

# **Database Management System**

## **Module-1**

### **Introduction-Database Concept**

Prof. Suja Jayachandran

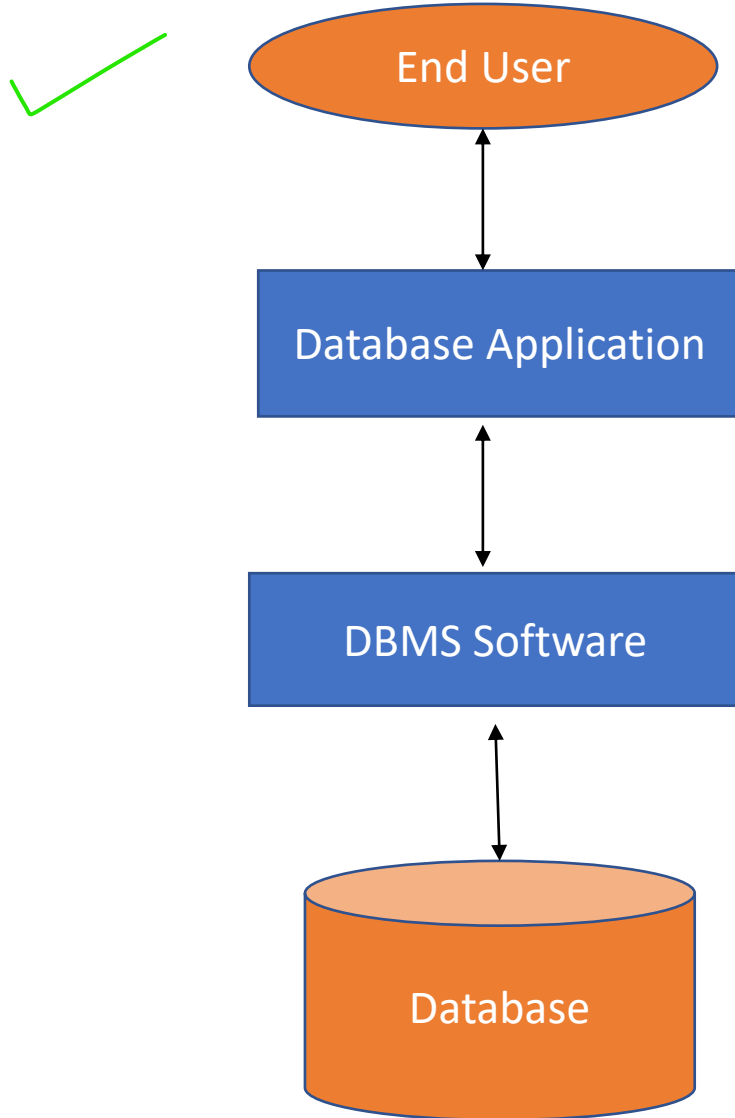
Assistant Professor  
Computer Engineering Department  
Vidyalkar Institute of Technology

# Introduction to Database Management System

## Basic Terminology

- **Data:** Raw or isolated facts about an entity. Eg.: Text, Image, Audio, Video
- **Information:** Processed, Meaningful, Usable data
- **Database:** Collection of similar/related Data
- **Database Management System(DBMS):** It is a collection of inter-related data and a set of programs to access and manipulate those data. Eg: MySQL, MS Access, Oracle, PostgreSQL etc.

# Component of DBMS



# Features of DBMS



# Difference between File Processing System and DBMS

	File Processing System	DBMS
Definition	A process that manages how and where a data in a storage disk is stored , accessed and managed.	An organized collection of data that can be easily accessed, managed and updated
Data Consistency	Has high Data <b>inconsistency</b>	Maintains data consistency
Structure	Simple	Complex
Data sharing	Difficult	easy
Redundancy	High redundancy	Low redundancy
Security	Not very secure	More secure
Backup and Recovery	Not efficient	Available
Example	NTFS and Ext	MySQL,Oracle,MS Access

- **Schema**

- **Logical Schema:** The overall logical structure of the database

Example : A database consisting information about a set of customer and accounts in a bank and the relationship between them.(Customer Schema will contain customer ID, Name, Account#)

- **Physical Schema:** the overall physical structure of the database.

