Database Management System

Unit I

Introduction:

A **database** is a collection of related data and data is a collection of facts and figures that can be processed to produce information. For example, consider the names, telephone numbers, and addresses, and addresses of the people you know. You may have recorded this data in an indexed address book or you may have stored it on a hard drive, using a personal computer and software such as Microsoft Access or Excel. This collection of related data with an implicit meaning is a database.

A **Database Management System (DBMS)** is a collection of programs that enables users to create and maintain a database. The DBMS is a *general-purpose software system* that facilitates the processes of ***defining***, ***constructing***, ***manipulating***, and ***sharing*** databases among various users and applications.

Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition or descriptive information is also stored in the database in the form of a database catalog or dictionary is called **meta-data.**

Constructing a database is the process of storing the data on some storage medium that is controlled by the DBMS.

Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data.

Sharing a database allows multiple users and programs to access the database simultaneously.

Characteristics of Database Approach:

The main characteristics of the database approach are as follows:

1. Self-describing nature of a database system
2. Insulation between programs and data, and data abstraction
3. Support of multiple views of the data
4. Sharing of data and multiuser transaction processing

Self-describing nature of a database system:

A fundamental characteristic of the database approach is that the database system contains not only the database itself but also a complete definition or description of the database structure and constraints. This definition is stored in the DBMS catalog, which contains information such as the structure of each file, the type and storage format of each data item, and various constraints on the data. The information stored in the catalog is called **meta-data**, and it describes the structure of the primary database.

Insulation between programs and data, and data abstraction:

In Database Management System, the structure of data files is embedded in the application programs, so any changes to the structure of a file do not require changing all programs that access the database. The structure of data files is stored in the DBMS catalog separately from the access programs. This property is called **program-data independence.**

The characteristic that allows program-data independence and program-orientation independence is called **data abstraction.**

Support of Multiple Views of the Data:
A database typically has many users, each of whom may require a different perspective or view of the database. A multiuser DBMS whose users have a variety of distinct applications must provide facilities for defining views. For example, one user of the student database may be interested only in accessing and printing the transcript of each student and another user may be interested only in checking which course the students enrolled.

Sharing of Data and Multiuser Transaction Processing:
A multiuser DBMS, as its name implies, must allow multiple users to access the database at the same time. This is essential if the data for multiple applications is to be integrated and maintained in a single database. The DBMS must include **concurrency control** software to ensure that several users trying to update the same data do in a controlled manner so that the result of the updates is correct. For example, when several reservation clerks try to assign a seat on a particular train, the DBMS should ensure that each seat can be accessed by only one clerk at a time for assignment to a passenger.

People Associated with DBMS:

1. **Database Administrators:**
   Database Administrator is a person who oversees and manages the resources such as the database itself, DBMS, and related software. The Database Administrator (DBA) is responsible for authorizing access to the database, coordinating and monitoring its use, and acquiring software and hardware resources.

2. **Database Designers:**
   Database Designers are responsible for identifying the data to be stored in the database and for choosing appropriate structures to represent and store the data. These tasks are mostly undertaken before the database is actually implemented and populated with data. It is the responsibility of database designers to communicate with all perspective database users in order to understand their requirements and to create a design that meets these requirements.

3. **End Users:**
   End users are the people whose jobs require access to the database for querying, updating, and generating reports. There are several categories of end users.
   - **Casual end users** occasionally access the database, but they may need different information each time.
   - **Naive** or **parametric end users** main job function involves around constantly querying and updating the database, using standard types of queries and updates. Example: Bank clerks, Reservation clerks etc.

- **Sophisticated end users** include engineers, scientists, business analysts and others who thoroughly familiarize themselves with the facilities of the DBMS in order to implement their applications to meet their requirements.
- **Standalone users** maintain personal database by using ready-made program packages that provide easy-to-use menu-based or graphics-based interfaces.

4. **System Analysts and Application Programmers (Software Engineers):**
   System analysts determine the requirements of end users, especially naïve and parametric end users, and develop specifications for transactions that meet these requirements.
   Application programmers implement these specifications and programs and then they test, debug, document.

Advantages of the DBMS Approach.

1. **Controlling Redundancy:**
   The Database Management System will not allow storing the redundant data (same data multiple times) in the database.
   The **redundancy** in storing the same data multiple times leads to several problems. The problems with the storage of redundant data are as follows. First, *duplication of effort* (multiple times data has to be recorded). Second, *storage space* is wasted when the same data is stored repeatedly. Third, files that represent the same data may become inconsistent and etc.

2. **Restricting Unauthorized Access:**
   The Database Management System restricts unauthorized access of the database by enforcing restrictions such as providing a password for an authorized person to access the data. In addition, some users may only be permitted to retrieve data, whereas others are allowed to retrieve and update.

3. **Providing Persistent Storage for Program Objects:**
   Databases can be used to provide **persistent storage** (continue to exist for a long time and preserve data or remain unchanged) for program objects and data structures. This is one of the main reasons for object-oriented database systems. Object-oriented database systems are compatible with programming languages such as C++ and Java, and the DBMS software automatically performs any necessary conversions.

4. **Providing Storage Structures for Efficient Query Processing:**
   Database systems must provide capabilities for efficiently executing queries and updates. Because the database is typically stored on disk, the DBMS must provide specialized data structures to speed up disk search for the desired records.
   The **query processing and optimization** module of the DBMS is responsible for choosing an efficient query execution plan for each query based on the existing storage structure.

5. **Providing Backup and Recovery:**
   A DBMS must provide facilities for recovering from hardware or software failures. The **backup and recovery subsystem** of the DBMS is responsible for recovery. For example, if the computer system fails in the middle of a complex update transaction, the recovery subsystem is responsible for making sure that the database is restored to the state is was in before the transaction started executing.

6. **Providing Multiple User Interfaces:**
   A DBMS should provide a variety of user interfaces (Graphical User Interfaces), because many types of users with varying levels of technical knowledge use a database. These include query languages for casual users, programming language interfaces for application programmers, forms and command codes for parametric users, and menu-driven interfaces and natural language interfaces for standalone users.

7. **Representing Complex Relationships among Data:**
   A database may include numerous varieties of data that are interrelated in many ways. A DBMS must have the capability to represent a variety of complex relationships among the data, to define new relationships as they arise, and to retrieve and update related data easily and efficiently.

8. **Enforcing Integrity Constraints:**
   Most database applications have certain integrity constraints that must hold for the data. A DBMS should provide capabilities for defining and enforcing these constraints. The simplest type of integrity constraint involves specifying a data type for each data item. For example, student's marks should be between 0 and 100, and an employee age between 20 and 60 etc.

When Not to Use a DBMS:

1. If unnecessary overhead costs are incurred as compared to traditional file processing.
2. If high initial investment in hardware, software, and training is needed.
3. If providing security, concurrency control, recovery, and integrity functions is difficult compared to traditional file processing.
4. If the existing database and applications are simple, well defined, and not expected to change.
5. If No multiple-user access to data.