BCA IV Semester

Paper Code CSA-EC-412

Paper Name- Datawarehousing and mining

Teacher's Name - Richa pathak

Department- computer science and application

Unit 2

The Building Blocks

DEFINING FEATURES

Let us examine some of the key defining features of the data warehouse based on these definitions.

Subject-Oriented Data

In operational systems, we store data by individual applications. In the data sets for an order processing application, we keep the data for that particular application. These data sets provide the data for all the functions for entering orders, checking stock, verifying customer's credit, and assigning the order for shipment. But these data sets contain only the data that is needed for those functions relating to this particular application. We will have some data sets containing data about individual orders, customers, stock status, and detailed transactions, but all of these are structured around the processing of orders.

Similarly, for a banking institution, data sets for a consumer loans application contain data for that particular application. Data sets for other distinct applications of checking accounts and savings accounts relatefto those specific applications.

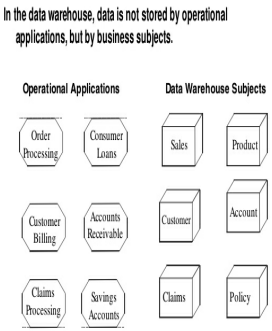

Figure 2-1 The data warehouse is subject oriented.

Figure 2-1 distinguishes between how data is stored in operational systems and in the data warehouse. In the operational systems shown, data for each application is organized separately by application: order processing, consumer loans, customer billing, accounts receivable, claims processing, and savings accounts. For example, Claims is a critical business subject for an insurance company. Claims under automobile insurance policies are processed in the Auto Insurance application. Claims data for automobile insurance is organized in that application.

Integrated Data

For proper decision making, you need to pull together all the relevant data from the various applications. The data in the data warehouse comes from several operational systems. Source data are in different databases, files, and data segments. These are disparate applications, so the operational platforms and operating systems could be different. The file layouts, character code representations, and field naming conventions all could be different.

In addition to data from internal operational systems, for many enterprises, data from outside sources is likely to be very important. Your data warehouse may need data from such sources. This is one more variation in the mix of source data for a data warehouse.

Figure 2-2 illustrates a simple process of data integration for a banking institution. Here the data fed into the subject area of account in the data warehouse comes from three different operational applications. Even within just three applications, there could be several variations. Naming conventions could be different; attributes for data items could be different. The account number in the Savings Account application could be eight byte long, but only six bytes in the Checking Account application.

Before the data from various disparate sources can be usefully stored in a data warehouse, you have to remove the inconsistencies. You have to standardize the various data elements and make sure of the meanings of data names in each source application. Before moving the data into the data warehouse, you have to go through a process of transformation, consolidation, and integration of the source data.

Here are some of the items that would need standardization:

1. Naming conventions

2.Codes

3. Data attributes

4. Measurements

Time-Variant Data

For an operational system, the stored data contains the current values. In an accounts receivable system, the balance is the current outstanding balance in the customer's account. In an order entry system, the status of an order is the current status of the order. In a consumer loans application, the balance amount owed by the customer is the current amount.

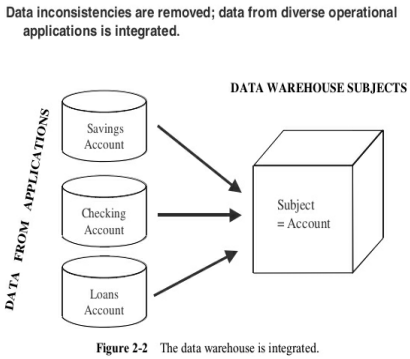

Figure 2-2 The data warehouse is integrated.

A data warehouse, because of the very nature of its purpose, has to contain historical data, not just current values. Data is stored as snapshots over past and current periods. Every data structure in the data warehouse contains the time element. You will find histor-ical snapshots of the operational data in the data warehouse. This aspect of the data warehouse is quite significant for both the design and the implementation phases.