- **Instance**

- The actual content of the database at a particular point of time.

Example: Customer Instance

Customer ID	Name	Account #
1	Ramesh	1212121212

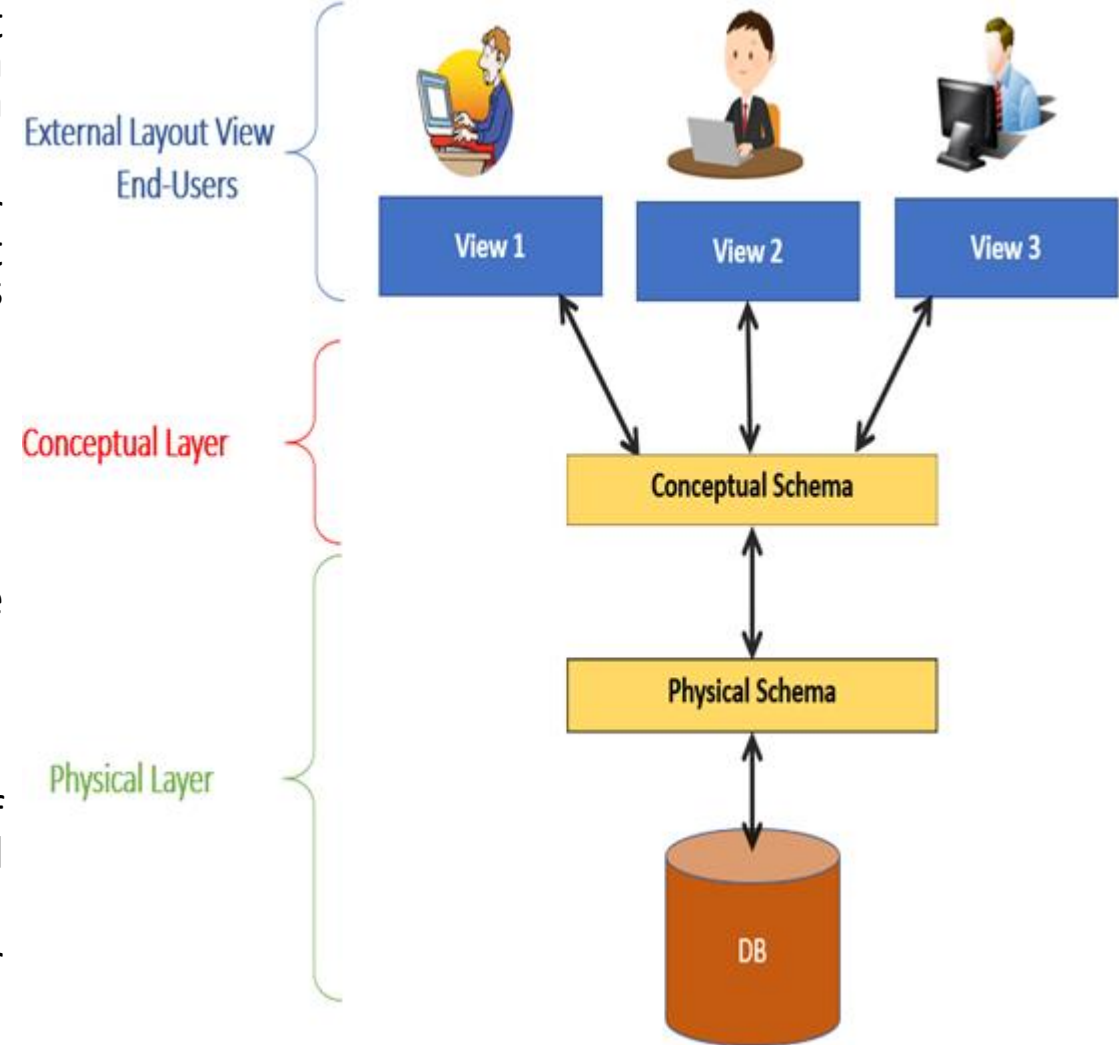
# Data Independence

Data Independence is defined as a property of DBMS that helps you to change the Database schema at one level of a database system without requiring to change the schema at the next higher level.

