

Unit -1: Introduction to DBMS

Introduction to Database Management System

- **Data** are the raw facts that can be found after some experiment, observation or experience. Data itself do not provide any meaning but after processing it becomes **information**.
- The collection of related data organized in some specific manner is known as **database**. Data is organized into rows, columns and tables, and it is indexed to make it easier to find relevant information.
- A **database-management system (DBMS)** is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient.

So, DBMS contains information about a particular enterprise

—**Collection of interrelated data**

—**Set of programs to access the data**

—**An environment that is both *convenient* and *efficient* to use**

- Database systems are generally designed to manage large volume of information.
- Management of data involves both defining structures for storage of information and providing mechanisms for the manipulation of information. In addition, the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous results.
- DBMS can also define as a general purpose software system that is designed to define, manipulate, retrieve and manage data in a database.
- A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data.
- A DBMS relieves users of framing programs for data maintenance. Fourth- generation query languages, such as SQL, are used along with the DBMS package to interact with a database.
- It provides fast and convenient access to information from data stored in database. DBMS interfaces with application programs so data contained in database can be accessed by

multiple applications and users. Some popular DBMS softwares are: Oracle, SQL – Server, IBM-DB2, MySQL, MS Access, Sybase etc.

DBMS Vs. Flat File

DBMS	Flat File Management System
Multi-user access	It does not support multi-user access
Design to fulfill the need for small and large businesses	It is only limited to smaller DBMS system.
Remove redundancy and Integrity	Redundancy and Integrity issues
Expensive. But in the long term Total Cost of Ownership is cheap	It's cheaper
Easy to implement complicated transactions	No support for complicated transactions

Objective of DBMS

1. Eliminate redundant data.
2. Make access to the data easy for the user.
3. Provide for mass storage of relevant data.
4. Protect the data from physical harm and un-authorized systems.
5. Allow for growth in the data base system.
6. Make the latest modifications to the data base available immediately.
7. Allow for multiple users to be active at one time.
8. Provide prompt response to user requests for data.

Applications of DBMS

- Database systems are widely used. Here are some representative applications:





Sector	Use of DBMS
Banking	For customer information, account activities, payments, deposits, loans, etc.
Airlines	For reservations and schedule information.
Universities	For student information, course registrations, colleges and grades.
Telecommunication	It helps to keep call records, monthly bills, maintaining balances, etc.
Finance	For storing information about stock, sales, and purchases of financial instruments like stocks and bonds.
Sales	Use for storing customer, product & sales information.
Manufacturing	It is used for the management of supply chain and for tracking production of items. Inventories status in warehouses.
HR Management	For information about employees, salaries, payroll, deduction, generation of paychecks, etc.

Characteristics of database approach:

The main characteristics of database approach are discussed below:

1. Self describing nature of database system:

The fundamental characteristic of the database approach is that the database system contains not only the database itself but also complete definition or description of the database structure. This definition is stored in the DBMS catalog which contains the information such as the structure of each file, the type and storage format of each data item. Eg.

Field	Type	Collation	Attributes	Null	Default	Extra	
titel	varchar(200)	latin1_general_ci		Yes	<i>NULL</i>		
interpret	varchar(200)	latin1_general_ci		Yes	<i>NULL</i>		
jahr	int(11)			Yes	<i>NULL</i>		
<u>id</u>	bigint(20)		UNSIGNED	No		auto_increment	

Row Statistics		Space usage		
Statements	Value	Type	Usage	
Format	dynamic	Data	148	Bytes
Collation	latin1_general_ci	Index	2,048	Bytes
Rows	3	Total	2,196	Bytes
Row length σ	49			
Row size σ	732 Bytes			
Next Autoindex	5			
Creation	Oct 25, 2005 at 01:32 PM			
Last update	Oct 25, 2005 at 01:32 PM			

2. Insulation between programs and data :

In traditional file processing system, the structure of data file is embedded in the application program. Hence if we have to make any changes in the format of the data, the whole application program will also have to be changed.

But in the case of DBMS, database and the application program are separately situated. The structure of data files is stored in the DBMS catalog separately from the access programs. Hence in most cases DBMS access programs do not need such changes. We call this property as program-data independence.

For example, a file access program may be written in such a way that it can just access the name of a customer of any company but if we want to add another field, say, address of the customer, then in such a case this program will no longer be useful. But in the case of the DBMS, we just have to add the new field “address” in the catalog and the next time DBMS refers to the catalog, the new structure of the records will be accessed and hence we do not have to make change in the program.

3. Support of multiple view of the data:

Let us consider an example of student information system of a college, where all the data of student, courses, information of college etc are stored in a database. It is obvious that there is more than one user of this system and also their interest is different. i.e. student are generally interested in finding out the marks obtained, whereas teachers are able to put marks, view information about students, similarly, visitors are able to find the information of the college.

A database typically has many users, and their perspective of viewing the database is different. In the example given above, every user is accessing data from the same database but what they

see and can access is different. A view may be subset of the database or it may contain virtual data that is derived from the database files but is not stored explicitly.

