

DATABASE MANAGEMENT SYSTEM (DBMS)

CONTENT

- View of Data
- Three Schema Architecture

View of Data

- A database system is a collection of interrelated files and a set of programs that allow users to access and modify these files.
- A major purpose of a database system is to provide users with an *abstract view of the data*.
- Data Abstraction
 - For the system to be usable, it must retrieve data efficiently. The need for efficiency has led designers to use complex data structures to represent data in the database.
 - Thus *abstraction refers to hiding the complexity from users* through several levels of abstraction, to simplify users' interactions with the system.

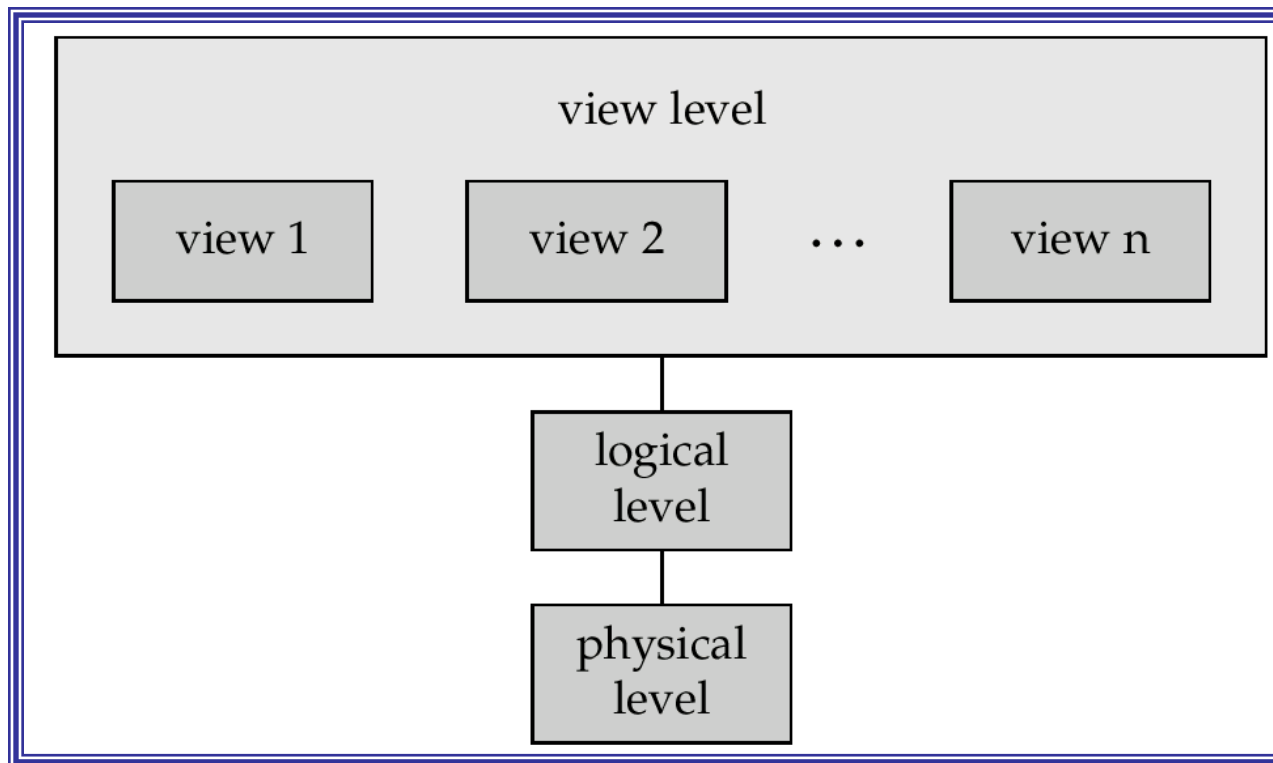
Data Abstraction

Data retrieval from database should be made easy & efficient since database user are not computer trained .

So the developer hide the complexity from user for several level of abstraction.

