Introduction to DBMS

By :
Dr. Rinkle Rani
Associate Professor, CSED
TIET, Patiala

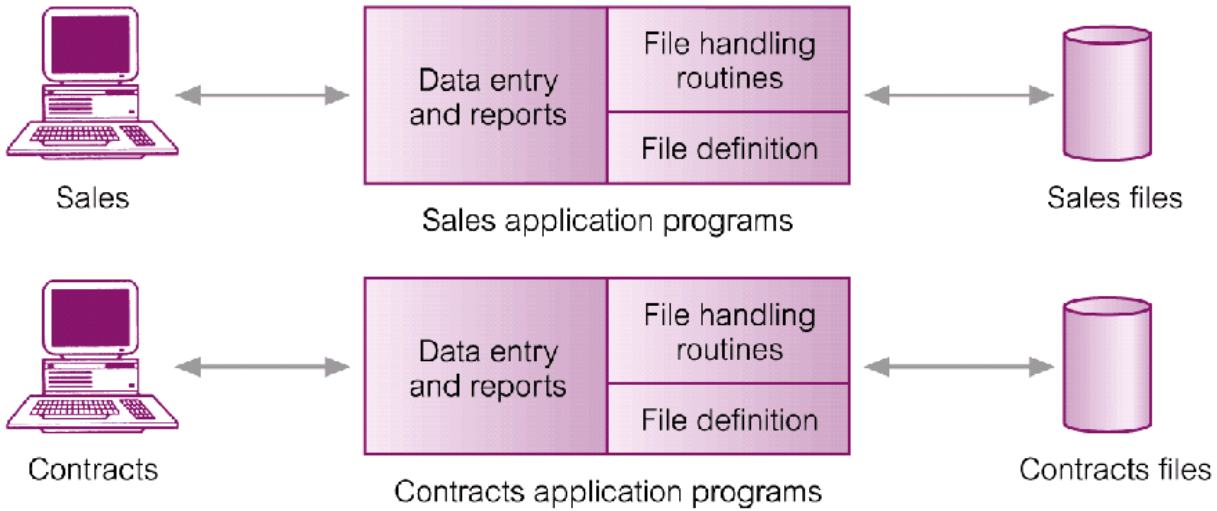
Outline

- Introduction
  - File based approach
  - Database approach
  - Basic definitions
- Database systems concepts
  - Data model
  - Three schema architecture – Data independence
  - Database schema – state – instance
  - DBMS languages
  - Classification of DBMS
  - Database users

File-based Approach

- Data is stored in one or more separate computer files
- Data is then processed by computer programs - **applications**

File-based Approach



Sales Files

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)

PrivateOwner (ownerNo, fName, lName, address, telNo)

Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Contracts Files

Lease (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)

PropertyForRent (propertyNo, street, city, postcode, rent)

Client (clientNo, fName, lName, address, telNo)

Drawbacks of File system

Data redundancy: Data redundancy refers to the duplication of data, lets say we are managing the data of a college where a student is enrolled for two courses, the same student details in such case will be stored twice, which will take more storage than needed. Data redundancy often leads to higher storage costs and poor access time.

Data inconsistency: Data redundancy leads to data inconsistency, lets take the same example that we have taken above, a student is enrolled for two courses and we have student address stored twice, now lets say student requests to change his address, if the address is changed at one place and not on all the records then this can lead to data inconsistency.

Data Isolation: Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult.

Dependency on application programs: Changing files would lead to change in application programs.

Data Security: Data should be secured from unauthorised access, for example a student in a college should not be able to see the payroll details of the teachers, such kind of security constraints are difficult to apply in file processing systems.

Database Approach

Database systems offer solutions to all the above problems

What is a Database?

Å A collection of related pieces of data:

Å A database Represents/captures the information about a real-world enterprise or part of an enterprise.

For Example

University Database:
Data about students, faculty, courses, research-laboratories, course registration/enrollment etc.

Reflects the state of affairs of the academic aspects of the university.

Purpose: To keep an accurate track of the academic activities of the university.

Å **Database Management System (DBMS)**:

DBMS stands for **D**ata**b**ase **M**anagement **S**ystem. We can break it like this DBMS = Database + Management System. Database is a collection of data and Management System is a set of programs to store and retrieve those data. Based on this we can **define DBMS** like this: DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

DBMS : A software package/ system to facilitate the creation and maintenance of a computerized database.

