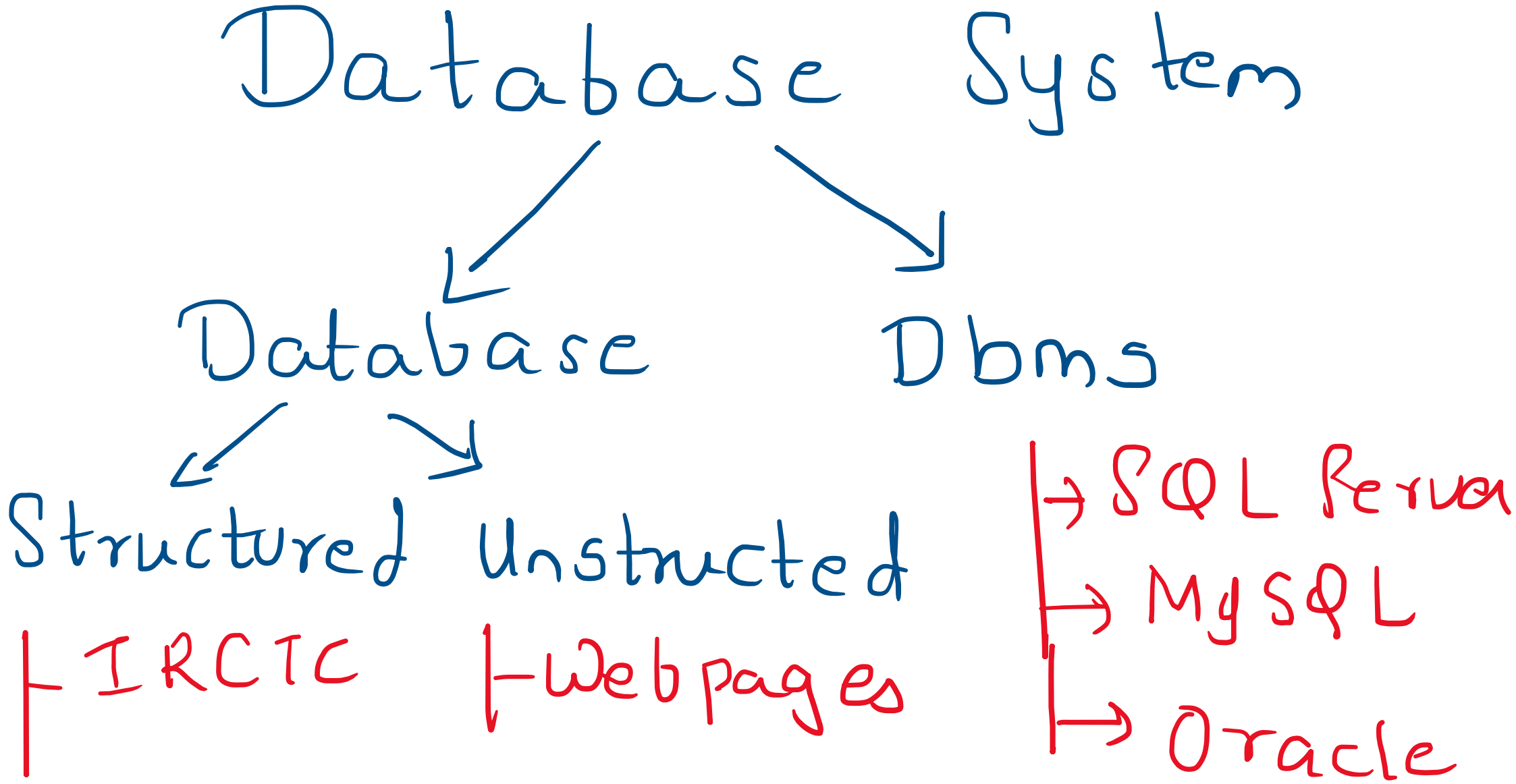
Chapter 1: Introduction

----Presented by: Prof. Charul Singh

Outline

- Database System
- Database System Examples
- File System v/s Database system
- Three-level schema architecture/ Data Abstraction
- Data Independence
- Data Models
- Users
- DBA
- DBMS Architecture


Database System
Database
Dbms
Structured
Unstructed
IRCTC
Webpages
SQL Server
MySQL
Oracle

Database Systems

- DBMS contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to access the data
  - An environment that is both *convenient* and *efficient* to use
- Database systems are used to manage collections of data that are:
  - Highly valuable
  - Relatively large
  - Accessed by multiple users and applications, often at the same time.
- A modern database system is a complex software system whose task is to manage a large, complex collection of data.
- Databases touch all aspects of our lives

Database Example

Applications are

University

- Add new students, instructors, and courses
- Register students for courses, and generate class rosters
- Assign grades to students, compute grade point averages (GPA) and generate transcripts

- Enterprise Information
  - Sales: customers, products, purchases
  - Accounting: payments, receipts, assets
  - Human Resources: Information about employees, salaries, payroll taxes.

File System v/s Database system

- Data redundancy and inconsistency
  - Multiple file formats, duplication of information in different files
- Difficulty in accessing data
  - Need to write a new program to carry out each new task
- Data isolation — multiple files and formats
- Integrity problems
  - Integrity constraints (e.g., account balance > 0) become “buried” in program code rather than being stated explicitly
  - Hard to add new constraints or change existing ones

- Atomicity of updates
  - Failures may leave database in an inconsistent state with partial updates carried out
  - Example: Transfer of funds from one account to another should either complete or not happen at all

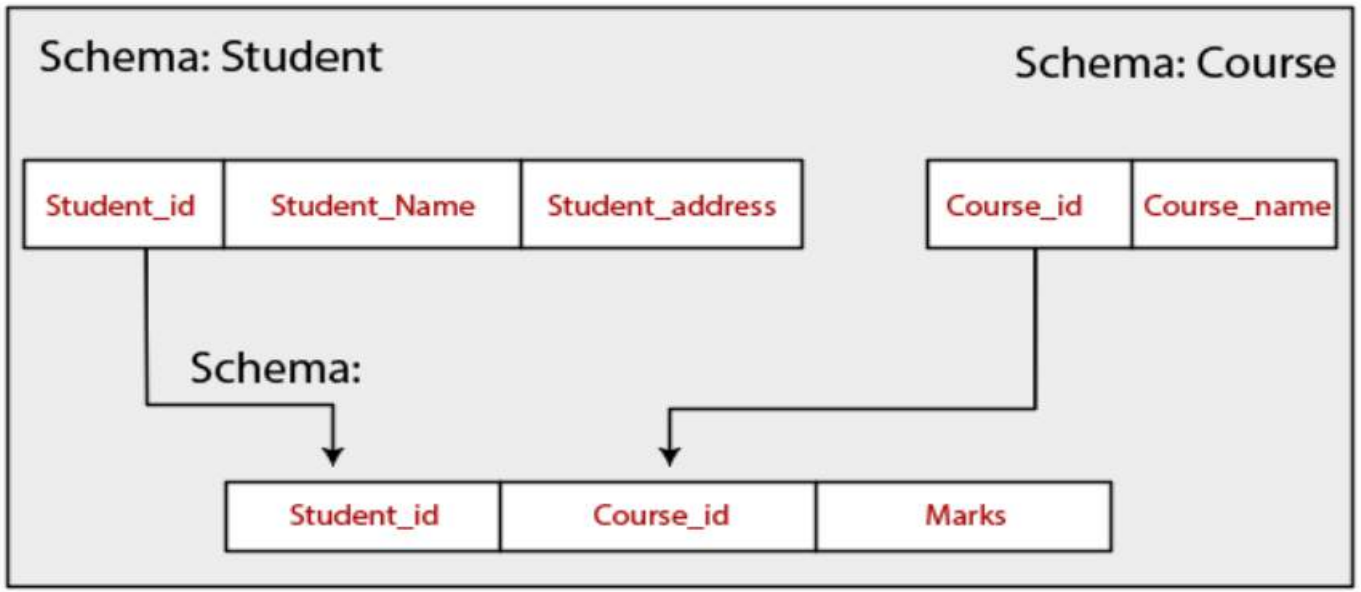
- Concurrent access by multiple users
  - Concurrent access needed for performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
    - Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time