View of Data

An architecture for a database system



View of Data

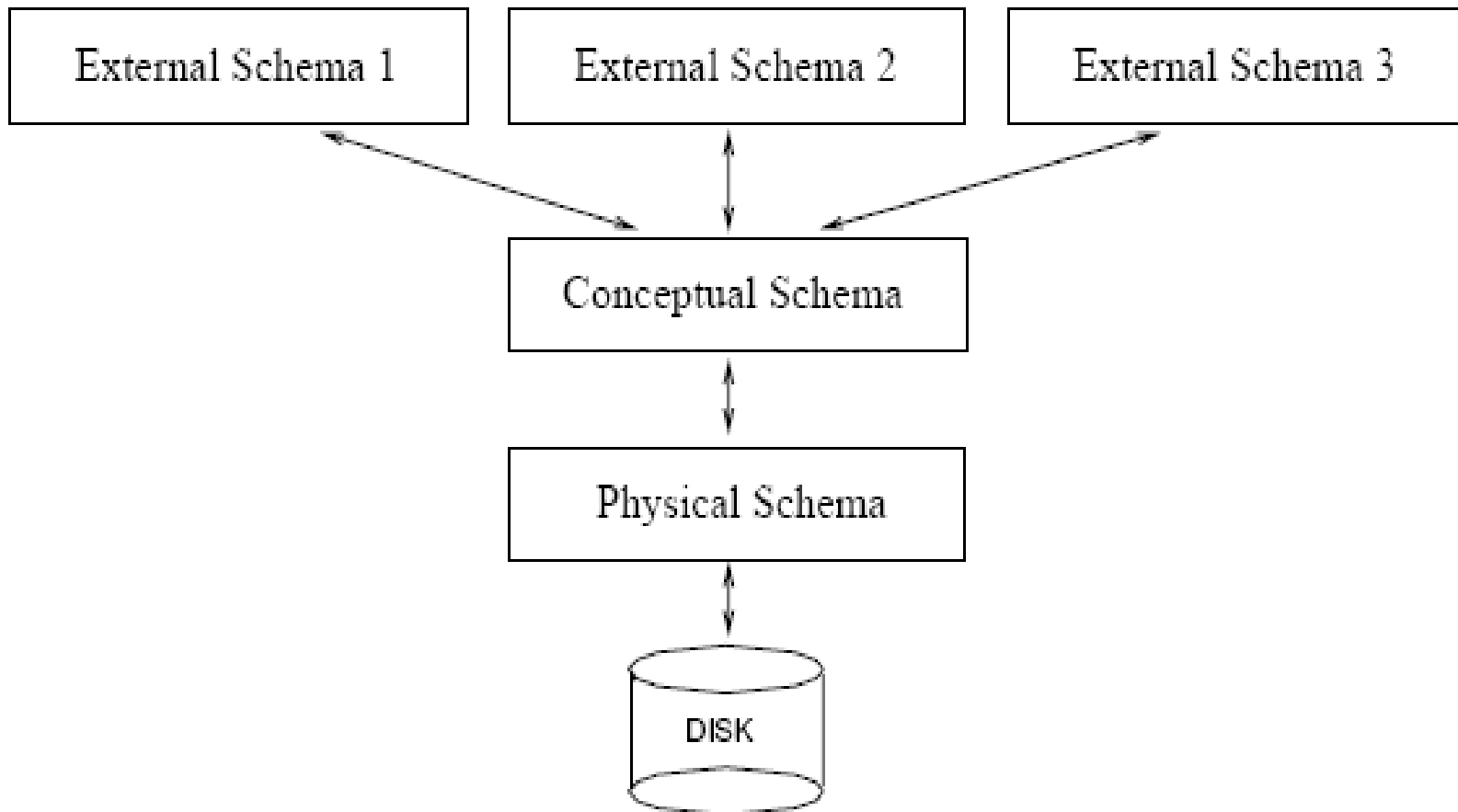


Figure 1.2 Levels of Abstraction in a DBMS

View of Data

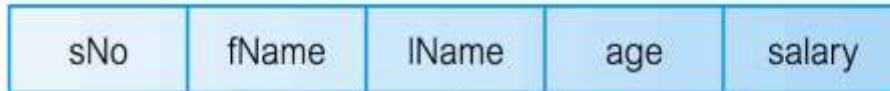
- **Physical level.** (Physical schema describes the files and indexes used.)
 - The lowest level of abstraction describes *how* the data are actually stored.
 - The physical level describes complex low-level data structures in detail. The design of data structure is described at this level called physical schema.
 - It specifies that records are stored in either as pages.
- **Logical level.** (Conceptual schema defines logical structure)
 - This is middle level of abstraction and it describes **what data are stored in the database, and what relationships exist among those data, there is only one schema only for one database.**
 - The logical level thus describes the entire database in terms of a small number of relatively simple structures.
 - **The logical level of abstraction is used by database administrator, who decides what information has to be kept inside the database.**

View of Data

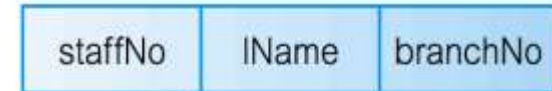
- **View level.**(External schemata describe how users see the data.)