Data from the operational systems are moved into the data warehouse at specific intervals. Depending on the requirements of the business, these data movements take place twice a day, once a day, once a

week, or once in two weeks. The units of sales may be moved once a day. As illustrated in Figure 2-3, every business transaction does not update the data in the data warehouse. The business transactions update the operational system databases in real time. We add, change, or delete data from an operational system as each transaction happens but do not usually update the data in the data warehouse. You do not delete the data in the data warehouse in real time. Once the data is captured in the data warehouse, you do not run individual transactions to change the data there. Data updates are commonplace in an operational database; not so in a data warehouse. The data in a data warehouse is not as volatile as the data in an operational database is. The data in a data warehouse is primarily for query and analysis.

Data Granularity

In an operational system, data is usually kept at the lowest level of detail. In a point-ofsale system for a grocery store, the units of sale are captured and stored at the level of units of a product per transaction at the check-out counter. In an order entry system, the quantity ordered is captured and stored at the level of units of a product per order received from the customer. Whenever you need summary data, you add up the individual transactions. If you are looking for units of a product ordered this month, you read all the orders entered for the entire month for that product and add up. You do not usually keep summary data in an operational system. When a user queries the data warehouse for analysis, he or she usually starts by looking at summary data. The user may start with total sale units of a product in an entire region. Then the user may want to look at the breakdown by states in the region. The next step may be the examination of sale units by the next level of individual stores. Frequently, the analysis begins at a high level and moves down to lower levels of detail.

In a data warehouse, therefore, you find it efficient to keep data summarized at different levels. Depending on the query, you can then go to the particular level of detail and satisfy the query. Data granularity in a data warehouse refers to the level of detail. Figure 2-4 shows examples of data granularity in a typical data warehouse.

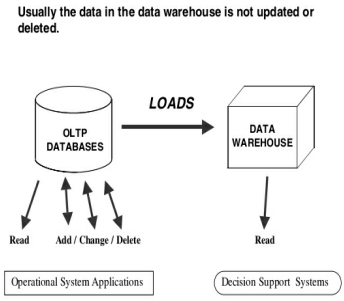

Figure 2-3 The data warehouse is nonvolatile.

DATA WAREHOUSES AND DATA MARTS

If you have been following the literature on data warehouses for the past few years, you would, no doubt, have come across the terms “data warehouse” and “data mart.” Many who are new to this paradigm are confused about these terms. At this point, it would be worthwhile for us to examine these two terms and take our position.

Before deciding to build a data warehouse for your organization, you need to ask the following basic and fundamental questions and address the relevant issues:

Top-down or bottom-up approach

Enterprise-wide or departmental?

Which first—data warehouse or data mart?

Build pilot or go with a full-fledged implementation?

Dependent or independent data marts?

These are critical issues requiring careful examination and planning. Should you look at the big picture of your organization, take a top-down approach, and build a mammoth data warehouse? Or, should you adopt a bottom-up approach, look at the individual local and departmental requirements, and build bite -size departmental data marts? Should you build a large data warehouse and then let that repository feed data into local, departmental data marts? Should these local data marts be independent of one another? Or, should they be dependent on the overall dat warehouse for data feed? Should you build a pilot data mart? These are crucial questions.

THREE DATA LEVELS IN A BANKING DATA WAREHOUSE

Daily Detail	Monthly Summary	Quarterly Summary
Account	Account	Account
Activity Date	Month	Month
Amount	Number of transactions	Number of transactions
Deposit/Withdrawal	Withdrawals	Withdrawals
	Deposits	Deposits
	Beginning Balance	Beginning Balance
	Ending Balance	Ending Balance

Data granularity refers to the level of detail. Depending on the requirements, multiple levels of detail may be present. Many data warehouses have at least dual levels of granularity.

Figure 2-4 Data granularity.

How are They Different?

Let us take a close look at Figure 2-5. Here are the two different basic approaches:

(1)overall data warehouse feeding dependent data marts, and

(2) several departmental or local data marts combining into a data warehouse. In the first approach, you extract data from the operational systems; you then transform, clean, integrate, and keep the data in the data warehouse. So, which approach is best in your case, the top-down or the bottom up approach? Let us examine these two approaches carefully.

Figure 2-5 Data warehouse versus data mart.