Typical DBMS Functionality

- Define a database : in terms of data types, structures and constraints
- Construct or Load the Database on a secondary storage medium
- Manipulating the database : querying, generating reports, insertions, deletions and modifications to its content
- Concurrent Processing and Sharing by a set of users and programs – yet, keeping all data valid and consistent

Typical DBMS Functionality

- Other features:
  - Protection or Security measures to prevent unauthorized access
  - “Active” processing to take internal actions on data
  - Presentation and Visualization of data

Advantage of DBMS over file system

- **No redundant data**: Redundancy removed by data normalization. No data duplication saves storage and improves access time.
- **Data Consistency and Integrity**: As we discussed earlier the root cause of data inconsistency is data redundancy, since data normalization takes care of the data redundancy, data inconsistency also been taken care of as part of it
- **Data Security**: It is easier to apply access constraints in database systems so that only authorized user is able to access the data. Each user has a different set of access thus data is secured from the issues such as identity theft, data leaks and misuse of data.

- **Privacy**: Limited access means privacy of data.
- **Easy access to data** – Database systems manages data in such a way so that the data is easily accessible with fast response times.
- **Easy recovery**: Since database systems keeps the backup of data, it is easier to do a full recovery of data in case of a failure.
- **Flexible**: Database systems are more flexible than file processing systems.

Data Sharing

Data Sharing is the primary advantage of Database management systems. DBMS system allows users and applications to share Data with multiple applications and users.

Data Concurrency

In DBMS, Data are stored in one or more servers in the network and that there is some software locking mechanism that prevents the same set of data from being changed by two people at the same time.

DBMS Architecture

Database management systems architecture will help us understand the components of database system and the relation among them.The architecture of DBMS depends on the computer system on which it runs.

Types of DBMS Architecture

There are three types of DBMS architecture:

1. Single tier architecture
2. Two tier architecture
3. Three tier architecture

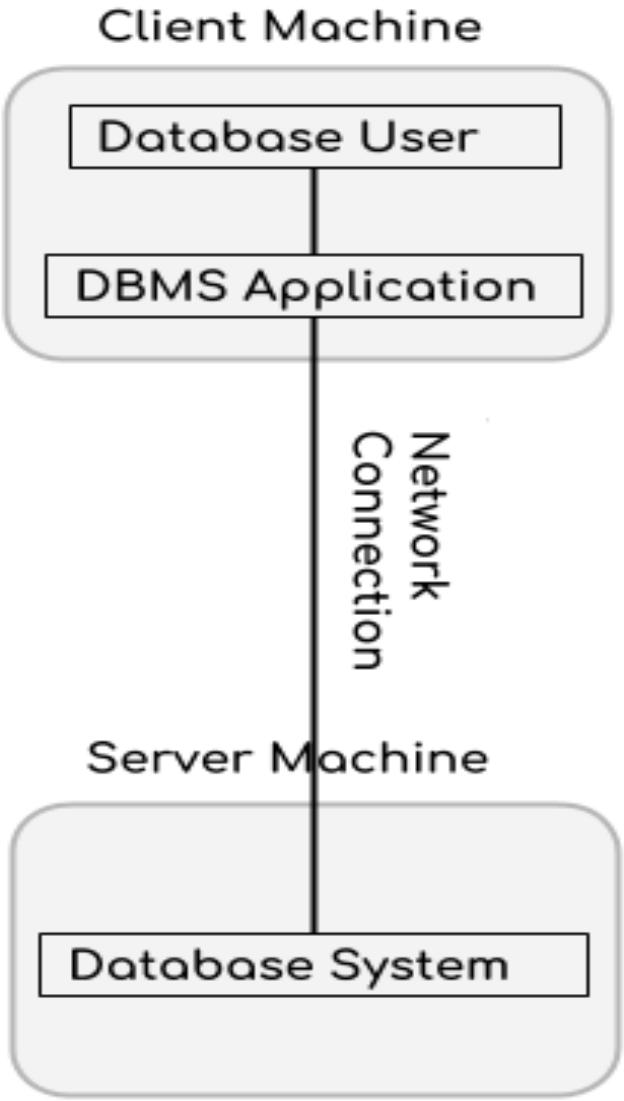
1. Single tier architecture

In this type of architecture, the database is readily available on the client machine, any request made by client doesn't require a network connection to perform the action on the database.

For example, lets say you want to fetch the records of employee from the database and the database is available on your computer system, so the request to fetch employee details will be done by your computer and the records will be fetched from the database by your computer as well. This type of system is generally referred as local database system.

