COM 213 Database Concepts

Zablon Okari

September 2020

CHAPTER ONE

INTRODUCTION

Today, the success of an organization depends on its ability to acquire accurate and timely data about its operations, to manage this data effectively, and to use it to analyze and guide its activities.

The amount of information available to us is literally exploding, and the value of data as an organizational asset is widely recognized. Yet without the ability to manage this vast amount of data, and to quickly find the information that is relevant to a given question, as the amount of information increases, it tends to become a distraction and a liability, rather than an asset. This paradox drives the need for increasingly powerful and flexible data management systems. To get the most out of their large and complex datasets, users must have tools that simplify the tasks of managing the data and extracting useful information in a timely fashion. Otherwise, data can become a liability, with the cost of acquiring it and managing it far exceeding the value that is derived from it. A database is a collection of persistent data. The purpose of database is to store information about certain types of objects termed entities or objects.

A database represents some aspects of the real world sometimes called the miniworld or the universe of discourse. Changes to the miniworld are reflected in the database.

A database is a collection of data, typically describing the activities of one or more related organizations. For example, a university database might contain information about the following:

Entities such as students, faculty, courses, and classrooms.

Relationships between entities, such as students' enrollment in courses, faculty teaching courses, and the use of rooms for courses.

A database management system, or DBMS, is software designed to assist in maintaining and utilizing large collections of data, and the need for such systems, as well as their use, is growing rapidly. The alternative to using a DBMS is to use adhoc approaches that do not carry over from one application to another; for example, to store the data in files and write application-specific code to manage it. The use of a DBMS has several important advantages. A

database is designed, built and populated with data for a specific purpose. It has an intended group of users and preconceived applications in which these users are interested.

Database management system is a combination of computer hardware and software designed to collect, organize, store, manipulate and analyze data.

Examples of commercial available microcomputer relational DBMS includes: -

- Lotus
- Borland paradox
- Borland Dbase
- Oracle
- File maker pro
- Microsoft Access

Data management, which focuses on data collection, storage and retrieval, constitutes a core activity for any organization. To generate relevant information efficiently you need quick access to data (raw facts) from which the required information is produced. Efficient data management requires the use of a computer database. A database is a shared, integrated computer structure that houses a collection of: -

- User data i.e. raw facts of interest to the user.
- Meta data i.e. data about data through which the data is integrated. The Meta data provides a description of the data characteristics and the set of relationships that link the data found within the database. The database resembles a very well organized electronic filing cabinet in which powerful software referred to as DBMS helps manage the cabinet's contents.

A HISTORICAL PERSPECTIVE

From the earliest days of computers, storing and manipulating data have been a major application focus. The first general-purpose DBMS was designed by Charles Bachman at General Electric in the early 1960s and was called the Integrated Data Store. It formed the basis for the network data model, which was standardized by the Conference on Data Systems Languages (CODASYL) and strongly influenced database systems through the 1960s. Bachman was the first recipient of ACM's Turing Award (the computer science equivalent of a Nobel prize) for work in the database area; he received the award in 1973.

In the late 1960s, IBM developed the Information Management System (IMS) DBMS, used even today in many major installations. IMS formed the basis for an alternative data representation framework called the hierarchical data model. The SABRE system for making airline reservations was jointly developed by American Airlines and IBM around the same time, and it allowed several people to access the same data through a computer network. Interestingly, today the same SABRE system is used to power popular Web-based travel services such as Travelocity!

In 1970, Edgar Codd, at IBM's San Jose Research Laboratory, proposed a new data representation framework called the relational data model. This proved to be a water- shed in the development of database systems: it sparked rapid development of several DBMSs based on the relational model, along with a rich body of theoretical results that placed the field on a firm foundation. Codd won the 1981 Turing Award for his seminal work. Database systems matured as an academic discipline, and the popu- larity of relational DBMSs changed the commercial landscape. Their benefits were widely recognized, and the use of DBMSs for managing corporate data became stan- dard practice.

In the 1980s, the relational model consolidated its position as the dominant DBMS paradigm, and database systems continued to gain widespread use. The SQL query language for relational databases, developed as part of IBM's System R project, is now the standard query language. SQL was standardized in the late 1980s, and the current standard, SQL-92, was adopted by the American National Standards Institute (ANSI) and International Standards Organization (ISO). Arguably, the most widely used form of concurrent programming is the concurrent execution of database programs (called transactions). Users write programs as if they are to be run by themselves, and the responsibility for running them concurrently is given to the DBMS. James Gray won the 1999 Turing award for his contributions to the field of transaction management in a DBMS.