Top-Down Versus Bottom-Up Approach

Top-Down Approach

The advantages of this approach are:

A truly corporate effort, an enterprise view of data

Inherently architected—not a union of disparate data marts

Single, central storage of data about the content

Centralized rules and control

May see quick results if implemented with iterations

The disadvantages are:

Takes longer to build even with an iterative method

High exposure/risk to failure

Needs high level of cross-functional skills

High outlay without proof of concept

This is the big-picture approach in which you build the overall, big, enterprise-wide data warehouse. Here you do not have a collection of fragmented islands of information. The data warehouse is large and integrated. This approach, however, would take longer to build and has a high risk of failure. If you do not have experienced professionals on your team, this approach could be dangerous. Also, it will be difficult to sell this approach tosenior management and sponsors. They are not likely to see results soon enough.

Bottom-Up Approach

The advantages of this approach are:

Faster and easier implementation of manageable pieces

Favorable return on investment and proof of concept

Less risk of failure

Inherently incremental; can schedule important data marts first

Allows project team to learn and grow

The disadvantages are:

Each data mart has its own narrow view of data

Permeates redundant data in every data mart

Perpetuates inconsistent and irreconcilable data

Proliferates unmanageable interfaces

In this bottom-up approach, you build your departmental data marts one by one. You would set a priority scheme to determine which data marts you must build first. The most severe drawback of this approach is data fragmentation. Each independent data mart will be blind to the overall requirements of the entire organization.

A Practical Approach

In order to formulate an approach for your organization, you need to examine what exactly your

organization wants. Is your organization looking for long-term results or fast data marts for only a few subjects for now? Does your organization want quick, proof-of-concept, throw-away implementations? Or, do you want to look into some other practical approach? Although both the top-down and the bottom-up approaches each have their own advantages and drawbacks, a compromise approach accommodating both views appears to be practical. The steps in this practical approach are as follows:

1. Plan and define requirements at the overall corporate level

2. Create a surrounding architecture for a complete warehouse

3. Conform and standardize the data content

4. Implement the data warehouse as a series of supermarts, one at a time

In this practical approach, you go to the basics and determine what exactly your organization wants in the long term. The key to this approach is that you first plan at the enterprise level. You gather requirements at the overall level. You establish the architecture for the complete warehouse. Then you determine the data content for each supermart. Supermarts are carefully architected data marts. You implement these supermarts, one at a time. Before implementation, you make sure that the data content among the various supermarts are conformed in terms of data types, field lengths, precision, and semantics.

A data mart, in this practical approach, is a logical subset of the complete data warehouse, a sort of pie-wedge of the whole data warehouse. A data warehouse, therefore, is a conformed union of all data marts. Individual data marts are targeted to particular business groups in the enterprise, but the collection of all the data marts form an integrated whole, called the enterprise data warehouse.

OVERVIEW OF THE COMPONENTS

When we build an operational system such as order entry, claims processing, or savings account, we put together several components to make up the system. The front-end component consists of the GUI (graphical user interface) to interface with the users for data input. The data storage component includes the database management system, such as Oracle, Informix, or Microsoft SQL Server. The display component is the set of screens and reports for the users. The data interfaces and the network software form the connectivity component. Depending on the information requirements and the

Architecture is the proper arrangement of the components.

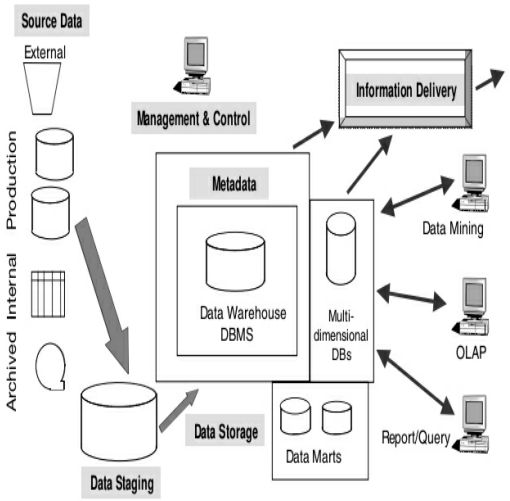

Figure 2-6 Data warehouse: building blocks or components.

framework of our organization, we arrange these components in the most optimum way. Architecture is the proper arrangement of the components. You build a data warehouse with software and hardware components. Figure 2-6 shows the basic components of a typical warehouse.

