CHAPTER - 1
Introduction

1.1 Concept and Application

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. Name of a student, age, class and her subjects can be counted as data for recording purposes. Mostly data represents recordable facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks etc.

A **database** is a collection of data, typically describing the activities of one or more related organizations. A **database system** is a computer generated software program which can be used to access the data stored in database in an organized manner. The term *database* is a structured collection of data stored in digital form. Before the actual data is stored in the database, we should clearly specify the schema of the database and different techniques used to manipulate the data stored in a database.

A **database management system** (DBMS) is a collection of interrelated data (i.e. database) and a set of programs or software to access those data. Primary goal of DBMS is to provide a way to store and retrieve database information that is both convenient and efficient. DBMS also provides safety for the information stored despite system crashes or attempts of authorized access only.

DBMS can also define as a general purpose software system that enables user to create, maintain and manipulate 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 software are: Oracle, SQL – Server, IBM-DB2, MySQL, MS Access, Sybase etc.

Application Areas of Database System

Some of the most common application of database management system are:

- **Airlines and railways:** Airlines and railways use online databases for reservation, and for displaying the schedule information.
- **Banking:** Banks use databases for customer inquiry, accounts, loans, and other transactions.
- **Education:** Schools, colleges and universities use databases for course registration, result, and other information.
- **Telecommunications:** Telecommunication departments use databases to store information about the communication network, telephone numbers, record of calls, for generating monthly bills, etc.
- **Credit card transactions:** Databases are used for keeping track of purchases on credit cards in order to generate monthly statements.
- **E-commerce:** Integration of heterogeneous information sources for business activity such as online shopping, booking of holiday package, consulting a doctor, etc.
- **Health care information systems and electronic patient record:** Databases are used for maintaining the patient health care details.
- **Digital libraries and digital publishing:** Databases are used for management and delivery of large bodies of textual and multimedia data.
- **Finance:** Databases are used for storing information such as sales, purchases of stocks and bonds or data useful for online trading.
- **Sales:** Databases are used to store product, customer and transaction details.
- **Human resources:** Organizations use databases for storing information about their employees, salaries, benefits, taxes, and for generating salary checks.

1.2 Objective and Evolution

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:

1. **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.

2. **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.

3. **Data isolation**
   Since data are scatter in different files and data may store in different format, so it is difficult to write program to retrieve appropriate data.

4. **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.

5. **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.

6. **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.

7. **Security problems**
   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.

Evolution to DBMS

Traditionally data was organized in file formats. DBMS was all new concepts then and all the research was done to make it to overcome all the deficiencies in traditional style of data management. Modern DBMS has the following characteristics:

- **Real-world entity**: Modern DBMS are more realistic and uses real world entities to design its architecture. It uses the behavior and attributes too. For example, a school database may use student as entity and their age as their attribute.

- **Relation-based tables**: DBMS allows entities and relations among them to form as tables. This eases the concept of data saving. A user can understand the architecture of database just by looking at table names etc.

- **Isolation of data and application**: A database system is entirely different than its data. Where database is said to active entity, data is said to be passive one on which the database works and organizes. DBMS also stores metadata which is data about data, to ease its own process.

- **Less redundancy**: DBMS follows rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Following normalization, which itself is a mathematically rich and scientific process, make the entire database to contain as less redundancy as possible.

- **Consistency** (*Correctness*): DBMS always enjoy the state on consistency where the previous form of data storing applications like file processing does not guarantee this. Consistency is a state where every relation in database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state.

- **Query Language**: DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many and different filtering options, as he or she wants. Traditionally it was not possible where file-processing system was used.

- **ACID Properties**: DBMS follows the concepts for ACID properties, which stands for Atomicity, Consistency, Isolation and Durability. These concepts are applied on transactions, which manipulate data in database. ACID properties maintains database in healthy state in multi-transactional environment and in case of failure.

- **Multiuser and Concurrent Access**: DBMS support multi-user environment and allows them to access and manipulate data in parallel. Though there are restrictions on transactions when they attempt to handle same data item, but users are always unaware of them.

- **Multiple views**: DBMS offers multiples views for different users. A user who is in sales department will have a different view of database than a person working in production department. This enables user to have a concentrate view of database according to their requirements.

- **Security**: Features like multiple views offers security at some extent where users are unable to access data of other users and departments. DBMS offers methods to impose constraints while entering data into database and retrieving data at later stage. DBMS offers many different levels of security features, which enables multiple users to have different view with different features. For example, a user in sales department cannot see data of purchase department is one thing, additionally how much data of sales department he can see, can also be managed. Because DBMS is not saved on disk as traditional file system it is very hard for a thief to break the code.