 - The highest level of abstraction describes only part of the entire database. Even though the logical level uses simpler structures, complexity remains because of the variety of information stored in a large database.
 - Many users 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.

Differences between Three Levels of ANSI-SPARC Architecture

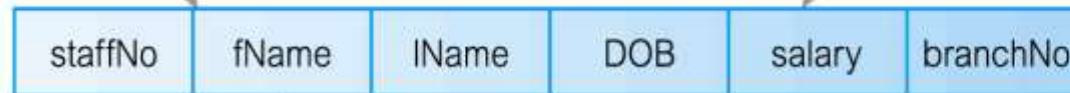
External view 1



External view 2



Conceptual level



Internal level

```
struct STAFF {  
    int staffNo;  
    int branchNo;  
    char fName [15];  
    char lName [15];  
    struct date dateOfBirth;  
    float salary;  
    struct STAFF *next;           /* pointer to next Staff record */  
};  
index staffNo; index branchNo; /* define indexes for staff */
```

Levels of Abstraction(View of Data)

- Physical level: It describes how a record (e.g., customer) is stored.
- Logical level: describes data stored in database, and the relationships among the data.

```
type customer = record  
    name : string;  
    street : string;  
    city : integer;  
end;
```

- View level: application programs hide details of data types. Views can also hide information (e.g., salary) for security purposes.