Data component shown on the left. The Data Staging component serves as the next building block. In the middle, you see the Data Storage component that manages the data warehouse data. This component not only stores and manages the data, it also keeps track of the data by means of the metadata repository. The Information Delivery component shown on the right consists of all the different ways of making the information from the data warehouse available to the users. Each data warehouse is put together with the same building blocks. The essential difference for each organization is in the way these building blocks are arranged. The variation is in the manner in which some of the blocks are made stronger than others in the architecture. We will now take a closer look at each of the components.

Source Data Component

Source data coming into the data warehouse may be grouped into four broad categories,as discussed here.

Production Data. This category of data comes from the various operational systems of the enterprise. Based on the information requirements in the data warehouse, you choose segments of data from the different operational systems. While dealing with this data, you come across many variations in the data formats. You also notice that the data resides on different hardware platforms. Further, the data is

supported by different database systems and operating systems. This is data from many vertical applications.

In operational systems, information queries are narrow. You query an operational system for information about specific instances of business objects. You may want just the name and address of a single customer.

Again, you do not expect a particular query to run across different operational systems. What does all of this mean? Simply this: there is no conformance of data among the various operational systems of an enterprise. A term like an account may have different meanings in different systems. The significant and disturbing characteristic of production data is disparity.

Internal Data. In every organization, users keep their "private" spreadsheets, documents, customerprofiles, and sometimes even departmental databases. This is the internal data, parts of which could be useful in a data warehouse. If your organization does business with the customers on a one-to-one basis and the contribution of each customer to the bottom line is significant, then detailed customer profiles with ample demographics are important in a data warehouse. Profiles of individual customers become very important for consideration. When your account representatives talk to their assigned customers or when your marketing department wants to make specific offerings to individual customers, you need the details.

. It is a collective judgment call on how much of the internal data should be included in the data warehouse. The IT department must work with the user departments to gather the internal data.

Internal data adds additional complexity to the process of transforming and integrating the data before it can be stored in the data warehouse. You have to determine strategies for collecting data from spreadsheets, find ways of taking data from textual documents, and tie into departmental databases to gather pertinent data from those sources.

Archived Data. Operational systems are primarily intended to run the current business. In every operational system, you periodically take the old data and store it in archived files. The circumstances in your organization dictate how often and which portions of the operational databases are archived for storage. Some data is archived after a year. Sometimes data is left in the operational system databases for as long as five years. Many different methods of archiving exist. There are staged archival methods. At the

first stage, recent data is archived to a separate archival database that may still be online.

At the second stage, the older data is archived to flat files on disk storage. At the next stage, the oldest data is archived to tape cartridges or microfilm and even kept off-site.

As mentioned earlier, a data warehouse keeps historical snapshots of data. You essentially need historical data for analysis over time.

External Data. Most executives depend on data from external sources for a high percentage of the information they use. They use statistics relating to their industry produced by external agencies. They use market share data of competitors. They use standard values of financial indicators for their business to check on their performance.

Usually, data from outside sources do not conform to your formats. You have to devise conversions of data into your internal formats and data types. You have to organize the data transmissions from the external sources. Some sources may provide information at regular, stipulated intervals. Others may give you the data on request. You need to accom modate the variations.

Data Staging Component

After you have extracted data from various operational systems and from external sources, you have to prepare the data for storing in the data warehouse. The extracted data coming from several disparate sources needs to be changed, converted, and made ready in a format that is suitable to be stored for querying and analysis.

Three major functions need to be performed for getting the data ready. You have to extract the data, transform the data, and then load the data into the data warehouse storage.

These three major functions of extraction, transformation, and preparation for loading take place in a staging area. The data staging component consists of a workbench for these functions. Data staging provides a place and an area with a set of functions to clean, change, combine, convert, deduplicate, and prepare source data for storage and use in the data warehouse.When we implement an operational system, we are likely to pick up data from different sources, move the data into the new operational system database, and run data conversions. A separate staging area, therefore, is a necessity for preparing data for the data warehouse.

