DBMS – Database Users

Database Users

Actors on the Scene

- ★ Database Administrators
- ★ Database Designers
- ★ End Users
- ★ System Analysts & Application Programmers (Software Engineers)

Workers Behind the Scene

- ★ System designers & implementers
- ★ Tool developers
- ★ Operators & maintenance personnel

Front Actors (1/5)

★ Database Administrators:

- ➜ In database environment, primary resource → database, secondary resource → DBMS & related software.
- ➜ Database Administrator (DBA) responsibilities:
  1) Administering primary/secondary resources.
  2) Authorizing access to the database.
  3) Co-ordinating & monitoring use of database.
  4) Acquiring hardware & software resources as needed.

Front Actors (2/5)

★ Database Designers:

→ Responsible for:

1) Identifying the data to be stored in the database.
2) Choosing appropriate structures to represent and store data.
3) Communicating with database users → understand their requirements → designs database.

Front Actors (3/5)

★ End Users:

- → End users → people whose jobs require access to the database → for querying, updating, generating reports.
- → Several categories of end users:
  - Casual end users: Accesses database occasionally → typically middle or high-level managers or other occasional browsers.

Front Actors (4/5)

★ End Users:

→ Several categories of end users:

- Naive or parametric end users: constantly querying and updating database using canned transactions.
- Sophisticated end users: Engineers, scientists, business analysts.
- Stand-alone users: Maintains personal databases → using ready-made program packages.

Front Actors (5/5)

★ System Analysts & Application Programmers (Software Engineers):

- ➔ System Analysts → determine the requirements of end users → develop specifications for canned transactions.
- ➔ Application Programmers → test, debug, document and maintain these canned transactions.

Workers

- ★ **System designers and implementers:** Design and implement DBMS modules & interfaces as a package.
- ★ **Tool Developers:** Persons who design and implement tools ↓ Software packages
- ★ **Operators and maintenance personnel:** Responsible for actual running and maintenance of hardware & software.

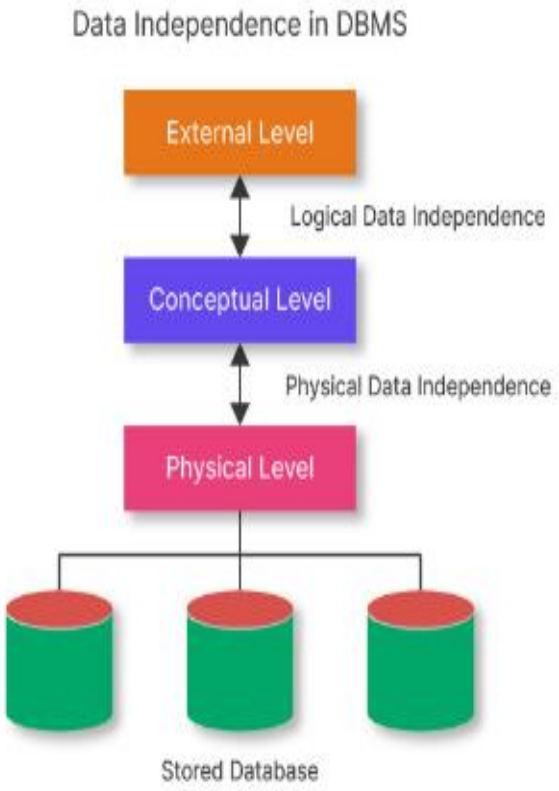
Data Independance

Data independence

- Definition:
- The ability of the data to change at one level of the database with change the schema at the next hi

<u>In simple words:</u>

It allows the User or Database Adm change the schema at one level wit affecting the data or schema at anc



- There are two levels of data independence in DBMS:
  - **Physical level data independence**
  - **Logical level data independence**

1. Physical Level Data Independence

- Physical Data Independence can be defined as the ability to change the physical level without affecting the logical or Conceptual level.
- Physical data independence gives us the freedom to modify the - Storage device, File structure, location of the database, etc. without changing the definition of conceptual or view level.
- Example: For example, if we take the database of the banking system and we want to scale up the database by changing the storage size and also want to change the file structure, we can do it without affecting any functionality of logical schema.
- Below changes can be done at the physical layer without affecting the conceptual layer -
  - Changing the storage devices like SSD, hard disk and magnetic tapes, etc.
  - Changing the access technique and modifying indexes.
  - Changing the compression techniques or hashing algorithms.