In the late 1980s and the 1990s, advances have been made in many areas of database systems. Considerable research has been carried out into more powerful query lan- guages and richer data models, and there has been a big emphasis on supporting complex analysis of data from all parts of an enterprise. Several vendors (e.g., IBM's DB2, Oracle 8, Informix UDS) have extended their systems with the ability to store new data types such as images and text, and with the ability to ask more complex queries. Specialized systems have been developed by numerous vendors for creating data warehouses, consolidating data from several databases, and for carrying out spe- cialized analysis.

An interesting phenomenon is the emergence of several enterprise resource planning (ERP) and management resource planning (MRP) packages, which add a substantial layer of application-oriented features on top of a DBMS. Widely used packages include systems from Baan, Oracle, PeopleSoft, SAP, and Siebel. These packages identify a set of common tasks (e.g., inventory management, human resources planning, finan- cial analysis) encountered by a large number of organizations and provide a general application layer to carry out these tasks. The data is stored in a relational DBMS, and the application layer can be cus-

tomized to different companies, leading to loweroverall costs for the companies, compared to the cost of building the application layer from scratch.

Most significantly, perhaps, DBMSs have entered the Internet Age. While the first generation of Web sites stored their data exclusively in operating systems files, the use of a DBMS to store data that is accessed through a Web browser is becoming widespread. Queries are generated through Web-accessible forms and answers are formatted using a markup language such as HTML, in order to be easily displayed in a browser. All the database vendors are adding features to their DBMS aimed at making it more suitable for deployment over the Internet.

Database management continues to gain importance as more and more data is brought on-line, and made ever more accessible through computer networking. Today the field is being driven by exciting visions such as multimedia databases, interactive video, digital libraries, a host of scientific projects such as the human genome mapping effort and NASA's Earth Observation System project, and the desire of companies to consolidate their decision-making processes and mine their data repositories for useful information about their businesses. Commercially, database management systems represent one of the largest and most vigorous market segments. Thus the study of database systems could prove to be richly rewarding in more ways than one!

FILE SYSTEMS VERSUS A DBMS

To understand the need for a DBMS, let us consider a motivating scenario: A company has a large collection (say, 500 GB1) of data on employees, departments, products, sales, and so on. This data is accessed concurrently by several employees. Questions about the data must be answered quickly, changes made to the data by different users must be applied consistently, and access to certain parts of the data (e.g., salaries) must be restricted.

We can try to deal with this data management problem by storing the data in a collection of operating system files. This approach has many drawbacks, including the following:

We probably do not have 500 GB of main memory to hold all the data. We must therefore store data in a storage device such as a disk or tape and bring relevant parts into main memory for processing as needed.

Even if we have 500 GB of main memory, on computer systems with 32-bit ad- dressing, we cannot refer directly to more than about 4 GB of data! We have to program some method of identifying all data items.

1 A kilobyte (KB) is 1024 bytes, a megabyte (MB) is 1024 KBs, a gigabyte (GB) is 1024 MBs, a terabyte (TB) is 1024 GBs, and a petabyte (PB) is 1024 terabytes.

We have to write special programs to answer each question that users may want to ask about the data. These programs are likely to be complex because of the large volume of data to be searched.

We must protect the data from inconsistent changes made by different users ac- cessing the data concurrently. If programs that access the data are written with such concurrent access in mind, this adds greatly to their complexity.

We must ensure that data is restored to a consistent state if the system crashes while changes are being made.

Operating systems provide only a password mechanism for security. This is not sufficiently flexible to enforce security policies in which different users have per- mission to access different subsets of the data.

A DBMS is a piece of software that is designed to make the preceding tasks easier. By storing data in a DBMS, rather than as a collection of operating system files, we can use the DBMS's features to manage the data in a robust and efficient manner. As the volume of data and the number of users grow—hundreds of gigabytes of data and thousands of users are common in current corporate databases—DBMS support becomes indispensable.

Such a typical filing /processing system has the limitation of more and more files and application programs being added to the system at any time. Such a scheme has a number of major disadvantages:

1. Data redundancy and inconsistency - Since the files and application programs are created by different programmers over a long period of type, the files are likely to have different formats and the programs may be written in several programming languages. Moreover, the same piece of information may be duplicated in several files. This redundancy leads to higher storage and access costs. It may also lead to inconsistency i.e. the various copies of the same data may no longer agree.