Now that we have clarified the need for a separate data staging component, let us understand what happens in data staging. We will now briefly discuss the three major functions that take place in the staging area.

Data Extraction. This function has to deal with numerous data sources. You have to employ the appropriate technique for each data source. Source data may be from different source machines in diverse data formats. Part of the source data may be in relational database systems. Some data may be on other legacy network and hierarchical data models. Many data sources may still be in flat files. You

may want to include data from spreadsheets and local departmental data sets. Data extraction may become quite complex. Tools are available on the market for data extraction. You may want to consider using outside tools suitable for certain data sources.

After you extract the data, where do you keep the data for further preparation? You may perform the extraction function in the legacy platform itself if that approach suits your framework. More frequently, data warehouse implementation teams extract the source into a separate physical environment from which moving the data into the data warehousewould be easier.

Data Transformation. In every system implementation, data conversion is an important function. For example, when you implement an operational system such as a magazine subscription application, you have to initially populate your database with data from the prior system records. You may be converting over from a manual system. Or, you may be moving from a file-oriented system to a modern system supported with relational database tables. In either case, you will convert the data from the prior systems.

Again, as you know, data for a data warehouse comes from many disparate sources. If data extraction for a data warehouse poses great challenges, data transformation presents even greater challenges. Another factor in the data warehouse is that the data feed is not just an initial load. You will have to continue to pick up the ongoing changes from the source systems. First, you clean the data extracted from each source. Cleaning may just be correction of misspellings, or may include resolution of conflicts between state codes and zip codes in the source data, or may deal with providing default values for missing data elements, or elimination of duplicates when you bring in the same data from multiple source systems. Standardization of data elements forms a large part of data transformation.When two or more terms from different source systems mean the same thing, you resolve the synonyms. When a single term means many different things in different source systems, you resolve the homonym.

Data transformation involves many forms of combining pieces of data from the different sources. You combine data from a single source record or related data elements from many source records.

In many cases, the keys chosen for the operational systems are field values with builtin meanings. For example, the product key value may be a combination of characters indicating the product category, the code of the warehouse where the product is stored, and some code to show the production batch. Primary keys in the data warehouse cannot have built-in meanings.

A grocery chain point-of-sale operational system keeps the unit sales and revenue amounts by individual transactions at the check-out counter at each store. But in the dat warehouse, it may not be necessary to keep the data at this detailed level. You may want to summarize the totals by product at each store for a given day and keep the summary totals of the sale units and revenue in the data warehouse storage. In such cases, the data transformation function would include appropriate summarization.

Data Loading. Two distinct groups of tasks form the data loading function. When you complete the design and construction of the data warehouse and go live for the first time, you do the initial loading of the data into the data warehouse storage. The initial load moves large volumes of data using up

- This function is time-consuming
- Initial load moves very large volumes of data
- The business conditions determine the refresh cycles

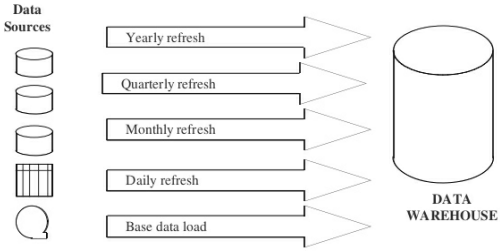

Figure 2-7 Data movements to the data warehouse.

substantial amounts of time. As the data warehouse starts functioning, you continue to extract the changes to the source data, transform the data revisions, and feed the incremental data revisions on an ongoing basis. Figure 2-7 illustrates the common types of data movements from the staging area to the data warehouse storage.

Data Storage Component

The data storage for the data warehouse is a separate repository. The operational systems of your enterprise support the day-to-day operations. These are online transaction processing applications. The data repositories for the operational systems typically contain only the current data. Also, these data repositories contain the data structured in highly normalized formats for fast and efficient processing. In contrast, in the data repository for a data warehouse, you need to keep large volumes of historical data for analysis. Further, you have to keep the data in the data warehouse in structures suitable for analysis, and not for quick retrieval of individual pieces of information. Therefore, the data storage for the data warehouse is kept separate from the data storage for operational systems.