- Security problems
  - Hard to provide user access to some, but not all, data

- **Database systems offer solutions to all the above problems**

Schema

- It is a logical representation of data that appears in the database management system.
- In simple words, a schema is the structure of any database.
- It defines how the data is stored in a database and relationship among those data, but it does not show the data available in those tables.
- it can only be modified or changed by modifying the DDL statements.



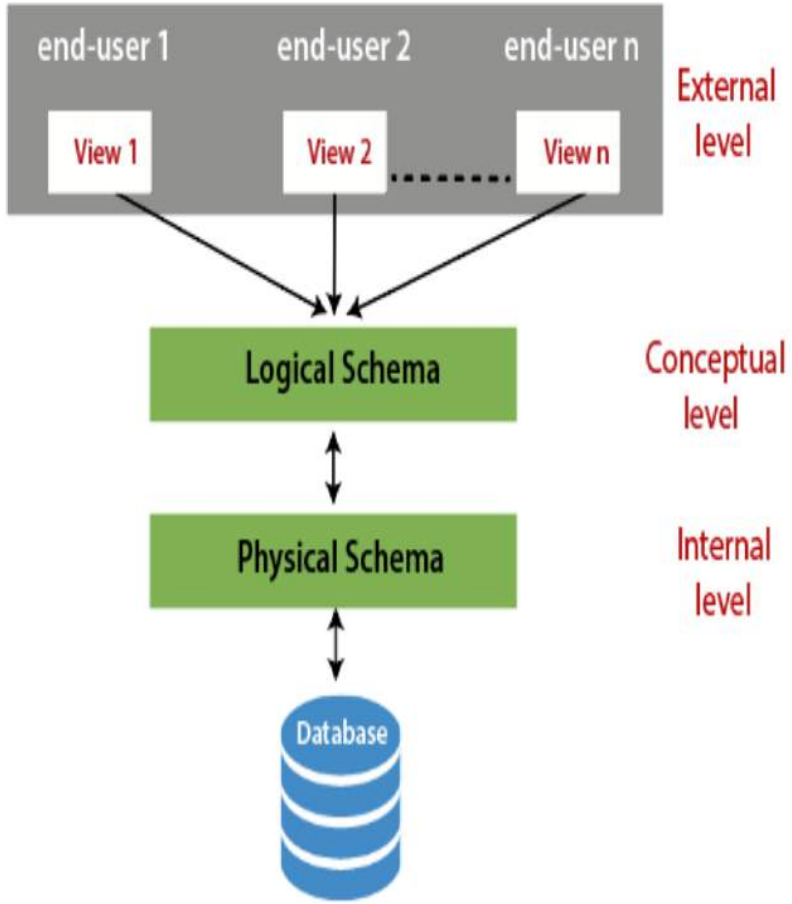
Three-Levels Schema Architecture/ Data Abstraction

Goal: To separate the user applications and the physical database.

3 Levels:

Internal Level:

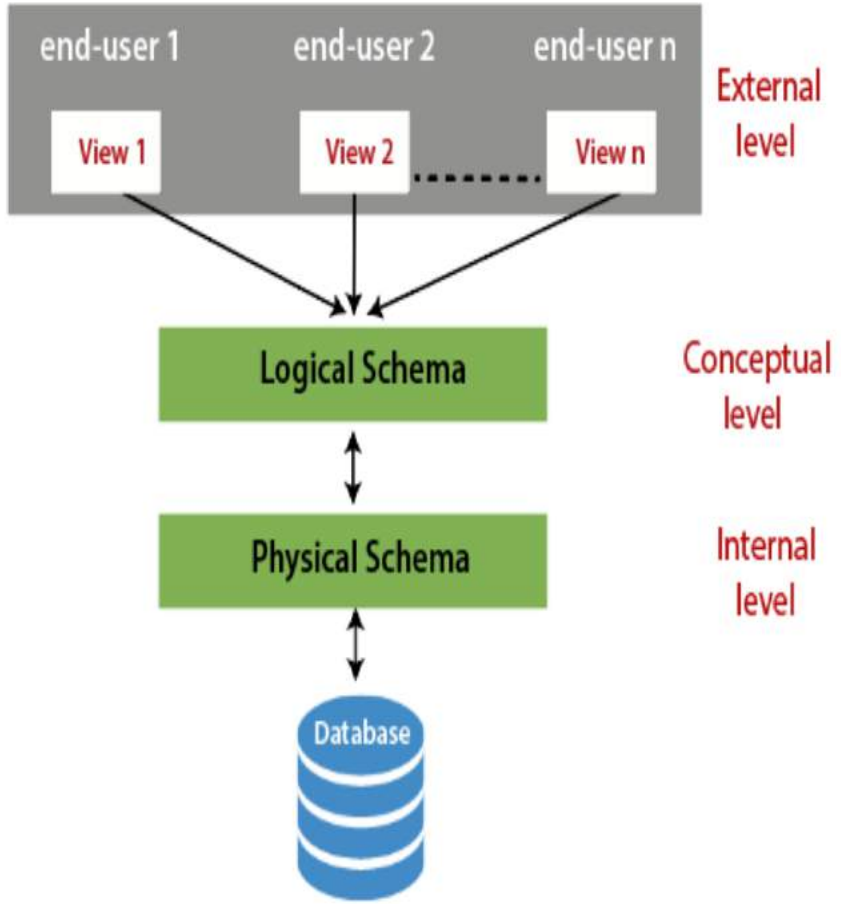
- Describes the physical storage structure Of the database.
- Describes complete details of data storage and access paths



Three-Levels Schema Architecture/ Data Abstraction

Conceptual Level:

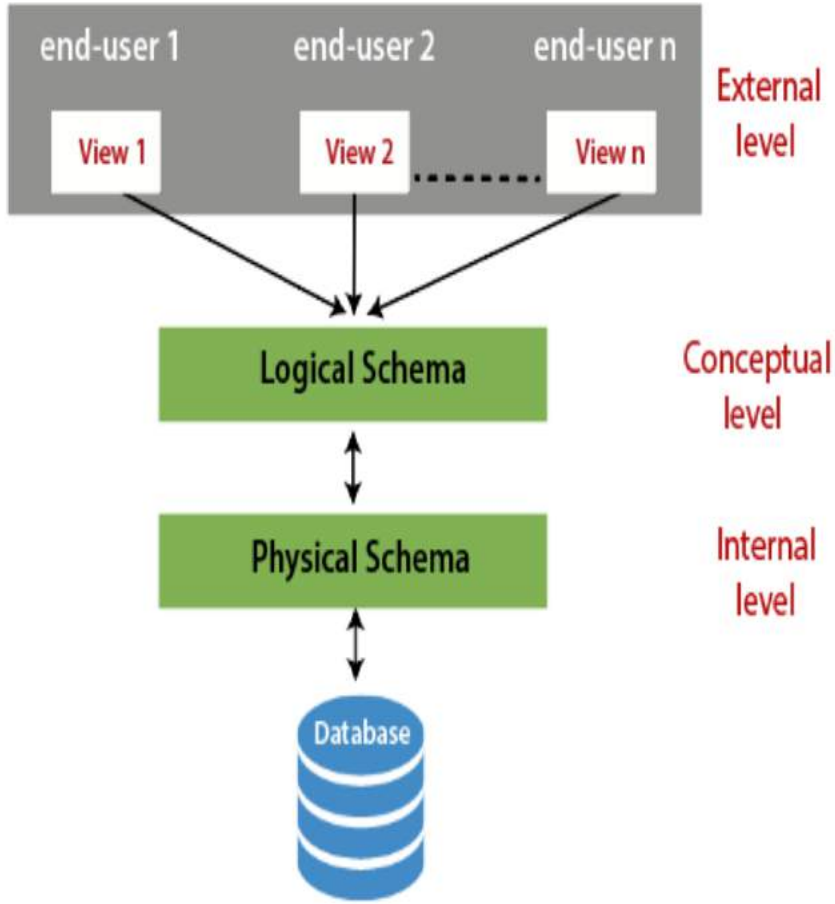
➢Hides the details of the physical storage Structure and concentrate on describing Entities, data types, relationships, constraints, etc



Three-Levels Schema Architecture/ Data Abstraction

- External Level:

➢Describes the part of the database that a user is interested in and hides the rest of the Database from the user group.



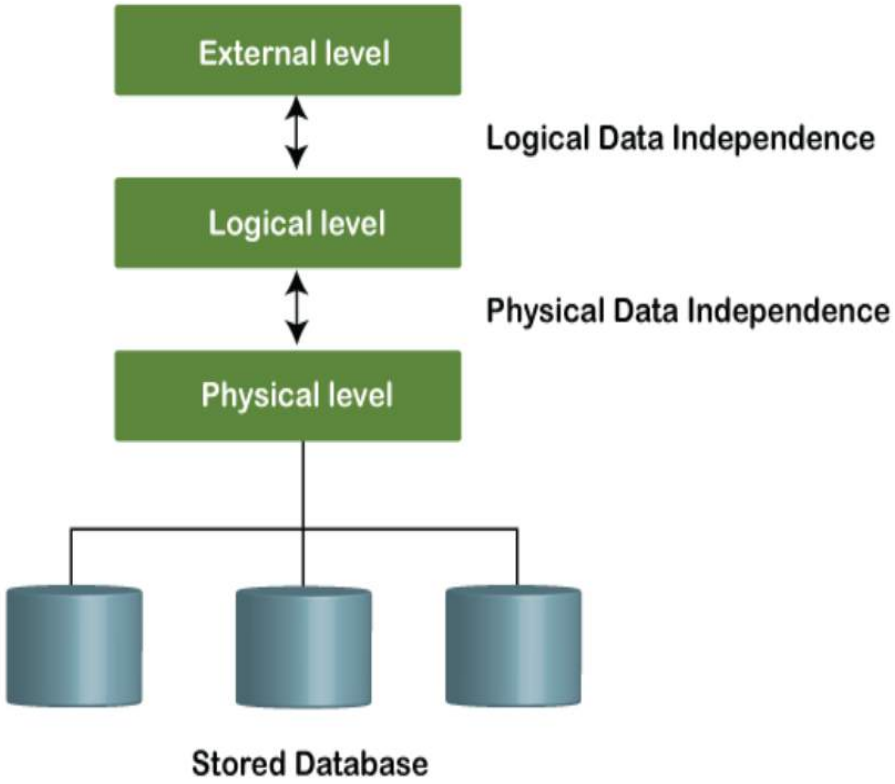
Advantages of Three-schema Architecture

- This architecture makes the database abstract. It is used to hide the details of how data is physically stored in a computer system, which makes it easier to use for a user.
- This architecture allows each user to access the same database with a different customized view of data.
- This architecture enables a database admin to change the storage structure of the database without affecting the user currently on the system.

Data Independence

Defn:

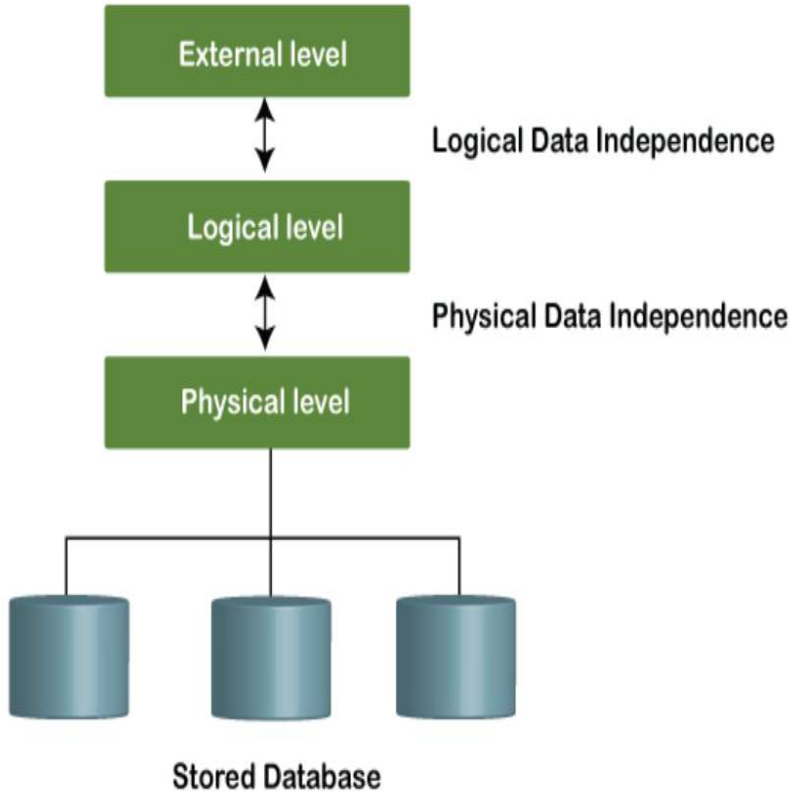
➢Capacity to change the schema at one level of a database system without having to change the schema at the next higher level.



Data Independence

❖Logical Data Independence:

➢Ability to modify the conceptual schema without changing the external schemas or application programs.