4. Sharing of data and multi user transaction processing:

Database system should allow multiple users to access same database at the same time. For example, in online air ticket reservation system many users are accessing the same site at the same time. Hence for every seat, DBMS should ensure that only one user should be given access to reserve the seat.

Hence to ensure the correctness of this type of transaction DBMS must include concurrency control software to ensure that several users trying to update same data do so in a controlled manner so that the result of the updates is correct. In order to do so, DBMS uses lock based protocol and time stamp based protocol.

Purpose of Database Systems

Traditionally, file processing system was used to manage information. It stores data in various files of different application programs to extract or insert data to appropriate file. File processing system has several drawbacks due to which database management system is required. **Database management system removes problems found in file processing system.** Some major problems of file processing systems are:

- **Data redundancy and inconsistency**

In file processing system, different programmer creates files and writes application programs to access it. After a long period of time files may exist with different formats and application programs may written in many different programming languages. Moreover, same information may be duplicated in several files. We have to pay for higher storage and access cost for such redundancy. It may leads database in inconsistent state because update made in one file may reflected in one file but it may not reflected in another files where same information exist in another files.

- **Difficulty in accessing data**

In file processing system, we cannot easily access required data stored in particular file. For each new task we have to write a new application program. File processing system cannot allow data to be retrieve in convenient and efficient manner.

- **Integrity problem:**

In database, we required to enforce certain type consistency constraints to ensure the database correctness or to enforce certain business rules. It is in fact called integrity constraints (e.g. account balance > 0), integrity of database need not to be violated. In file processing system, integrity constraint becomes the part of application program. Programmer need to write appropriate code to enforce it. When new constraints are required to add or change existing one, it is difficult to change program to enforce it.

- **Atomicity problem:**

Failures may lead database in an inconsistent state with partial updates. For example, failure occurs while transferring fund from account A to B. There would be the case that certain amount from account A is retrieved and it is updated but failure occurs just before it is deposited to account B, such case may lead database in inconsistent state.

- **Concurrent access problem:**

Concurrent accessed increase the overall performance of system providing fast response time but uncontrolled concurrent accesses can lead inconsistencies in system. File processing system allow concurrent access but it is unable to coordinate different application programs so database may lead in inconsistent state. E.g. two people reading a balance and updating it at the same time.

- **Security problem:**

Since file processing system consist large no. of application programs and it is added in ad hoc manner. So it is difficult to enforce security to each application to allow accessing only part of data/database for individual database users.

Database management systems offer solutions to all the above problems

Importance of Database

- A database management system is important because it manages data efficiently and allows users to perform multiple tasks with ease.
- A database management system stores, organizes and manages a large amount of information within a single software application. Use of this system increases efficiency of business operations and reduces overall costs.
- Database management systems are important to businesses and organizations because they provide a highly efficient method for handling multiple types of data.
- Some of the data that are easily managed with this type of system include: employee records, student information, payroll, accounting, project management, inventory and library books. These systems are built to be extremely versatile.
- Without database management, tasks have to be done manually and take more time. Data can be categorized and structured to suit the needs of the company or organization.
- Data is entered into the system and accessed on a routine basis by assigned users. Each user may have an assigned password to gain access to their part of the system. Multiple users can use the system at the same time in different ways

Merits and Demerits of DBMS

Advantages of DBMS

- **Data independence:** DBMS provides abstract view of data. Application programs are independent from details of data representation and storage. 3
- **Efficient data access:** DBMS provides verity of sophisticated techniques to store and retrieve data efficiently.
- **Data integrity and security:** DBMS allow to enforce integrity constraints on data. For example before inserting salary information for an employee, DBMS can enforce integrity constraint to check salary is not exceeded department budget. DBMS can also enforce access controls, what data is visible to what class of users.

- **Data administration:** DBMS provides centralized administration of data. It is appropriate when several no. of database user shares data. It improves the overall performance of database system.
- **Concurrent access and crash recovery:** DBMS has a capability manage concurrent access. It schedules concurrent access to the data in such a manner that user feel data is being accessed by only one user at a time. Moreover, DBMS protects users from the effects of system failures.
- **Reduced application development time:** since DBMS supports many important functions that are common to many applications accessing data stored in database. It provides high level interface to data and facilitates quick development of applications.

Disadvantages of DBMS

- Cost of Hardware and Software of a DBMS is quite high which increases the budget of your organization.
- Most database management systems are often complex systems, so the training for users to use the DBMS is required.
- In some organizations, all data is integrated into a single database which can be damaged because of electric failure or database is corrupted on the storage media
- Use of the same program at a time by many users sometimes lead to the loss of some data.
- DBMS can't perform sophisticated calculations
- Abstract view of data presented by DBMS may not match for certain applications. For example, relational databases does not supports flexible analysis of text data
- If specialized performance or data manipulation requirements are central to an application, DBMS is not appropriate for such application. The added benefits of a DBMS (e.g. flexible querying, security, concurrent access and crash recovery) may not require for applications.
-