2. Two tier architecture

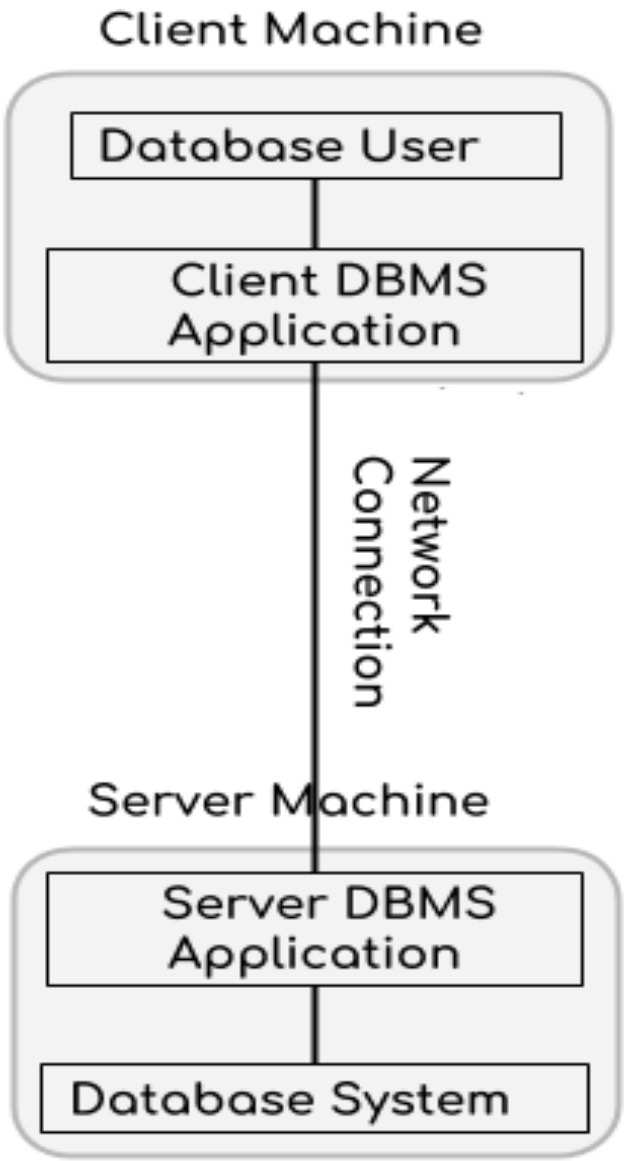
In two-tier architecture, the Database system is present at the server machine and the DBMS application is present at the client machine, these two machines are connected with each other through a reliable network as shown in the above diagram.



Whenever client machine makes a request to access the database present at server using a query language like sql, the server perform the request on the database and returns the result back to the client. The application connection interface such as JDBC, ODBC are used for the interaction between server and client.

3. Three tier architecture

In three-tier architecture, another layer is present between the client machine and server machine. In this architecture, the client application doesnôt communicate directly with the database systems present at the server machine, rather the client application communicates with server application and the server application internally communicates with the database system present at the server.



3-tier architecture is consist of the Presentation layer (PC, Tablet, Mobile, etc.), the Application layer (server) and Database Server