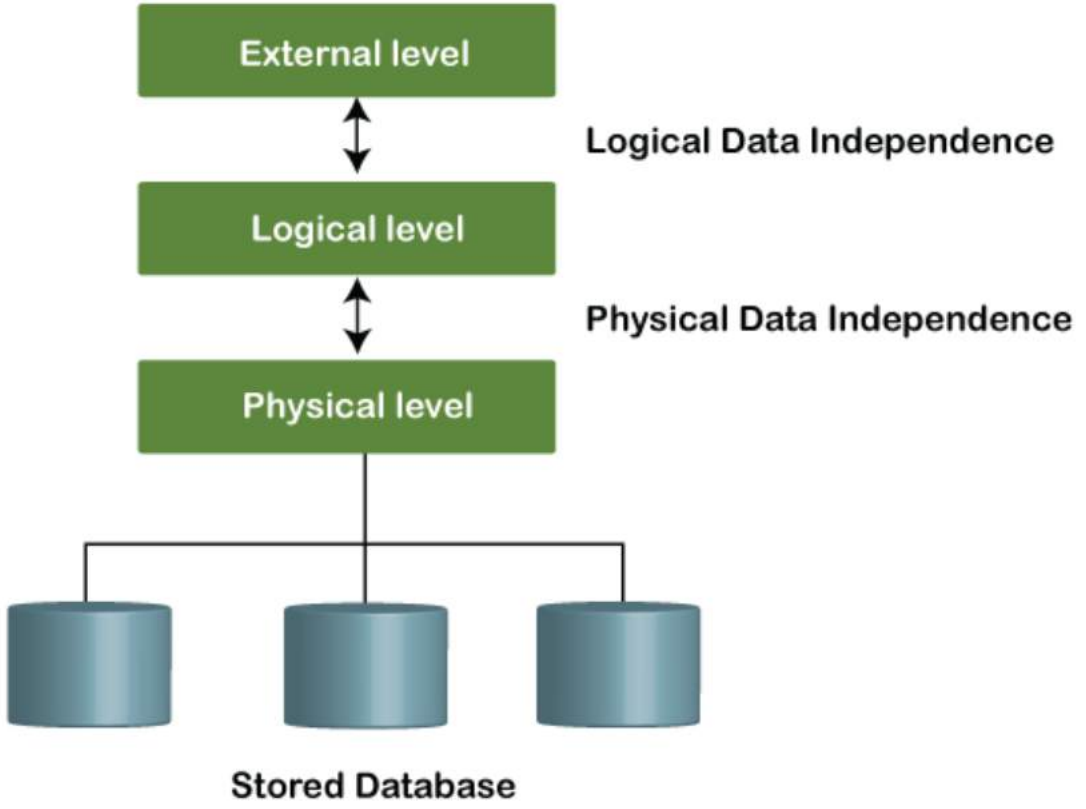


Data Independence

❖Physical Data Independence:

➢Ability to modify the internal schema without changing the Conceptual schema.

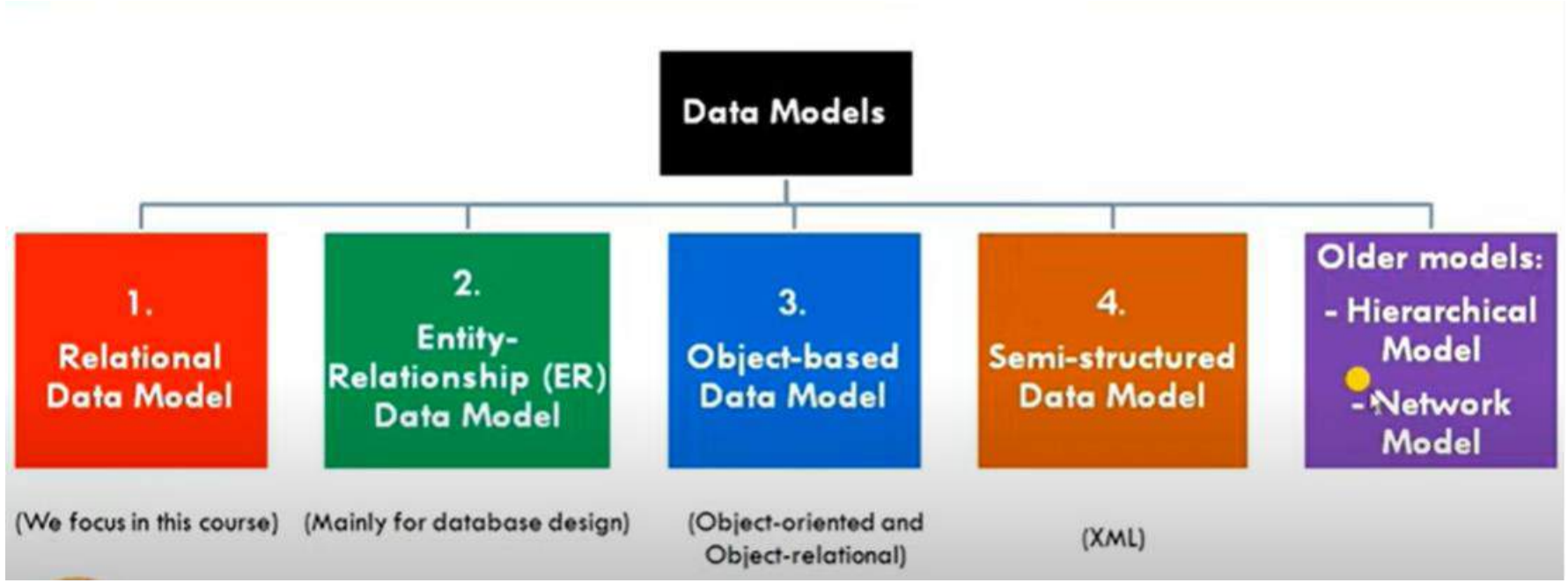
➢Changes may be needed to improve the performance.



Data Models in DBMS

- Data Model gives us an idea how the final system will look after its implementation.
- A Data Model in DBMS, is the concept of tools that are developed to summarize the description of the database.
- It defines how the logical structure of a database is modeled,.
- A Data Model is collection of conceptual tools for describing:
  - Data
  - Data Relationships
  - Data semantics
  - Consistency Constraints
- It describes the design of a database at each level of data abstraction.
- It defines how data is connected to each other and how they are processed & stored inside the system

