

Introduction to Database Management System

04 August 2024 11:42

As the name suggest, the database management system consists of two parts. They are:

- 1) Database and
- 2) Management System

What is database?

To find out what database is, we have to start from data, which is basically building block of any dbms.

Data: Facts, figures, statistics. Etc. Having no particular meaning. e.g., (1, Srija, 19) Etc.

Record: Collection of related data items. Example in the above example, the three data items had no meaning. But if we organize them in the following way, then they collectively represent meaningful information.

Roll	Name	Age
1	Srija	19

Table or relation: Collection of related records.

Roll	Name	Age
1	Srija	19
2	Rajdeep	20
3	Tanveer	19

The columns of this relation are called **fields or attribute or domain**. The rows are called **tuples or records**.

Database: A collection of related relation. Consider the following collection of tables:

T1

Roll	Name	Age
1	Srija	19
2	Rajdeep	20
3	Tanveer	19

T2

Roll	Address
1	Hwh
2	Rjp
3	kmz

T3

Roll	Year
1	1
2	2
3	1

T4

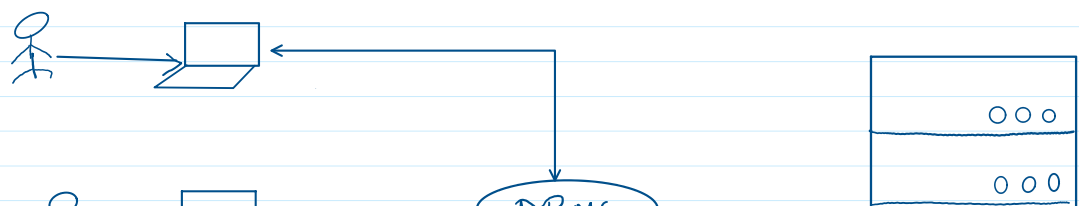
Year	Hostel
1	H1
2	H2
3	H3
4	H4

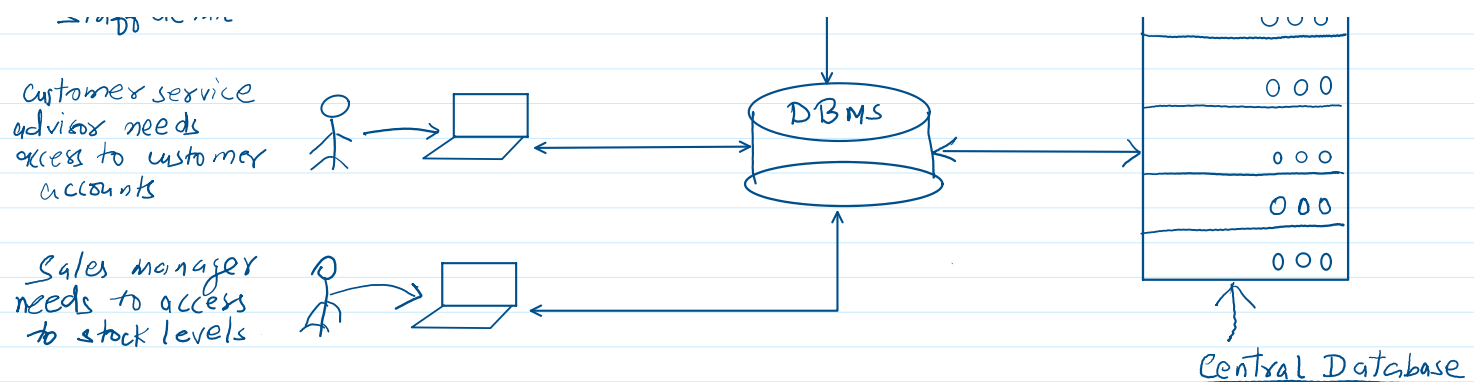
If a relation or table is a part of database, then there exist a common attribute with other table or relation.

A database in a DBMS could be viewed by lots of different people with different responsibilities.

Payroll officer
needs access to
staff detail.

Customer service





Employees are accessing Data through DBMS.

For example, within a company, there are different departments as well as customers who each need to see different kinds of data. Each employee in the company will have different levels of access to the database with their own customized **front-end** application.

Summary:

In a database data is organized strictly in row and column format. The rows are called tuple or record. The data items within one row may belong to different data types. On the other hand, the columns are often called domain or attribute. All the data items within a single attribute are of the same data type.