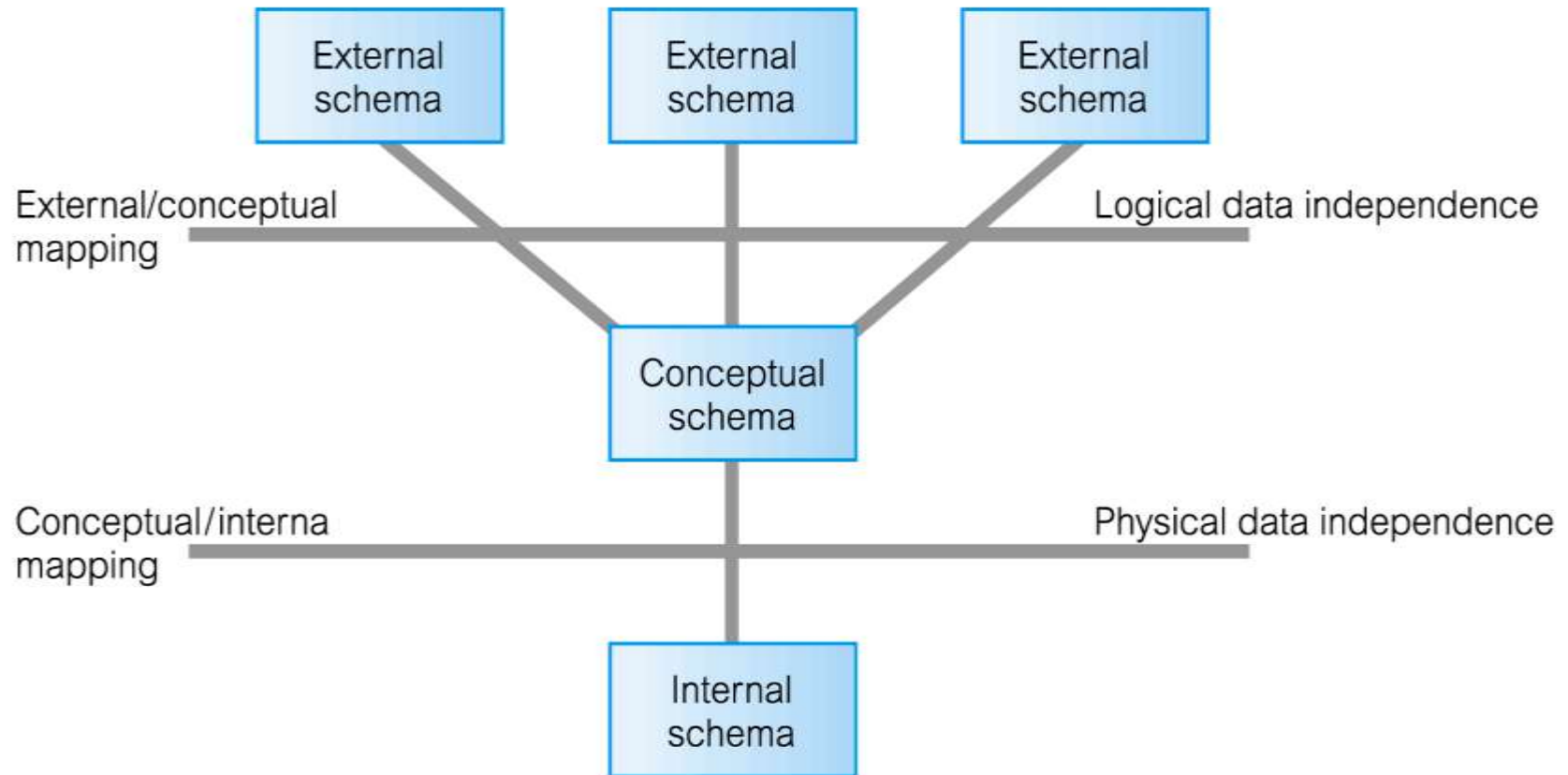
Three-Schema Architecture

- This idea was first described by the ANSI/SPARC committee in late 1970's. The goal is to separate (i.e., insert layers of "insulation" between) user applications and the physical database.
- C.J. Date points out that it is an ideal that few, if any, real-life DBMS's achieve fully.
- Proposed to support DBMS characteristics of:
 - **Program-data independence.**
 - Support of **multiple views** of the data.

Three-Schema Architecture

- Defines DBMS schemas at **three** levels:
 - **Internal schema** at the internal level to describe physical storage structures and access paths (e.g indexes).
 - Typically uses a **physical** data model.
 - **Conceptual schema** at the conceptual level to describe the structure and constraints for the whole database for a community of users.
 - Uses a **conceptual** or an **implementation** data model.
 - **External schemas** at the external level to describe the various user views.
 - Usually uses the same data model as the conceptual schema.