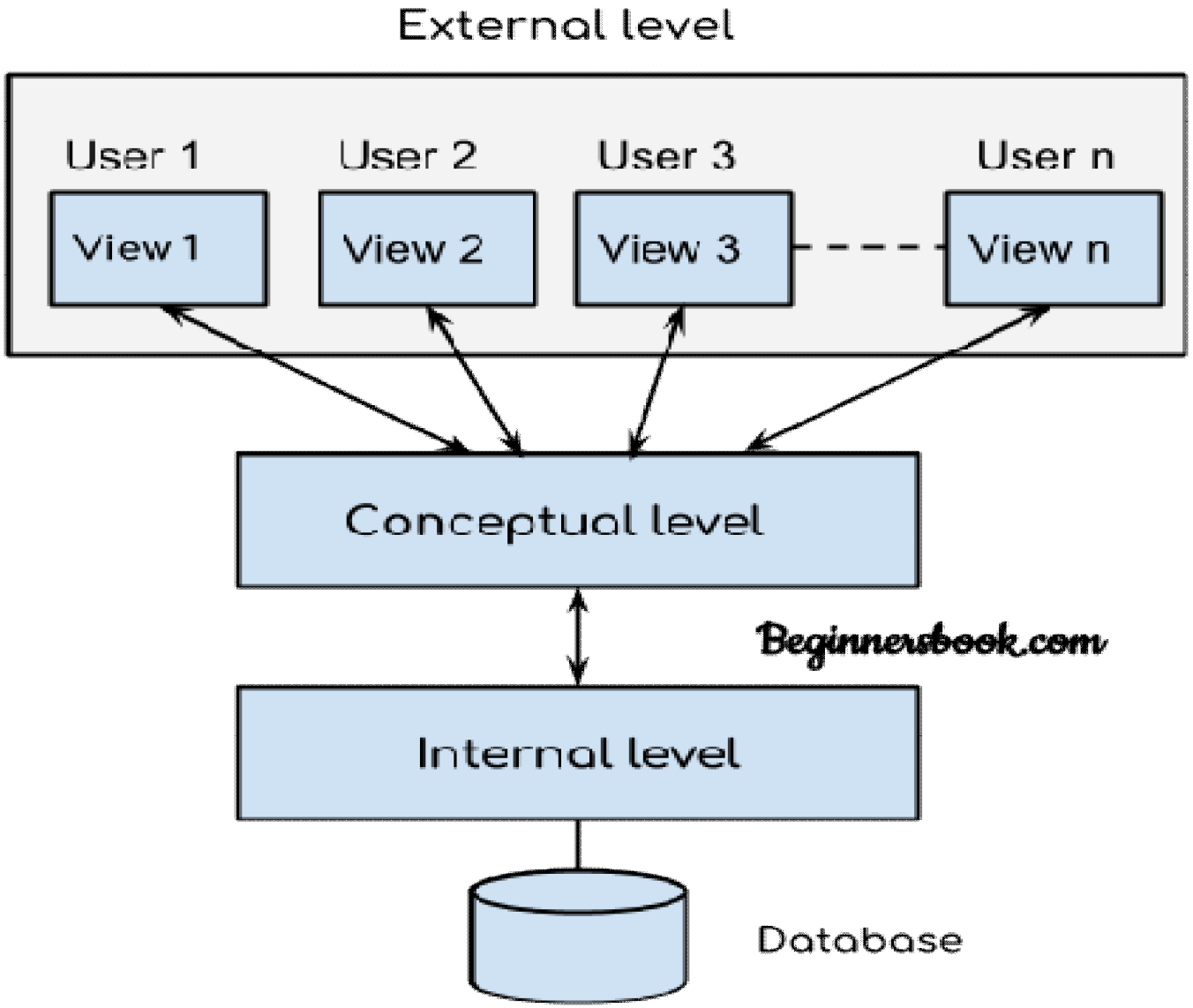
DBMS Three Level data Abstraction

Database systems comprise of complex data structures. Thus, to make the system efficient for retrieval of data and reduce the complexity of the users, developers use the method of Data Abstraction.

There are mainly three levels of data abstraction:

1. Internal Level: Actual PHYSICAL storage structure and access paths.
2. Conceptual or Logical Level: Structure and constraints for the entire database
3. External or View level: Describes various user views

DBMS Three Level data Abstraction Diagram



Instance and schema in DBMS

Instances :
Instance is the collection of information stored in the database at a particular moment. The instances can be changed by certain CRUD operations as like addition, deletion of data.

Schema :
Schema is the overall description of the database. The basic structure of how the data will be stored in the database is called schema.

Difference between Schema and Instance :

Schema	Instance
It is the overall description of the database.	It is the collection of information stored in a database at a particular moment.
Schema is same for whole database.	Data in instances can be changed using addition, deletion, updation.
Does not change Frequently.	Changes Frequently.
Defines the basic structure of the database i.e how the data will be stored in the database.	It is the set of Information stored at a particular time.

Internal Level/Schema

- The internal schema defines the physical storage structure of the database.
- The internal schema is a very low-level representation of the entire database.
- The internal view tells us what data is stored in the database and how
- It never deals with the physical devices. Instead, internal schema views a physical device as a collection of physical pages

Conceptual Schema/Level

- The conceptual schema describes the Database structure of the whole database for the community of users.
- This schema hides information about the physical storage structures and focuses on describing data types, entities, their attributes, and their relationships, etc.
- This logical level comes between the user level and physical storage view.
- However, there is only single conceptual view of a single database.