What is management system?

A **database management system or dbms**, is a collection of interrelated data and set of programs to access those data. This is a collection of related data with an implicit meaning and hence is a database. The collection of data usually referred to as the database contain information relevant to an enterprise. The primary goal of Dbms is to provide a way to store and retrieve database information that is both **convenient and efficient**. By data we mean known facts that can be recorded and that can have implicit meaning.

The management system is important because without the existence of some kind of rules and regulation, it is not possible to maintain the database. We have to select the particular attributes which should be included in a particular table. The common attributes to create relationship between 2 tables. If a new record has to be inserted or deleted, then which tables should be handled, Etc. These issues must be resolved by having some kinds of rules to follow in order to maintain the integrity of the database.

Because information is so important in most organization, computer scientists have developed a large body of concepts and techniques for managing data. These concepts and techniques we will focus in this whole course.

Database management system (DBMS) and its Applications:

A database management system is a computerized record keeping system. It is a repository or a container for a collection of computerized data files. The overall purpose of dbms is to allow the users to define store, retrieve and update the information contained in the database on demand. Information can be anything that is of significance to an individual or organization.

Database touch all aspects of our life. Some of the major areas of applications are as follows:

1. Banking
2. Railway Reservation System(IRCTC)
3. Airline Reservation System (centralized system) (makemytrip, flipkart, amazon, irctc, etc.)

4. Universities
5. Manufacturing and selling
6. Human Resource
And, Many more...

Enterprise Information

- **Sales:** For customer product and purchase information.
- **Accounting:** For payments, receipts, account balances, assets and other accounting information.
- **Human resource:** For information about employees, salaries, payroll taxes, benefits and for generation of paychecks.
- **Manufacturing:** For management of the supply chain and for tracking production of items in factories, inventories of items in warehouses and stores, and order for items.
- **Online retailers:** For sale data noted above online. order tracking generation of recommendation list and maintenance of online product evaluations.

Banking and finance

- **Banking:** For customer information accounts, loans and banking transactions.
- **Credit card transactions:** Four purchases on credit cards and general. of monthly statements.
- **Finance:** For restoring information about holding sales and purchase of financial instruments such as stocks and bonds. also for storing real time market data to enable online trading by customers and automated trading by the firm.

Universities: For student information course registration and grades (In addition to standard enterprise information such as human resource and accounting).

Airlines: Airlines were among the first to use databases in geographical distributed manner. For reservations and schedule information.

Telecommunication: For keeping records of calls made generating monthly bills, maintaining balances on prepaid calling cards and storing information about the communication networks.

Purpose of Database system

Database system arose in response to a. early methods of computerized management of commercial data. As an example of such methods, typical of the 1960s consider part of a university organization that, among other data, keeps information about all instructors, students, departments and course offering. One way to keep the information on the computer is to store it in the operating system files. to allow users to manipulate the information the system has a number of application programs that manipulates the file, including programs to:

- Add new students, instructors and courses.
- Register students for courses and generate class rosters.
- Assign grades to students compute grade point average (gpa) and generate transcripts.

System programmers wrote these application programs to meet the needs of the university. New application programs are added to the system as the need arises. For example, suppose that university decides to create a new major (say, computer science.) As a result, the university creates a new department and creates new permanent files. (or adds information to existing files.) to record information about all the instructors in the department. Student in that major course offerings degree requirements, Etc. The university may have to write new application programs to deal with rules specific to the new major. New application programs may also have to be written to handle new rules in the university. Thus, as time goes by, the system acquires more files and more application programs.

This typical file processing system is supported by a conventional operating system. The system stores permanent records in various files, and it needs different application programs to extract records from and add records to the appropriate files. Before database management system where? introduced organization. usually store information in such systems. Keeping organizational information in a file processing system has a number of major disadvantages:

Data redundancy and inconsistency: Since different programmers create. the files and

application programs over a long period of. various files are likely to have a different structures and programs may be written in several programming language. Moreover, the same information may be duplicated in several places(files). For example, if a student has a double major (say, music and mathematics) The address and telephone number of a student may appear in a file consisting of student records of student in the music department and in. a file that consists of student records of students in the mathematics department. the redundancy leads to higher storage and access cost. In addition, it may lead to data inconsistency, that is, the various copies of same data may no longer agree. For example, a change student address may be reflected in the music department records, but not elsewhere in the system.