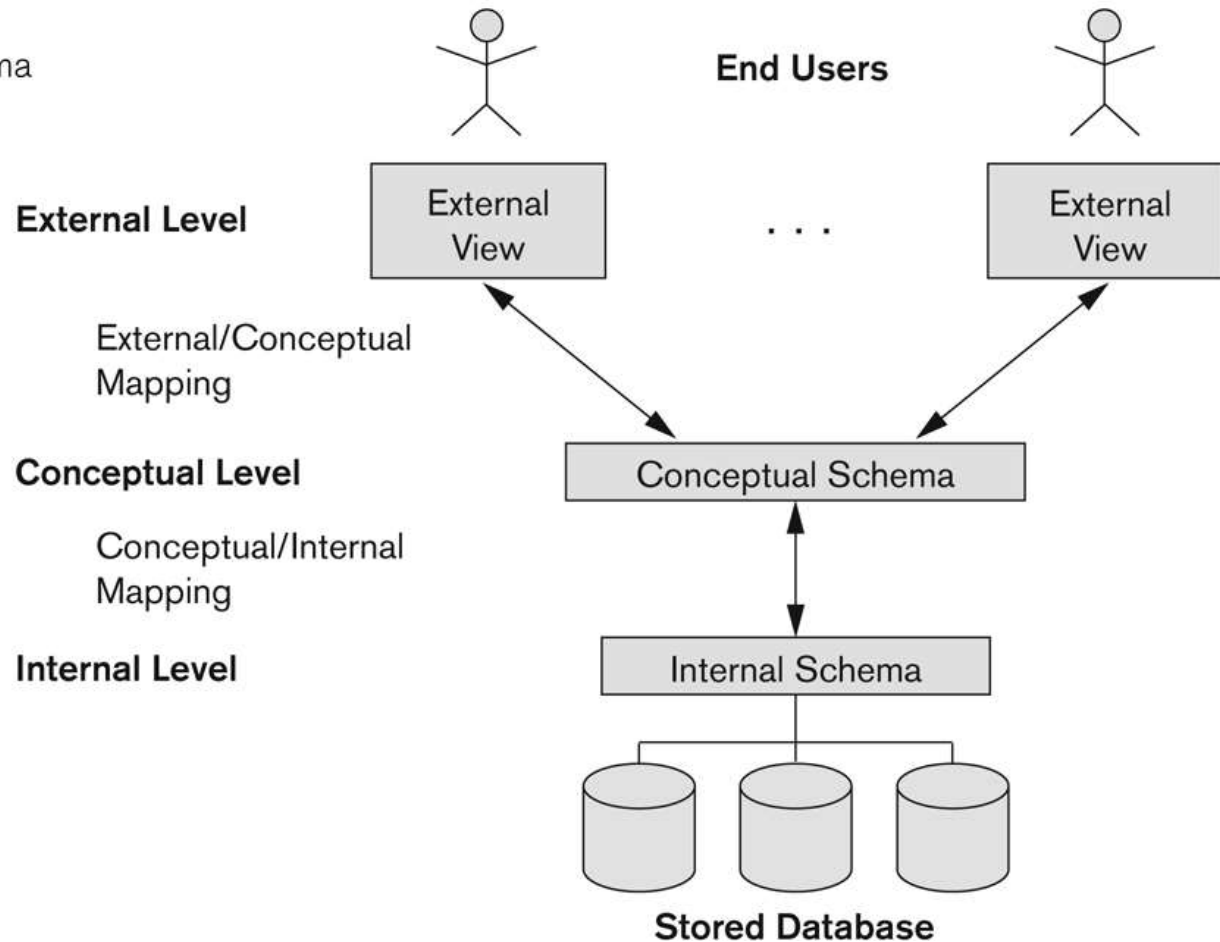
Data Independence and the ANSI-SPARC Three-Schema Architecture



Three-Schema Architecture

Figure 2.2

The three-schema architecture.



3 Level Architecture →

DBMS Architecture Divided in 3 levels →

- ① External
- ② Conceptual
- ③ Internal.

① External (or) View Level → It is concerned with individual users. The external schema is defined by the DBA for every user. The remaining part of Database is hidden from the user. DBA authorizes user for accessing a part of Database. This level is also Relational (or) very close to it. System provide multiple views for the same Database.