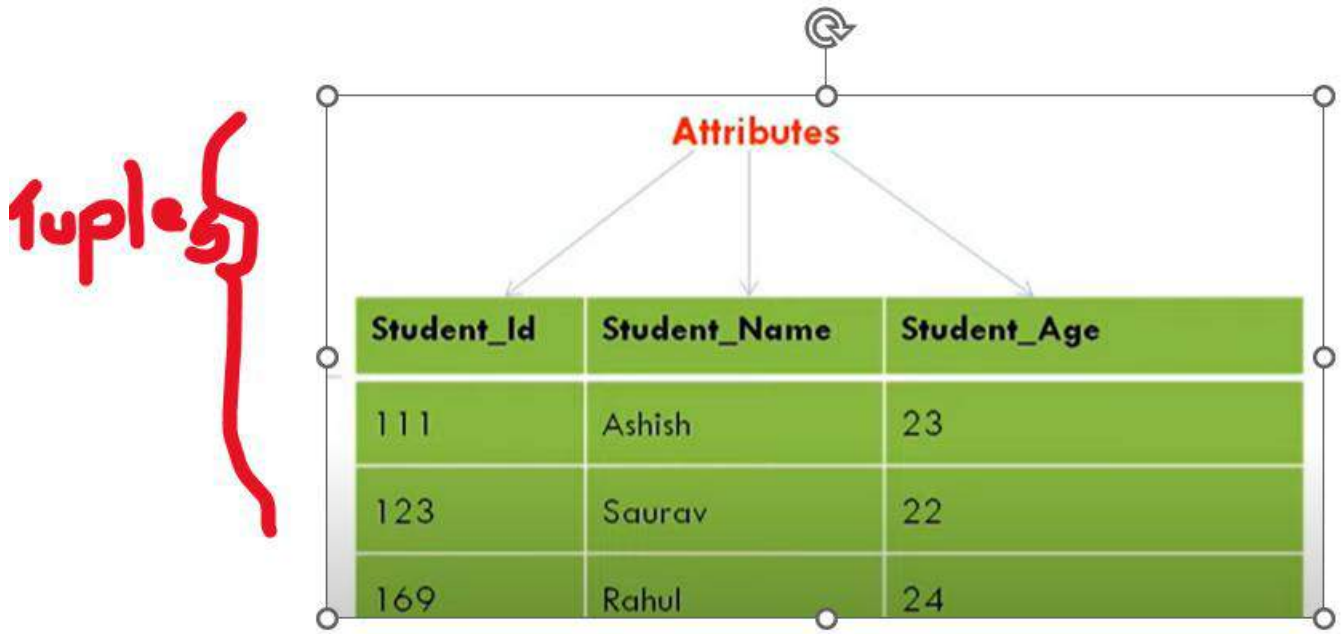
Types of Data Models.



1.Relational Model

- Most widely used model by commercial data processing applications
- It uses collection of tables for representing data and the relationships among those data.
- Data is stored in tables called Relations
- Each table is a group of column and rows where column represents attribute of an entity and rows represents records(or tuples)

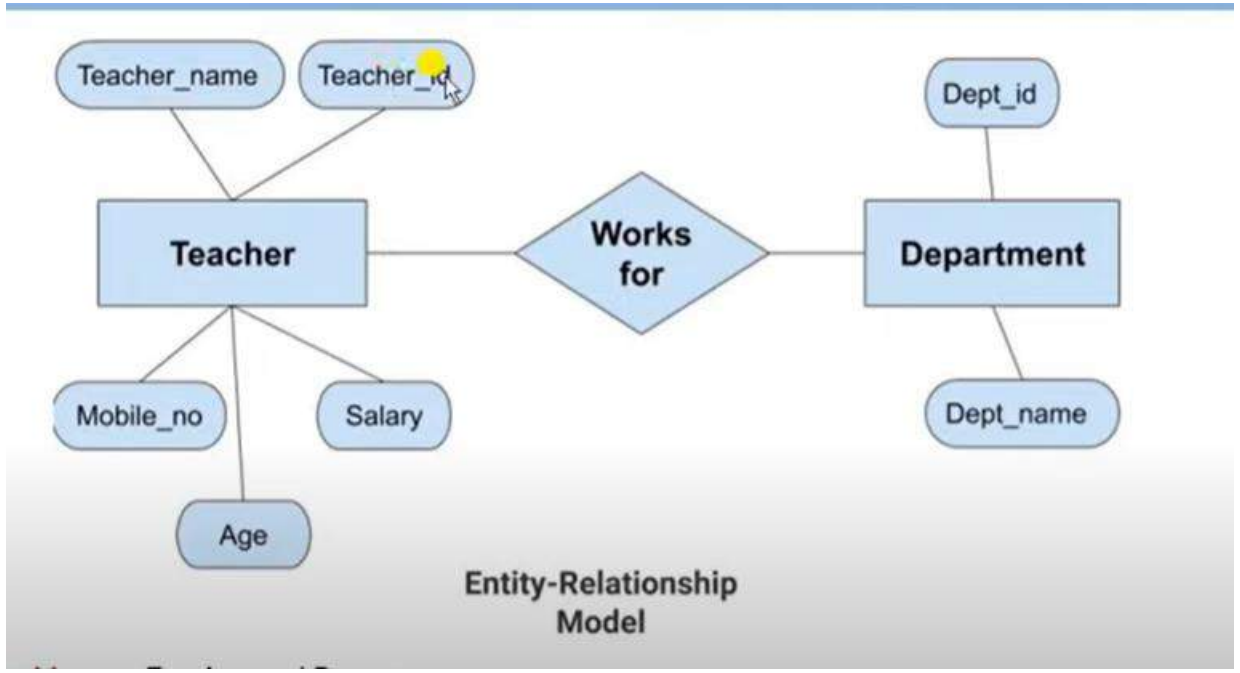
Student Table



2. Entity-Relationships Model (E-R)

- ER model is a high-level data model diagram
- ER model describes the structure of a database with the help of diagram, which is known as Entity-Relationship Diagram(ER Diagram)
- An ER model is a design or blue print of a database that can later be implemented as a database
- It is based on the notion of real –world entities and reltionships among them
- ER diagram has the following 3 components:

❖Entities, Attributes, Relationships


Teacher_name
Teacher_id
Dept_id
Teacher
Works for
Department
Mobile_no
Salary
Dept_name
Age
Entity-Relationship Model

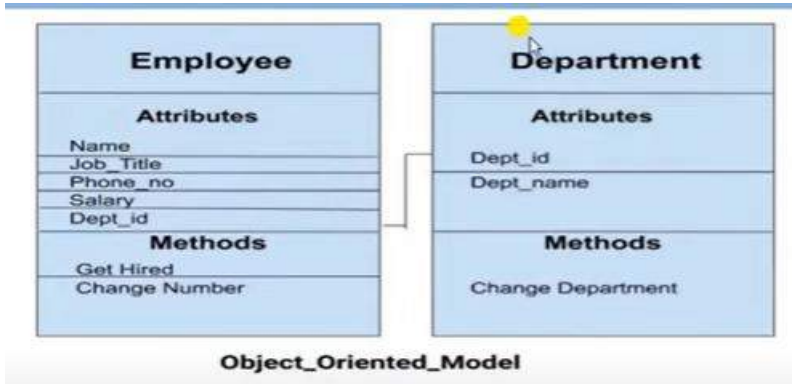
Object –based Data Model

- Two Types:

1. Object-oriented data model
2. Object-relational data model

1.Object-oriented data model

- An extension of the Er model with notions of functions, encapsulation, and object identity as well.
- In this model, both the data and relationships are represent are present in a single structure known as an object.
- Two or more objects are connected through links.We use this link to related one object to other objects.