Difficulty in accessing data: Suppose that one of the university clerks need to find out the name of all students who live within a particular **postal-code area**. The clerk asked the data processing department to generate such a list. because the designer of the original system did not anticipate this request. There is no application program on hand to meet it. There is, however, an application program to **generate the list of all students**.

The University Clerk has now two choices: Either obtain the list of all students and extract the needed information manually or ask. a programmer to write the necessary application program. both alternatives are obviously unsatisfactory. Suppose that such a program is written and that several days later the same clerk needs to trim that list to include only those students who have taken at least 60 credit hours as expected, a program to generate such a list does not exist. again, the clerk has the preceding two options. Neither of which is satisfactory. The point here is that conventional file processing involvement do not allow needed data to be retrieved in a conventional and efficient manner. More responsive data retrieval systems are required for general use.

Data isolation: Because data are scattered in various files And files may be in different formats. Writing new application program to retrieve the appropriate data is difficult.

Integrity problems: The data values stored in the database must satisfy certain types of **consistency constraints**. Suppose the unit. maintains an account for each department and records the balance amount in each account. Suppose also that the university requires that the account balance of a department may never fall below zero. Developers enforce these constraints in the system by adding appropriate code in the various application programs. However, when new constraints are added, it is difficult to change the program to enforce them. The problem is compounded when constraints involve several data items from different files.

Atomicity problems: A computer system like any other device is subject to failure. In many application, it is crucial that if a failure occurs, the data be restored to the consistent state that existed prior to the failure.

Consider a program to transfer ₹500. from the account balance of Department A. to the account balance of Department B. If a system failure occurs during the execution of the program, it is possible that the rupees 500 was removed from the balance of department A, but was not credited to the balance of Department B. resulting in an inconsistent database state. Clearly it is essential to database consistency that both the credit and the debit occur, or neither occur.

So that is the funds transfer must be atomic. It must happen in its entirely or not at all. It is difficult to ensure atomicity in a conventional file processing system.

Concurrent-access anomalies: For the sake of overall performance of the system and faster response, many systems allow multiple users to update the data simultaneously. Indeed, today, the largest Internet retailers may have millions of access per day to their data by shoppers. In such an environment, interaction of concurrent update is possible and may result in inconsistent data. Consider Department A with account balance of ₹10000. If two departments clerk debit the account balance (say, ₹500 and ₹100. respectively) of Department A Add almost exactly the same time, the result of the concurrent executions may leave the budget in an incorrect or inconsistent state. Suppose that the program is executing on behalf of each withdrawal. Read the old balance, reduce the value by amount being withdrawn, and write the result back. If the two programs run concurrently, they may both read the value ₹10000 and write back ₹ 9500 or ₹ 9900 respectively. Depending on which one writes the value last, The account balance of department a may contain either 9500 or 9900. rather than the correct value of ₹ 9400. to guard against this possibility, the system must maintain some form of

supervision.

But supervision is difficult to provide because data may be accessed by many different application programs that have not been coordinated previously.

Security problems: Not every user of a database system should be able to access all the data. For example, in a university payroll personal needs to see only the part of the database that his financial information. They do not need to access to information about academic records. But since application programs are added to the file processing system in an ad hoc manner, enforcing such security constraints is difficult.

These are difficulties among other prompted the development of database management system. In what follows, we shall see the concepts and algorithms that enable database systems to solve the problems with the file processing system.

Advantages of DBMS

Controlling of redundancy.

Improved data sharing.

Data integrity.

Security.

Data consistency.

Efficient data access.

Enforcement of standards.

Data independence.

Reduce application development and maintenance time.

Disadvantages of DBMS

1. It is a bit complex since. it supports multiple functionality to give the user the best. The underlying software has become complex. The designers and developers should have enough knowledge about the software to get the most out of it.
2. Because of its complexity and functionality, it uses large amount of memory. It also needs large memory to run efficiently.
3. DBMS system works on the centralized system. That is, all the users from the all over the world access this database. Hence, any failure of the dbms will impact all the users.
4. DBMS is generalized software. That is, it is written work on entire system, rather specific one. Hence, some of the application will run slow.