**Data Abstraction** is a process of hiding unwanted or irrelevant details from the end user. It provides a different view and helps in achieving data independence which is used to enhance the security of data

## • Importance of Data Independence

- Helps you to improve the quality of the data
- Database system maintenance becomes affordable
- Enforcement of standards and improvement in database security
- You don't need to alter data structure in application programs
- Permit developers to focus on the general structure of the Database rather than worrying about the internal implementation
- It allows you to improve state which is undamaged or undivided
- Easily make modifications in the physical level is needed to improve the performance of the system.



✓ 1. **Physical Data Independence:** It helps you to separate conceptual levels from the internal/physical levels. It allows you to provide a logical description of the database without the need to specify physical structures. Compared to Logical Independence, it is easy to achieve physical data independence.

Example:

- Using a new storage device like Hard Drive or Magnetic Tapes
- Modifying the file organization technique in the Database
- Switching to different data structures.
- Changing the access method.
- Modifying indexes.
- Changes to compression techniques or hashing algorithms.
- Change of Location of Database from say C drive to D Drive

## 2. Logical Data Independence

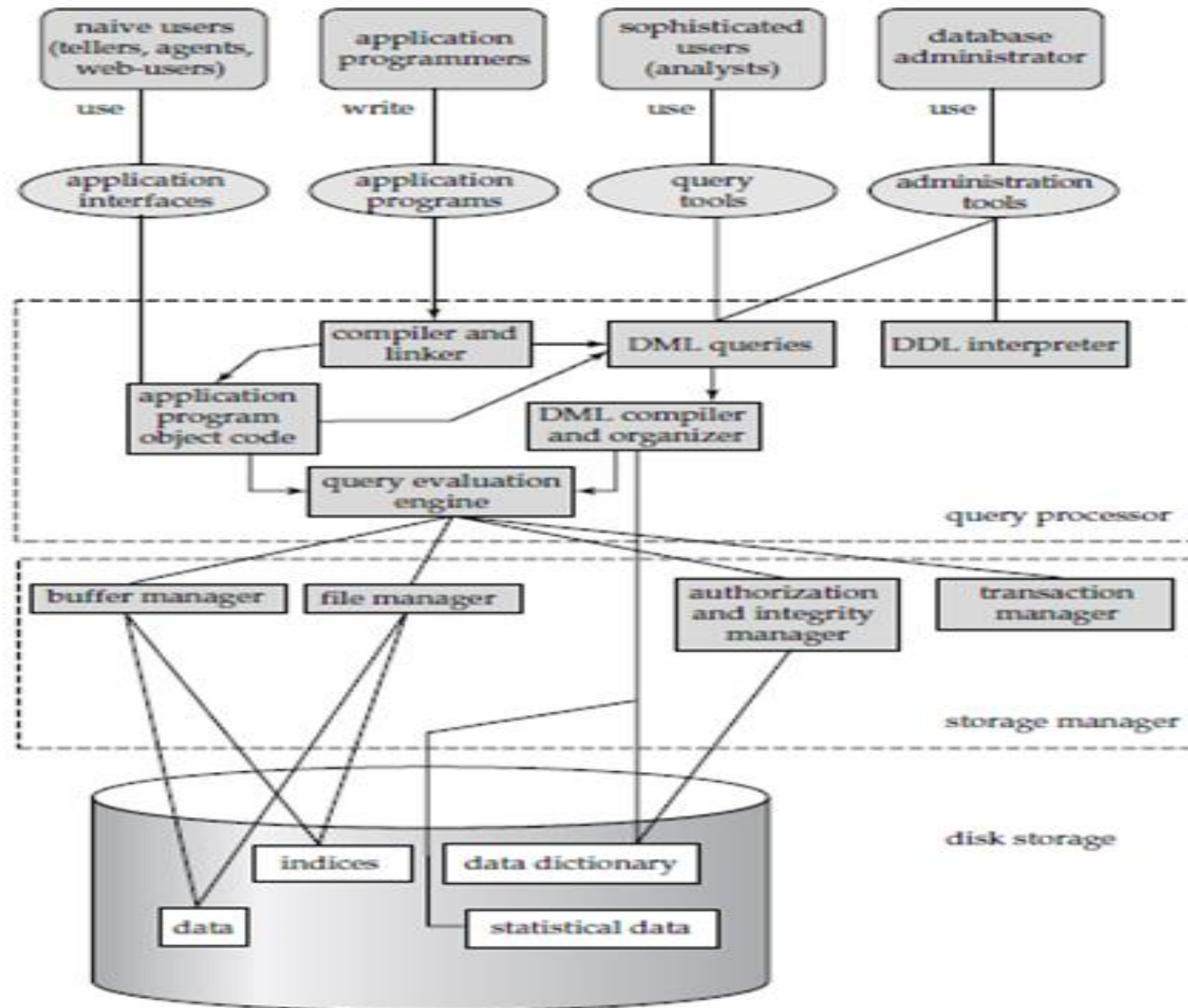
- Logical Data Independence is the ability to change the conceptual scheme without changing
  - External views
  - External API or programs
- Any change made will be absorbed by the mapping between external and conceptual levels.
- **When compared to Physical Data independence, it is challenging to achieve logical data independence.**