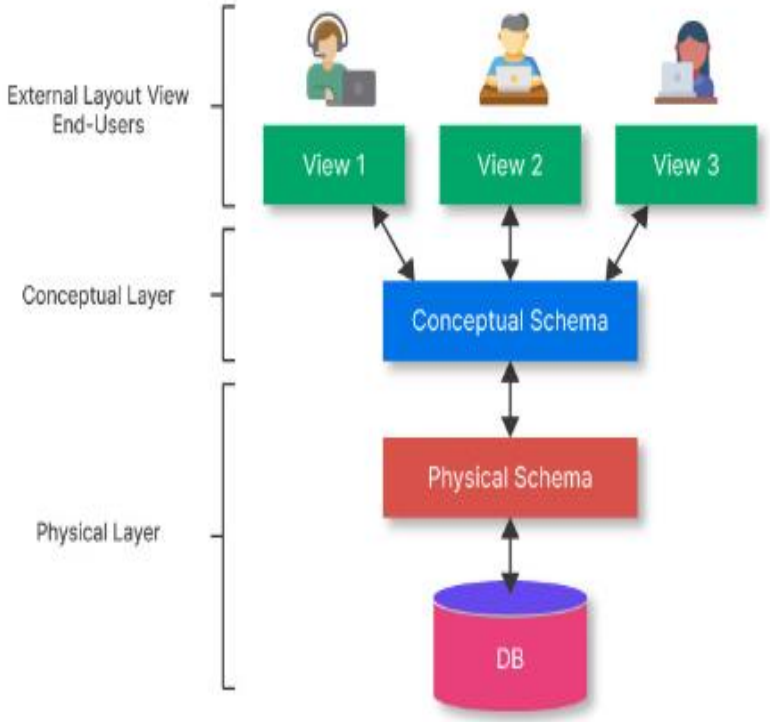
2. Logical Level Data Independence

- Logical Data Independence is a property of a database that can be used to change the logic behind the logical level without affecting the other layers of the database.
- Logical data independence is usually required for changing the conceptual schema without having to change the external schema or application programs.
- It allows us to make changes in a conceptual structure like adding, modifying, or deleting an attribute in the database.

- Example: If there is a database of a banking system and we want to add the details of a new customer or we want to update or delete the data of a customer at the logical level data will be changed but it will not affect the Physical level or structure of the database.
- These changes can be done at a logical level without affecting the application program or external layer.
  - Adding, deleting, or modifying the entity or relationship.
  - Merging or breaking the record present in the database.

Data abstraction

- **Achieving Data Independence in DBMS Through Data Abstraction**
- Data Abstraction can be defined as extracting the necessary data by ignoring the remaining irrelevant details.



- Example: ATM

There are three levels of abstraction

- **Physical or Internal Level -**
- Physical level is the lowest level of data abstraction, and It indicates how the data will be stored and describes the complex data structures and access methods to be used by the database.
- The internal level is used to describe the entire database architecture.
- **Conceptual or Logical Level -**
- The separation of the conceptual view from the internal view enables us to provide a logical description of the database concepts without the need to specify physical structures.
- The conceptual level comes between the physical level and the view level.
- It provides the link between the external schema and the internal schema of the database.
- **External or View Level -** It is the highest level of data abstraction.
- The external level describes the user interaction with the centralized database management system.
- This level is used to provide a Graphical User Interface to the user, and the user does not know about the file structure, access method, and other internal details of the database.