2. Difficulty in accessing - Suppose that one of the bank officers needs to find out the names of all customers who live within the city's 78-phone code. The officer would ask the data processing department to generate such a list. Such a request may not have been anticipated while designing the system originally and the only options available are:-

3. Extract the data manually

   Write the necessary application; therefore do not allow the data to be accessed conveniently and efficiently

4. Data isolation - Since data is scattered in various files and files may be in different formats, it may be difficult to write new applications programs to retrieve the appropriate data.

5. Concurrent access anomalies - Interaction of concurrent updates may result in inconsistent data e.g. if 2 customers withdraw funds say 50/= and 100/= from an account at about the same time the result of the concurrent execution may leave the account in an incorrect state.

6. Security problems - Not every user of the database system should be able to access all the data. Since application programs

   are added to the system in an ad-hoc manner, it is difficult to enforce security constraints.

7. Integrity - The data value stored in the database must satisfy certain types of consistency constraints e.g. a balance of a bank account may never fall below a prescribed value e.g. 5,000/=. These constraints are enforced in a system by adding appropriate code in the various application programs. However, when new constraints are added there is need to change the other programs to enforce.

Evaluation of the DBMS

Unlike the file system with may separate and unrelated files, the database consists of logically related data store in a single data repository. The problems inherent in file systems make using the database system very desirable and therefore, the database represents a change in the way the end user data are stored accessed and arranged.

Types of Database Systems

1. Single User database systems

   This is a database system that supports one user at a time such that if user A is using the database, users B and C must wait until user A complete his or her database work. If a single user database runs on a personal computer it's called a desktop database.

2. Multi-user database

   This is a database that supports multiple users at the same time for relatively small number e.g. 50 users in a department the database is referred to as a workgroup database. While one, which supports many departments is called an enterprise database.

3. Centralized Database system

   This is a database system that supports a database located at a single site.

4. Distributed database system

   This is a database system that supports a database distributed across several different sites.

5. Transaction DBMS/Production DBMS

   This is a database system that supports immediate response transaction e.g. sale of a product.

6. Decision Support DBMS

   It focuses primarily on the production of information required to make a tactical or strategic decision at middle and high management levels.

Advantages of the Database Systems

1. Centralized Control - Via the DBA it is possible to enforce centralized management and control of data. This means that necessary modifications, which do not affect other application changes, meet the data independence DBMS requirement.

2. Reduction of redundancies - Unnecessary duplication of data is avoided effectively reducing total amount of data required, consequently the reduction of storage space. It also eliminates extra processing necessary to trace the required data in a large mass of data. It also eliminates inconsistencies. Any redundancies that exist in the DBMS are controlled and the system ensures that his multiple copies are consistent.

3. Shared data - In a DBMS, sharing of data under its control by a number of application programs and user is possible e.g. backups.

4. Integrity - Centralized control can also ensure that adequate checks are incorporated to the DBMS provide data integrity. Data integrity means that the data contained in the database is both accurate and consistent e.g. employee age must be between 28-25 years.

5. Security - Only authorized people must access confidential data. The DBA ensures that proper access procedures are followed including proper authentication schemes process that the DBMS and additional checks before permitting access to sensitive data. Different levels of security can be implemented for various types of data or operations.

6. Conflict Resolution - The DBA is in a position to resolve conflicting resolve conflicting requirements of various users and applications. It is by choosing the best file structure and access method to get optimum performance for the response. This could be by classifying applications into critical and less critical applications.

7. Data Independence - It involves both logical and physical independence logical data independence indicates that the conceptual schemes can be changed without affecting the existing external schemes. Physical data independence indicates that the physical storage structures/devices used

for storing the data would be changed without necessitating a change in the conceptual view or any of the external use.

Disadvantages of Database Systemss

1. Cost - in terms of: The DBMS - software
   - Purchasing or developing S/W
   - H/W
   - Workspace (disks for storage)
   - Migration (movement from tradition separate systems to an integrated one)
2. Centralization Problems

   You would require adequate backup in case of failure

   You would require increased severity of security breaches and disruption of operation of the organization because of downtimes and failures.
3. Complexity of Backup and recovery