### Examples of changes under Logical Data Independence

- Due to Logical independence, any of the below change will not affect the external layer.
- Add/Modify/Delete a new attribute, entity or relationship is possible without a rewrite of existing application programs
- Merging two records into one
- Breaking an existing record into two or more records



# Database System Architecture



Users of  
Database

Query  
Processor

Storage  
Manager

Disk Storage



# Users of Database Management System

1. **Naive users** : They are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.
2. **Application programmers**: They are computer professionals who write application programs. Application programmers can choose from many tools to develop user interfaces.
3. **Sophisticated users**: They interact with the system without writing programs. Instead, they form their requests in a database query language.
4. **Specialized users**: They are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework.
5. **Database Administrator**: One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. A person who has such central control over the system is called a database administrator (DBA).

# Functions of Database Administrator(DBA)

1. Schema Definition
2. Storage structure and access-method Definition
3. Schema and Physical organization modification
4. Granting of authorization for data access
5. Routine Maintenance

- **The Query Processor** :The query processor components include

**1.DDL interpreter**, which interprets DDL statements and records the definitions in the data dictionary.

**2.DML compiler**, which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.A query can usually be translated into any of a number of alternative evaluation plans that all give the same result. The DML compiler also performs **query optimization**, that is, it picks the lowest cost evaluation plan from among the alternatives.

**3.Query evaluation engine**, which executes low-level instructions generated by the DML compiler.

- **Storage Manager**

A storage manager is a program module that provides the interface between the lowlevel data stored in the database and the application programs and queries submitted to the system. The storage manager is responsible for the interaction with the file manager. The raw data are stored on the disk using the file system, which is usually provided by a conventional operating system. The storage manager translates the various DML statements into low-level file-system commands. Thus , the storage manager is responsible for storing, retrieving, and updating data in the database.

- The storage manager components include:
  - **Authorization and integrity manager**, which tests for the satisfaction of integrity constraints and checks the authority of users to access data.
  - **Transaction manager**, which ensures that the database remains in a consistent(correct) state despite system failures, and that concurrent transaction executions proceed without conflicting.
  - **File manager**, which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.
  - **Buffer manager**, which is responsible for fetching data from disk storage into main memory, and deciding what data to cache in main memory. The buffer manager is a critical part of the database system, since it enables the database to handle data sizes that are much larger than the size of main memory.
- The storage manager implements several data structures as part of the physical system implementation:
  - Data files, which store the database itself.
  - Data dictionary, which stores metadata about the structure of the database, in particular the schema of the database.
  - Indices, which provide fast access to data items that hold particular values.

- **Metadata** is data that describes other data. ... For **example**, author, date created, date modified and file size are **examples** of very basic document **metadata**.
- A **Data Dictionary** is a file or a set of files that contains a database's metadata. The **data dictionary** contains records about other objects in the database, such as **data** ownership, **data** relationships to other objects, and other **data**. The **data dictionary** is a crucial component of any relational database
- Data dictionary components
  - Data object listings (names and definitions)
  - Data element properties (such as data type, unique identifiers, size, nullability, indexes and optionality)
  - Entity-relationship diagrams (ERD)
  - System-level diagrams
  - Reference data
  - Missing data and quality-indicator codes
  - Business rules (such as for validation of data quality and schema objects)