② Conceptual → It describes overall logical structure of whole Database. This level is Relational because data visible at this level will be Relational table. This level represents entire contents of ~~data~~ Database in an abstract form. It is a global view of Database. It describes what data stored & Relationship among them.

The Conceptual level Represents →

- ① All entities, Attribute & Relationship.
- ② Constraints on the Data.
- ③ Security & Integrity Information.

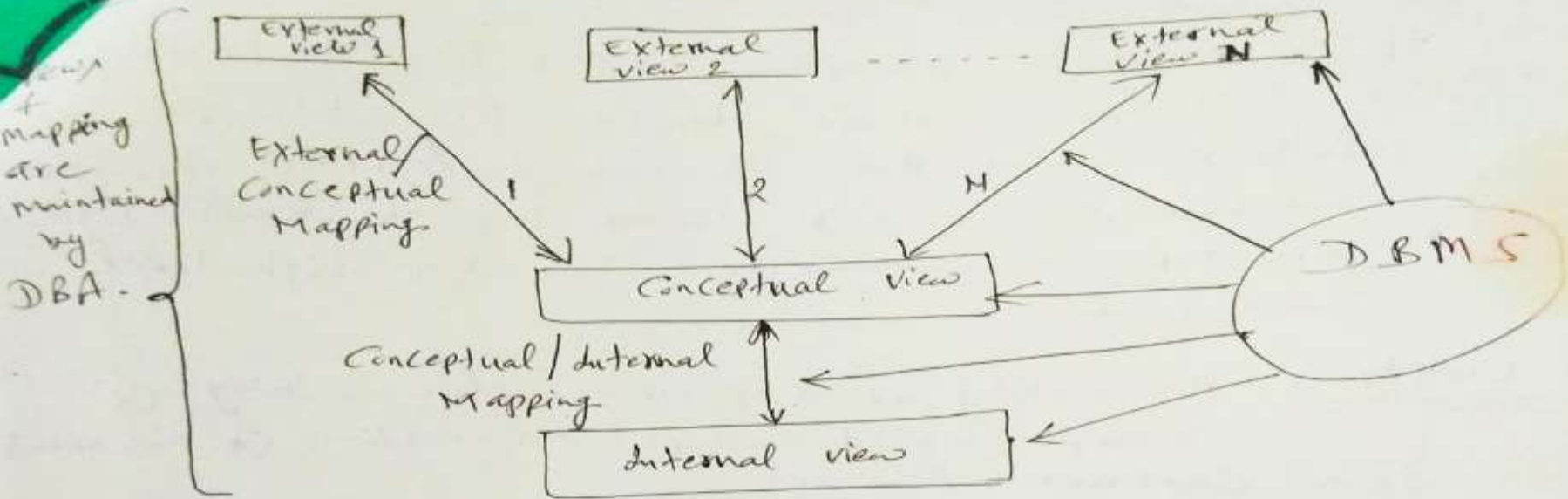
③ Internal level ⇒ It describes actual physical storage. This level is not Relational because data is stored according to various Coding scheme instead of tables. It gives complete details of data storage access paths. This level covers the physical implementation of the Database.

This level Represents →

- ① Storage space Allocation for Data.
- ② Data Compression & Data Encryption Techniques.
- ③ Record Description for Storage Data.

Mapping ⇒ ① Conceptual / Internal mapping ⇒

It defines operations between conceptual & physical view. It specifies how the Data is retrieved from conceptual level to physical level @ vice versa. It specifies how conceptual Records & fields are represented at the internal level. It also resolves problem of any Differences in entity Name, Attributes Name etc.



* 3 Level Architecture of DBMS *

- (2) External / Conceptual Mapping \Rightarrow It defines the correspondence between Conceptual & External view. It specifies how the data is retrieved from conceptual level & shown at external level because at external level some part of database is hidden from a user.
- \rightarrow Physical Data Independence is achieved through Conceptual / Internal mapping.
 - \rightarrow Logical Data Independence is achieved through External / Conceptual Mapping.