Object-relational data model

- It is a combination of the object-oriented data model and relational data model
- This model was built to fill the gap between object-oriented model and the relational model
- It has many advanced features like complex data types that can be formed using the existing data types.
- The problem with this model is that this can get complex and difficult to handle. So, proper understanding of this model is required.

4.Semi-structued Data model

- Semi-structured model is an evolved form of the relational model.
- The semi-structured data model allows the data specifications at places where the individual data items of the same type may have different sets of attributes.
- In this model, some entities may have missing attributes while others may have an extra attribute.
- This model gives flexibility in storing the data. It also gives flexibility to the attributes.
- Eg: If we are storing any value in any attribute then that vale can be either atomic values or a collection of values.

- The Extensible Markup Language (XML) is widely used for representing the semi-structured data
- In XML we can create use tags and use different makr ups to describe the data

Example: XML or JSON

```
<student 1>
    <Roll. No.>.....</Roll. No.>
    <Name>.....</Name>
     <Class>.....</Class>
     <Age>.....</Age>
</student 1>

<student 2>
    <Name>.....</Name>
     <Class>.....</Class>
     <Age>.....</Age>
</student 2>
```

Older Data Models

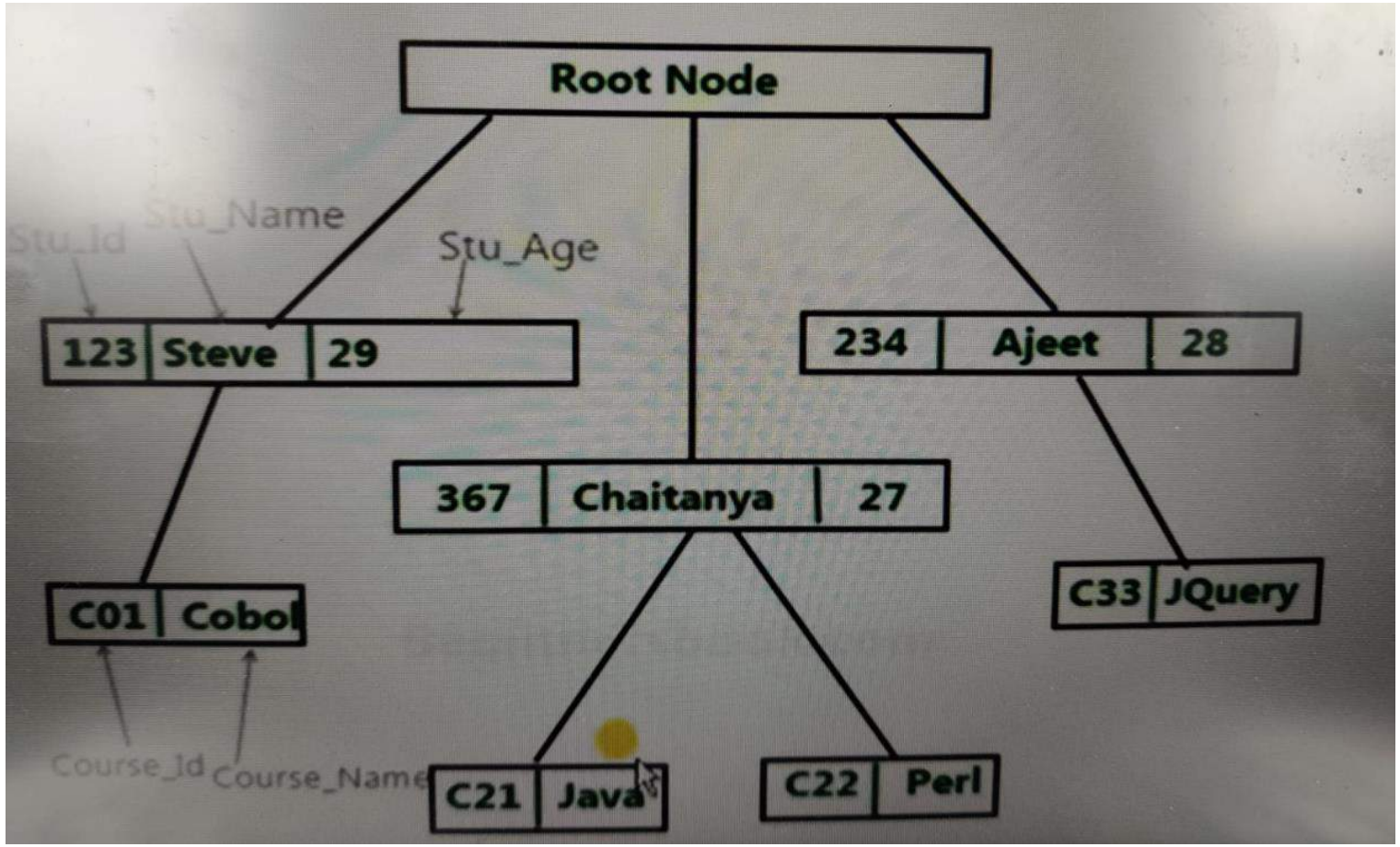
- Hierarchical Data Model
- Network Data Model

- Hierarchical Data Model & Network Data Model preceded the Relational data model
- But today they are accepted by Relational data model

Hierarchical Data Model

- It was the first DBMS model
- In Hierarchical Data Model , data is organized into a tree like structure with each record is having one parent record and many children.
- The main drawback of this model is that, it can have only one to many relationships between nodes.
- Hierarchical Data Model are rarely used now.