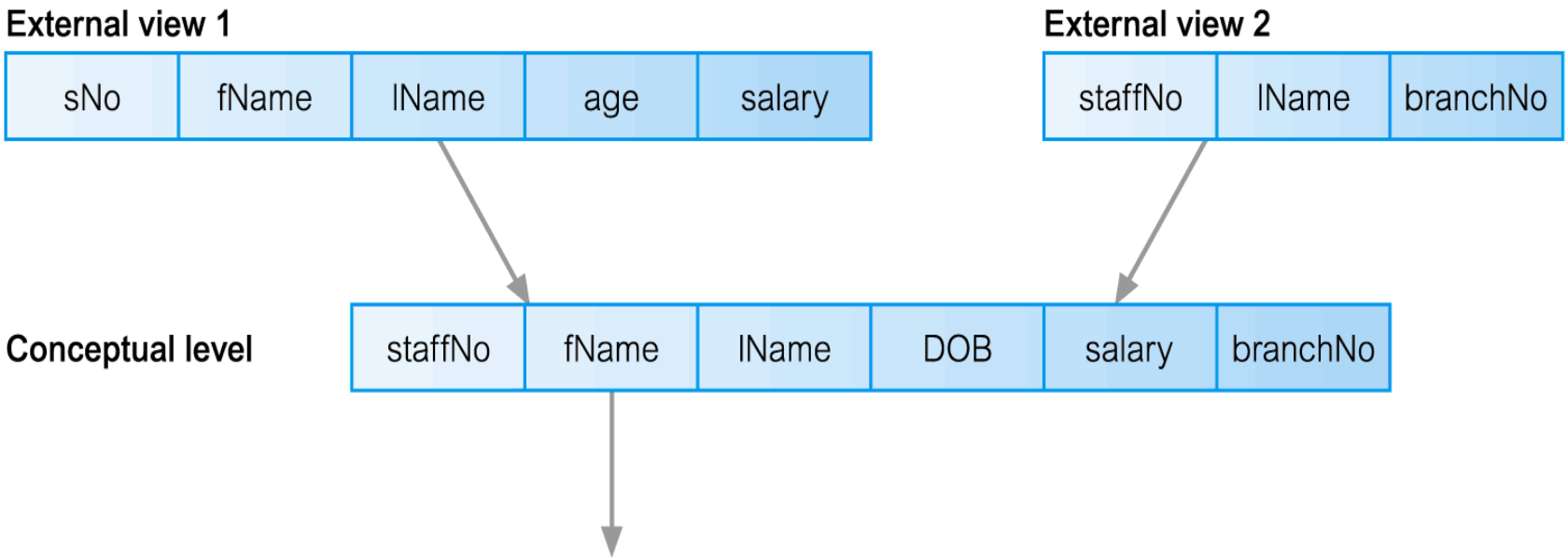
External Schema/Level

- An external schema describes the part of the database which specific user is interested in. It hides the unrelated details of the database from the user. There may be "n" number of external views for each database.
- Each external view is defined using an external schema, which consists of definitions of
- various types of external record of that specific view.
- An external view is just the content of the database as it is seen by some specific particular user. For example, a user from the sales department will see only sales related data.

Goals/Objectives of 3 level/schema of Database

- Here, are some Objectives of using Three schema Architecture:
- Every user should be able to access the same data but able to see a customized view of the data.
- The user need not to deal directly with physical database storage detail.
- The DBA should be able to change the database storage structure without disturbing the user's views
- The internal structure of the database should remain unaffected when changes made to the physical aspects of storage.