1.3 Need of DBMS

The DBMS become a importance for well-run of an modern enterprise because of the following facts:

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

1.4 Data Abstraction

A major purpose of a database system is to provide users with and abstract view of the data i.e. the system hides certain details of how the data are stored and maintained. It gives architecture to separate the user applications and the physical database. It defines in the following three levels

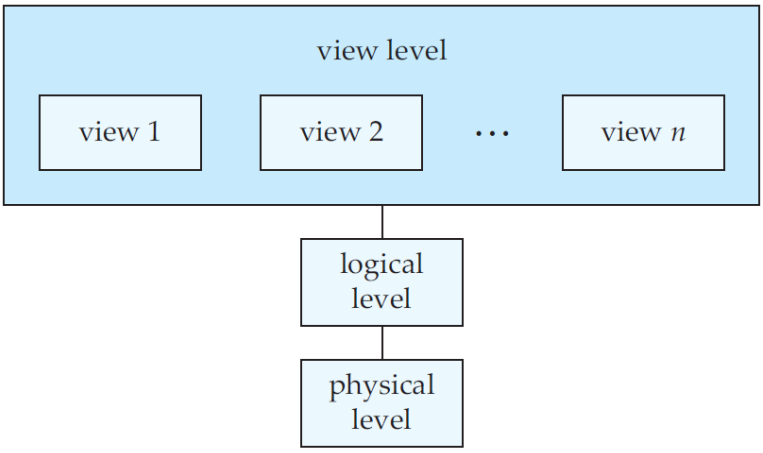

Figure The three levels of data abstraction.

1. **Physical Level:** The lowest level of abstraction describes how the data are actually stored. The physical level describes complex low-level data structures in detail.
2. **Conceptual level or logical level**: - Logical levels hide the detail of physical storage and describe about the entities data and constraints. It is the middle level and it is used by the application programmer it describes what data are stored in the database and what relationship among those data. All logical tasks are performed at this level.
3. **View Level:** The highest level of abstraction describes only part of the entire database. Many users of the database system do not need all this information; instead, they need to access only a part of the database. The view level of abstraction exists to simplify their interaction with the system. The system may provide many views for the same database.

Example:

```
View level
   • CS Majors
   • Math Majors
Logical level: entire database schema
   • Courses (CourseNo,CourseName,Credits,Dept)
   • Student (StudentID,Lname,Fname,Level,Major)
   • Grade (StudentID,CourseNo,mark)
Physical level:
   • How these tables are stored, how many bytes it required
     etc.
```

1.5 Data independence

Data independence is an ability to modify a schema definition in one level without affecting scheme definition in higher level. The concept of data independence is similar in many respects to the concept of abstract data type in modern programming languages like C++. Both hide implementation details from the users. This allows users to concentrate on the general structure rather than low-level implementation details. There are two types of data independence.

a. Physical data independence

Physical data independence allows changes in the physical storage devices or organization of the files to be made without requiring changes in the conceptual view or any of the external views and hence in the application programs using the database. Thus, the files may migrate from one type of physical media to another or the file structure may change without any need for changes in the application programs.

b. Logical data independence

Logical data independence implies that application programs need not be changed if fields are added to an existing record; nor do they have to be changed if fields not used by application programs are deleted. Logical data independence indicates that the conceptual schema can be changed without affecting the existing external schemas. Logical data independence is more difficult to achieve than physical independence. Since application programs are heavily dependent on the logical structure of the data they access.

1.6 Schema and Instances

Database change over time as information is inserted, deleted and updated. The collection of information stored in database at a particular moment called an instance of the database. The overall design of the database is called database schema. Schemas are change infrequently, if at all. The schema and instance can be compared with a program written in a programming language. The database schema is similar to a variable declared along with the type description in a program. The variable contains a value at a given point of time. This value of a variable corresponds to an instance of a database schema.

According to the level of abstraction schema are divided into **physical schema**, **logical schema** and **subschemas**. The physical schema describes the database design at the physical level. Logical schema describes database design at the logical level. Database system may have several schemas at the view level, it is called sun-schemas (can be query), it describes different views of database. Logical schema is more important for the development of application programs. Programmer constructs applications by using logical schema. The physical schema is hidden under the logical schema and it can change without affecting application programs.

1.7 Concept of DDL, DML and DCL

A database system provides a data-definition language (DDL) to specify the database schema, a data-manipulation language (DML) to express database queries and updates, and data-control language (DCL) to control the database transactions.

a. Data Definition Language

Database schema can be specified by a set of definitions expressed by data-definition language. The DDL can be used to specify additional properties of the data such as storage structure and access method used by the database system, such special set of DDL statement called *data storage and definition* language.

For ***example***, following DDL statement in SQL defines account relation.

```
create table account
(
  account_no   char(2),
  balance          integer
)
```

The execution of above DDL statement creates table account. Moreover, it updates special set of tables called data dictionary or data directory. Data dictionary contains meta data, that is data about data. For example table containing tables' information like table name, owner, created date, modified date etc. refers data dictionary and contain information are example of meta data.

b. Data-Manipulation Language (DML)

A data-manipulation language (DML) is a language that enables users to manipulate data as organized by the appropriate data model. The types of access are:

- Retrieval of information stored in the database
- Insertion of new information into the database
- Deletion of information from the database
- Modification of information stored in the database

DML established communication between user and database. There are two types of DML

(a) *Procedural DML:* user required to specify what data are needed and how they get those data.

