the data warehouse. These include tools that provide security, network monitoring, backup and more.

Developing the Data warehouse

Data warehouse development should be evolutionary and how the data is culled and presented and how queries are run is directly linked to current and future business processes. The data warehouse should therefore be flexible enough to change with changing business priorities. When developing a data warehousing project, the developers need to have an understanding of the various positions within a firm that will be using the new system and need to take into account their distinct needs and interests in accessing business data. These positions most often accessing the data warehouse include:

Casual user: someone who needs access to the information on occasional basis. They like big button navigation and prompting of choices for predefined analysis.

Business Analysts- This is the largest group of user. They use the information daily but don't have the knowledge to build reports completely from scratch. This type of user will want predefined navigation, ability to customize, and looks at reports n a variety of different ways.

Power users: This user will want to write their won macros, change parameters, and manipulate result sets. They are comfortable starting with a clean state and creating

their own reports/analyses. Power Users use the data from the DATA WAREHOUSE to segment customers, analysis product performance and sales, and project customer lifetime value.

Application developer: His/her primary responsibilities are to support the business rather and having other actual business responsibilities. They are trained to create reports/analyses for use by others and are the driving force in setting standards and identifying where and how reports will be named and located. (Berson, 1997)

Involving these users early on in the establishment of the business requirements will ensure a more successful data warehouse venture and a greater change for universal adoption within the business, and considering that Failures for DQ projects are in the 30-50% range, it is imperative to develop a strong strategy for the development, implementation, use, maintenance, support and marketing of the data warehouse. Once a data warehouse is used within the company, however, the benefits to profitability are almost immeasurable. It is clear from those organizations that have implemented and now use data warehouses (Walmart, Citibank. etc.) that data warehouses are becoming an essential part of a successful business plan.

References

- IBM online. Retrieved from the World Wide Web October 23, 2001. http://www-1.ibm.com/servers/eserver/iseriess/db2/dataware.htm#header_4
- Barquin, R. & Edelstein, H. 1997. Planning and Designing the Data Warehouse. Prentice Hall PTR, Upper Saddle River, NJ.
- Berson, A. & Smith, S. 1997. Data Warehousing, Data Mining, & OLAP. McGraw-Hill, New York, NY.
- Hammergren, T. 1998. Data Warehousing on the Internet: Accessing the Corporate Knowledge Base. International Thompson Computer Press: Boston, MA.
- Inmon, W.H., Imhoff, C., & Sousa, R. 2001. Corporate Information Factory. Wiley Computer Publishing: New York, NY.
- Kelly, Sean. 1994. Data Warehousing: The Route to Mass Customization. John Wiley & Sons: New York, NY.
- Kelly, S. 1997. Data Warehousing in Action. John Wiley & Sons, Ltd. New York: NY.
- Moeller, R.A. 2001. Distributed Data Warehousing Using Web Technology: How to Build a More Cost-Effective and Flexible Warehouse. AMACOM: New York, NY.
- Poe, Vidette. 1996. Building a Data Warehouse for Decision Support. Prentice Hall PTR: Upper Saddle River, NJ.

Westerman, Paul. 2001. Data Warehousing: Using the Walmart Model. Morgan
Kaufmann Publishers: San Francisco, CA.