In your databases supporting operational systems, the updates to data happen as transactions occur. These transactions hit the databases in a random fashion. How and when the transactions change the data in the databases is not completely within your control.

The data in the operational databases could change from moment to moment. When your analysts use the data in the data warehouse for analysis, they need to know that the data is stable and that it represents snapshots at specified periods. As they are working with the data, the data storage must not be in a state of continual updating. For this reason, the data warehouses are “read-only” data repositories.

Generally, the database in your data warehouse must be open. Depending on your requirements, you

are likely to use tools from multiple vendors. The data warehouse must be open to different tools. Most of the data warehouses employ relational database management systems. Many of the data warehouses also employ multidimensional database management systems.

Information Delivery Component

Who are the users that need information from the data warehouse? The range is fairly comprehensive. The novice user comes to the data warehouse with no training and, therefore, needs prefabricated reports and preset queries. The casual user needs information

once in a while, not regularly. This type of user also needs prepackaged information. The business analyst looks for ability to do complex analysis using the information in the data warehouse. The power user wants to be able to navigate throughout the data warehouse, pick up interesting data, format his or her own queries, drill through the data layers, and create custom reports and ad hoc queries.

In order to provide information to the wide community of data warehouse users, the information delivery component includes different methods of information delivery. Figure 2-8 shows the different information delivery methods. Ad hoc reports are predefined reports primarily meant for novice and casual users. Provision for complex queries, multidimensional (MD) analysis, and statistical analysis cater to the needs of the business analysts and power users. Information fed into Executive Information Systems (EIS) is meant for senior executives and high-level managers. Some data warehouses also provide data to data-mining applications.

In your data warehouse, you may include several information delivery mechanisms.

Most commonly, you provide for online queries and reports. The users will enter their requests online and will receive the results online. You may set up delivery of scheduled reports through e-mail or you may make adequate use of your organization’s intranet for information delivery.

For example, in a data warehouse containing units of sale, the quantity stored in each file record or table row relates to a specific time element. Depending on the level of the details in the data warehouse, the sales quantity in a record may relate to a specific date, week, month, or quarter.

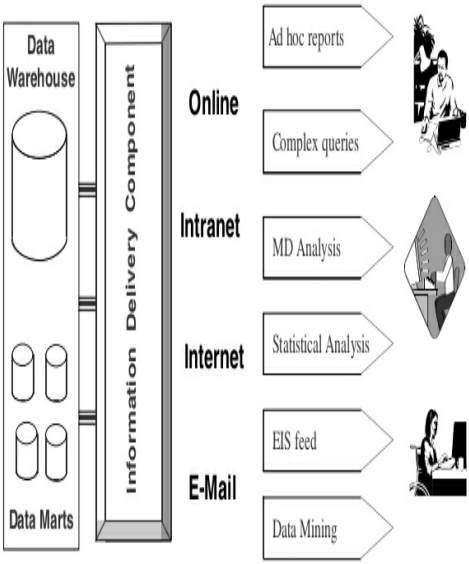

Figure 2-8 Information delivery component.

Metadata Component

Metadata in a data warehouse is similar to the data dictionary or the data catalog in database management system. In the data dictionary, you keep the information about thelogical data structures, the information about the files and addresses, the information about the indexes, and so on. The data dictionary contains data about the data in the database.

This definition is a commonly used definition. We need to elaborate on this definition.

Metadata in a data warehouse is similar to a data dictionary, but much more than a data dictionary. Later, in a separate section in this chapter, we will devote more time for the discussion of metadata. Here, for the sake of completeness, we just want to list metadata as one of the components of the data warehouse architecture.

Management and Control Component

This component of the data warehouse architecture sits on top of all the other components. The management and control component coordinates the services and activities within the data warehouse. This component controls the data transformation and the data transfer into the data warehouse storage. On the other hand, it moderates the information delivery to the users. It works with the database management systems and enables data to be properly stored in the repositories. It monitors the movement of data into the staging area and from there into the data warehouse storage itself. The management and control component interacts with the metadata component to perform the management and control functions. As the metadata component contains information about the datawarehouse itself, the metadata is the source of information for the management module.