(b) *Nonprocedural (Declarative) DML:* user only required to what data needed without specifying how to get those data.

Declarative DMLs are usually easier to learn and use than procedural DMLs. However, since a user does not have to specify how to get data, the database system has to figure out an efficient means of accessing data. The DML component of SQL is nonprocedural.

A query is statement requesting the retrieval of information. Special set of DML which only use to retrieve information from database called *query language*.

Example:

```
Select customer_name
        from customer
where customer.customer_id='c001'
```

This query retrieves those rows from table customer where the customer_id=c001.

c. Data Control Language (DCL)

The Data Control Language (DCL) is a subset of the Structured Query Language (SQL) that allows database administrators to configure security access to relational databases. It complements the Data Definition Language (DDL) and the Data Manipulation Language (DML). It consists of only three commands: GRANT, REVOKE, and DENY. These three commands provide administrators with the flexibility to set and remove database permissions.

1.8 Database Manager and Users

Database Manager

The database manager is a program module which provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system. Since database required lots of storage space so it must be stored on disks. Data need to move between disk and main memory as needed.

Since the goal of database system is to simplify and facilitate access to data providing optimal performance as far as possible. So the database manager module is responsible for:

- ***Interaction with the file manager:*** responsible to translate DML statements into low-level file system commands for storing, retrieving and updating data in the database.
- ***Integrity enforcement:*** responsible to check any updates in the database do not violate consistency constraints. (E.g. no bank account balance below $25).
- ***Security enforcement:*** responsible to ensure that users only have access to information they are permitted to see.
- ***Backup and recovery:*** Detecting failures due to power failure, disk crash, software errors, etc., and restoring the database to its state before the failure.
- ***Concurrency control:*** responsible to preserving data consistency when there are concurrent users.

Database administrator (DBA)

DBA is a person who has central control over both data and application programs. The responsibilities of DBA vary depending upon the job description and corporate and organization policies. Some of the responsibilities of DBA are given here.

- **Schema definition and modification**: The DBA creates the original database schema by executing a set of data definition statements in DDL. The DBA also carries out the changes to the schema according to the changing needs of the organization.
- **New software installation**: It is the responsibility of the DBA to install new DBMS software, application software, and other related software. After installation, the DBA must test the new software.
- **Security enforcement and administration:** DBA is responsible for establishing and monitoring the security of the database system. It involves adding and removing users, auditing, and checking for security problems.
- **Data analysis:** DBA is responsible for analyzing the data stored in the database, and studying its performance and efficiency in order to effectively use indexes, parallel query execution, etc.
- **Preliminary database design:** The DBA works along with the development team during the database design stage due to which many potential problems that can arise later (after installation) can be avoided.
- **Physical organization modification:** The DBA is responsible for carrying out the modifications in the physical organization of the database for better performance.
- **Routine maintenance checks:** The DBA is responsible for
  - Taking the database backup periodically in order to recover from any hardware or software failure or disasters.
  - Checking data storage and ensuring the availability of free disk space for normal operations, upgrading disk space as and when required.
  - Monitoring jobs running on the database and ensuring that performance is not degraded by very expensive tasks submitted by some users

Database Users

A primary goal of a database system is to retrieve information from and store new information in the database. Database users are those who interact with the database in order to query and update the database, and generate reports. There are four different types of database users, they are differentiated according to their interaction with the system. Moreover, there are different types of user interfaces for different types of users.

(a) Naïve Users: Naïve users (*general* or *simple users*) are unsophisticated users who interact with the system by invoking one of the application programs that are already written. For example, banks teller who needs to transfer fund from one account to another invoking a program called transfer. This program asks the teller for the amount of money to be transferred, and account to which the money is to be transferred.

The typical user interface for the native user is a form interface, where user can fill appropriate fields of the form. Native users may also simply read reports generated from the database.

(b) Application programmers: Application programmers are the computer professionals who implement the specifications, and develop application programs. They can choose tools, such as rapid application development (RAD) to develop the application program with minimal effort. The database application programmer develops application program to facilitate easy data access for the database users.

(c) Sophisticated users: Sophisticated user interact with system without writing programs but the requests by writing queries in database using DML query language. This query goes to query processor and it converted into instructions for the database manager module.

(d) Specialized users: Specialized users are responsible to write special database application programs it could be computer-aided design systems, knowledge based and expert systems that store data with complex data types (e.g. graphics data, audio/video data).

1.9 Advantages and Disadvantages of DBMS

Advantages:

- Reduced data redundancy
- Reduced updating errors and increased consistency
- Greater data integrity (*no-modification*) and independence from applications programs
- Improved data access to users through use of host and query languages
- Improved data security
- Reduced data entry, storage, and retrieval costs
- Concurrent access and crash recovery
- Facilitated development of new applications program

Disadvantages:

- Database systems are complex, difficult, and time-consuming to design
- Substantial hardware and software start-up costs
- Damage to database affects virtually all applications programs
- Extensive conversion costs in moving form a file-based system to a database system
- Initial training required for all programmers and users.