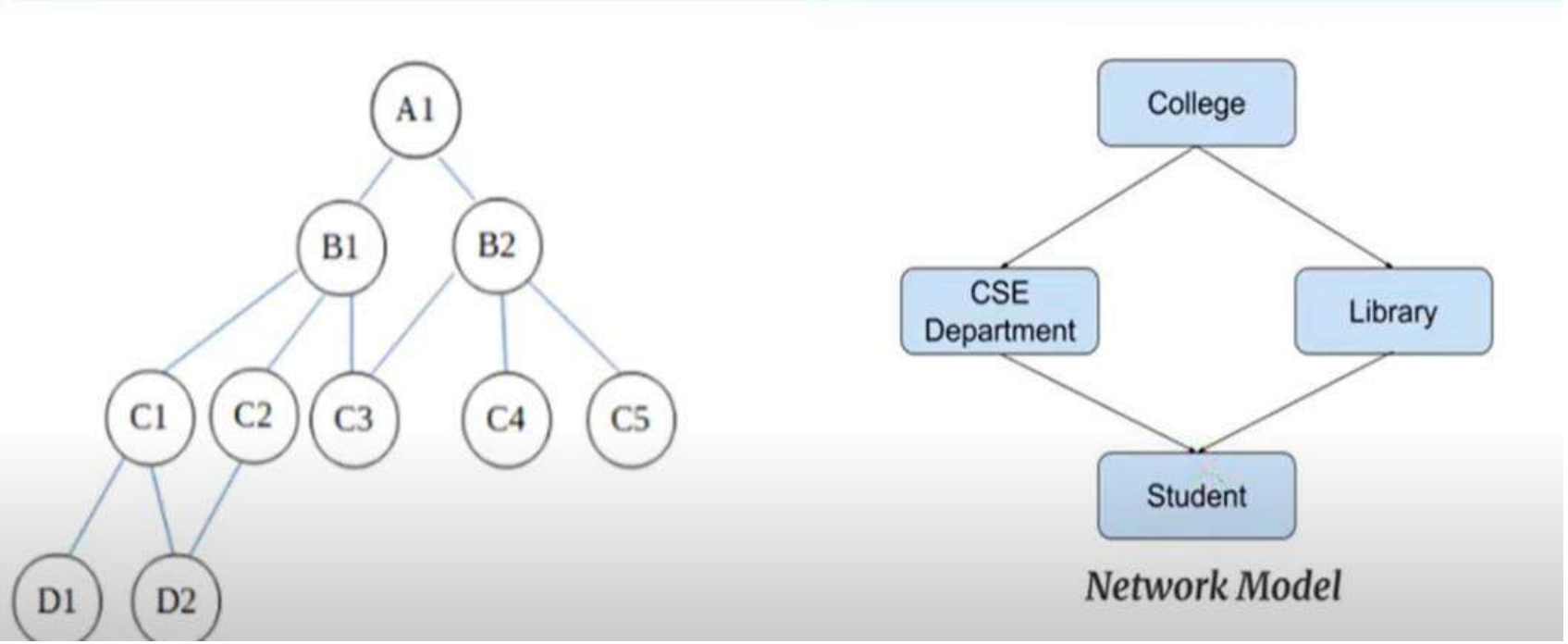
Eg : Hierarchical Data Model



Network Model

- This model is an extension of the Hierarchical Data Model .It was the most popular model before the relational model
- Network Model is a same as Hierarchical Data Model except that it has graph-like structure rather than a tree-based structure and are allowed to have mode than one parent node.
- It supports many-to-many data relationships
- This was the most widely used database model,before Relational Model was introduced.

Eg of Network Model



Database users

- Persons who interact with the database and take the benefits of the database
- Users are differentiated by the way expect to interact with the system.
- Four types of users:
- 1. Naive users/ End users
- 2.Application programmers
- 3.Sophisticated users
- 4.Specialized users

1. Naïve users/End users

- They are the unsophisticated users who use the existing application to interact with the database
- Eg: People accessing database over the web, clerical staffs etc

2. Application Programmers:

- They are the computer professionals who write the application programs. They interact with system through DML queries
- Eg: Writing C prog to generate the report of emp who are working in particular dept, will require a query to fetch data from the database.

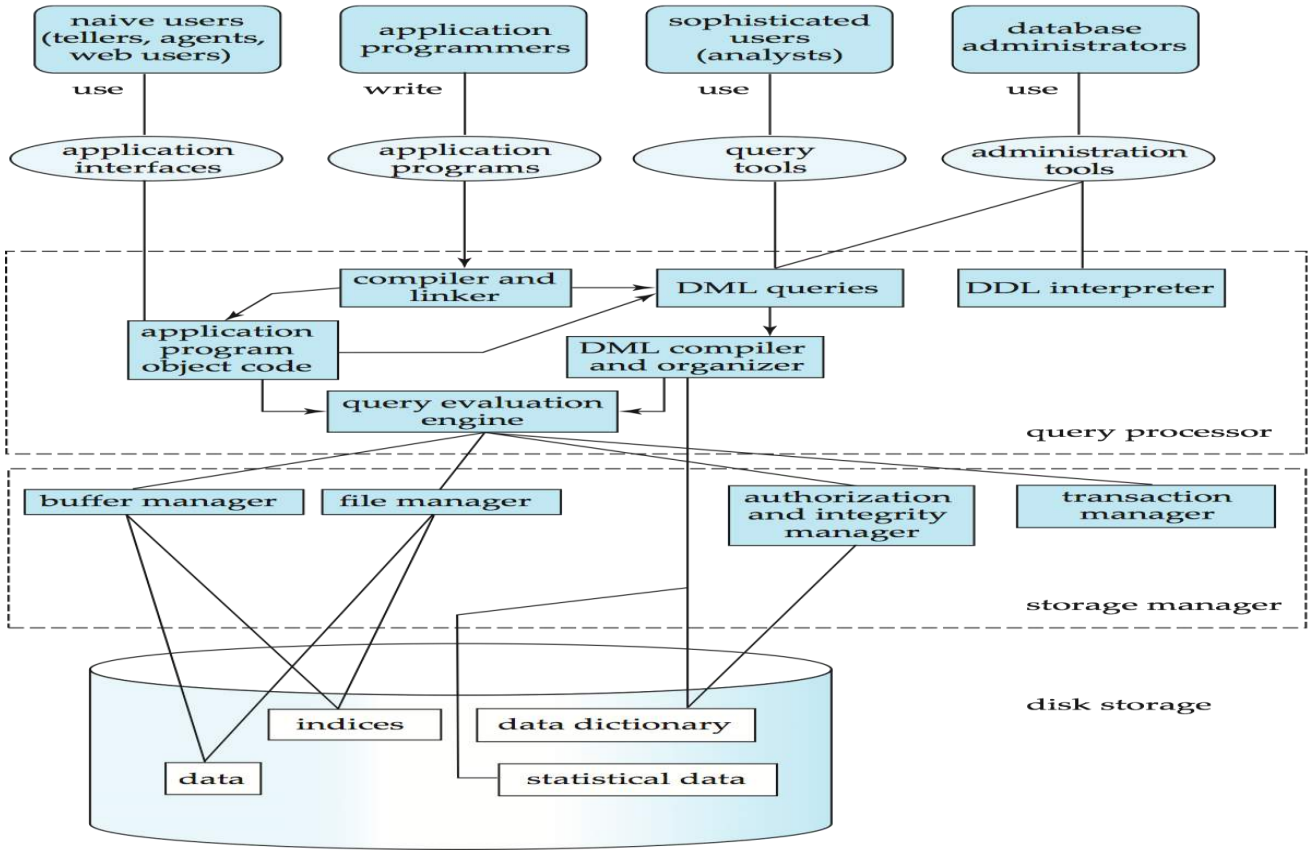
- 3. Sophisticated Users:
- They interact with the system by writing the SQL queries directly through the query processor (like SQL) without writing Application prog.
- Eg: Analyst who submits SQL queries to explore data in the DBMS

- 4. Specialized Users:
- They are also sophisticated users who write specialized database applications that do not fit into the traditional data processing framework. They are the developers who develop the complex programs to the requirement
- Eg: Expert System

Architecture of DBMS

- The Database System is divided into 3 components:

1. Query Processor
2. Storage Manager
3. Disk Storage



About Users & Programmers

1. Native Users: Unsophisticated Users,Interact with the system through application programs.

Eg: Online Banking screen

2. Application Programmers: Users who write & develop application programs by using different tools

3. Sophistictaed Users: Interact with the system by making request in the form of query of query language.These queries submitted to query processor.