Three-Schema Architecture



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 */
```

Data Independence

Data Independence is the capacity to change the schema at one level of a database system without having to change the schema at the next higher level

Two types of data independence are present in this architecture:

1. Physical Data Independence:
2. Logical Data Independence:

Physical Data Independence:

Change *internal schema* without having to change *conceptual schema*.

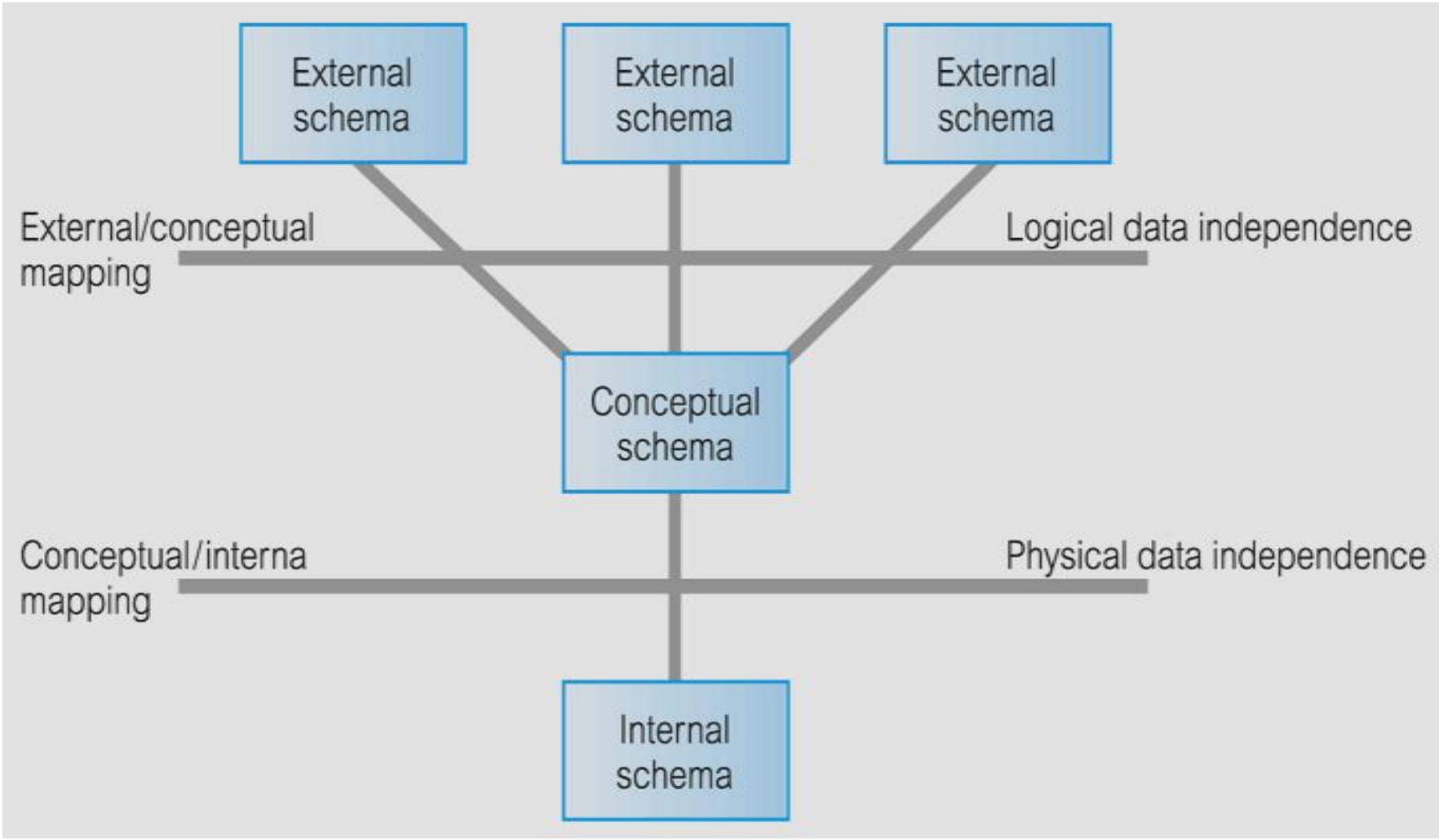
Any change in the physical location of tables and indexes should not affect the conceptual level or external view of data. This data independence is easy to achieve and implemented by most of the DBMS.

Conceptual/ Logical Data Independence:

Change *conceptual schema* without having to change *external schemas* and their application programs.

This means a change in conceptual schema should not affect external schema. e.g.; Adding or deleting attributes of a table should not affect the user's view of the table.

But this type of independence is difficult to achieve as compared to physical data independence because the changes in conceptual schema are reflected in the user's view.


External schema
External schema
External schema
External/conceptual mapping
Logical data independence
Conceptual schema
Conceptual/interna mapping
Physical data independence
Internal schema

DBMS Languages

- **Data Definition Language (DDL)** allows the DBA or user to describe and name entities, attributes, and relationships required for the application plus any associated integrity and security constraints
- **Data Manipulation Language (DML)** provides basic data manipulation operations on data held in the database
- **Data Control Language (DCL)** defines activities that are not in the categories of those for the DDL and DML, such as granting privileges to users, and defining when proposed changes to a databases should be irrevocably made