4. Database Administrator: Handle Physical & Logical level of database.Gives privileges to users.

About Query Processor

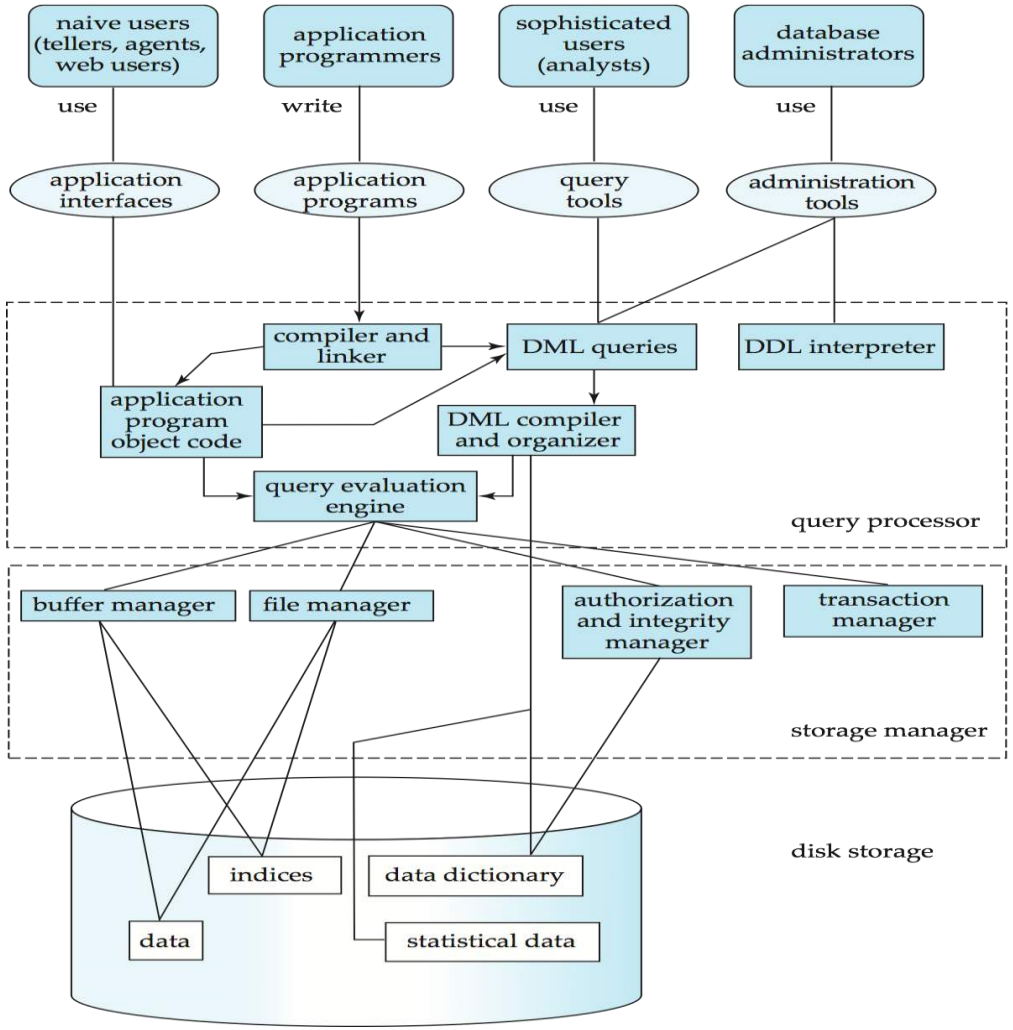
→ It Interprets the requests (queries) received from end user via an application programs into instructions

1.DML Compiler: It processes the DML statements into low level instruction (machine language)

2. DDL Interpreter: It processes the DDL statemnets into a set of table containing meta data (data about data)

3.Compiler &Linker: It processes & link DML statements embedded in an application program into procedural calls.

4.Query Evaluation Engine: It executes the instruction generated by DML Compiler



About Storage Manager

- It provides an interface between the data stored in the database & the queries received.It is also known as Database Control System

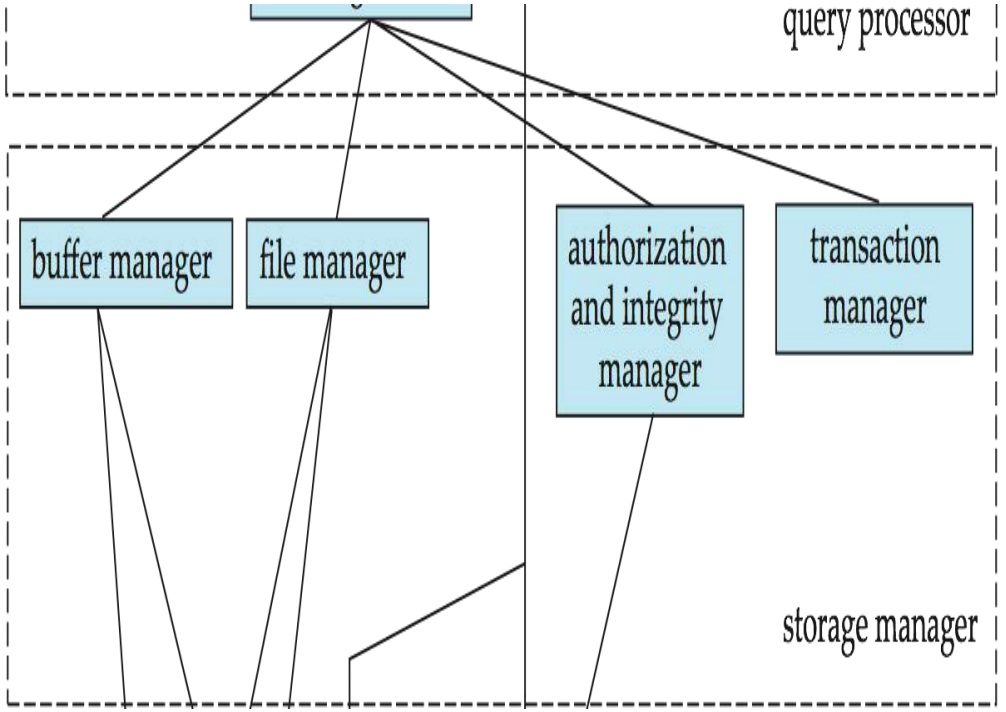
1. Authorization Manager: It ensures role-based access control, i.e checks whether the aprticular person is privileged to perform the requested operation or not.

2. Integrity Manager: It checks the integrity constraints when the database is modified.

3. Transaction Manager: It controls concurrent access by performing the operations in a scheduled way that it receives the transaction.

4. File Manager: Manages the file space and the data structure used to represent information in DB.

5. Buffer Manager: It is responsible for cache memory and the transfer of data between secondary storage and main memory



About Disk Storage

- It contains the following components:

1. Data Files:

It stores the data.

2. Data Dictionary:

It contains the information about the structure of any database object. Is is the repository of information that governs the metadata

3. Indices:

It provides faster retrieval of data item

DBA (Database Administrator)

- Can be a single person or group of users.
- Central control over both data and application programs.
- Responsible for everything that is related to database.
- Make policies ,strategies & provides technical support.

Roles / Responsibilties of DBA

- Schema definition & modification
- Granting of Authorizationof data access
- Routine maintenance
- New software installation
- Monitoring performance
- Security enforcement & administration
- Deciding the storage structure & access strategy
- Database availability
- Training & supporting users