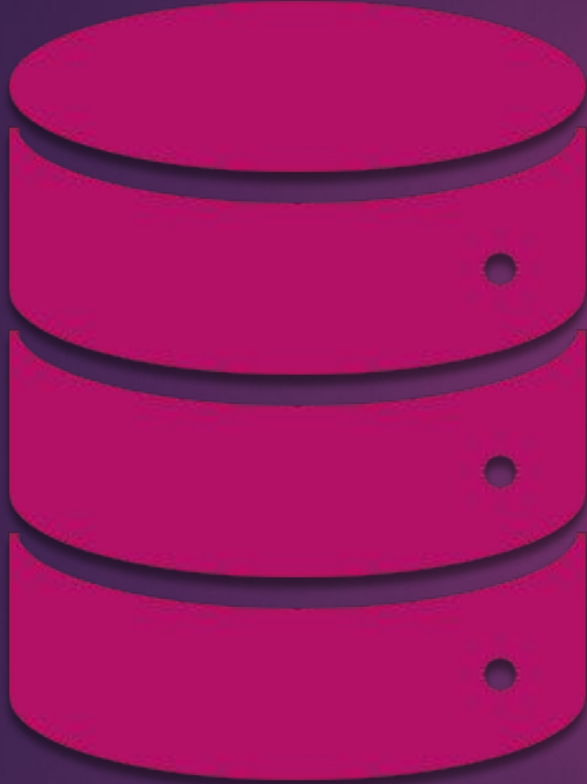


Database Systems

Week 2

Lecture 1& 2



Topics to Cover

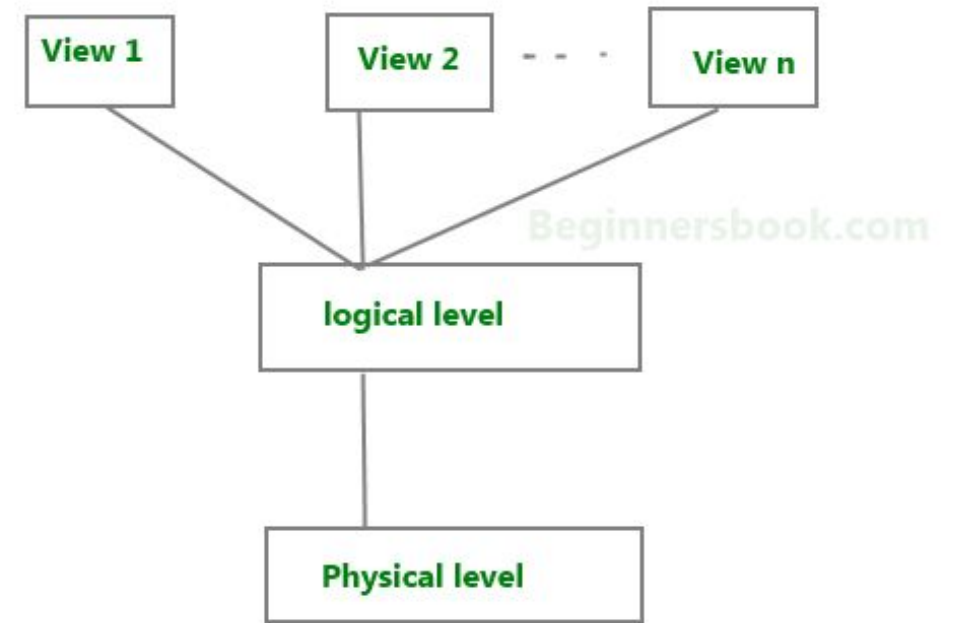
- **VIEWS OF DATA IN DBMS**
- **DATA ABSTRACTION**
- **DATA MODELS**
- **DBMS ARCHITECTURE.**
- **TYPES OF DBMS ARCHITECTURE.**
- **DATA INDEPENDENCE**
- **TYPES OF DATA INDEPENDENCE**

View of Data in DBMS

- ▶ It refers that how data is stored in database, what data and structure of data used by database for data, so describe all these database provide user with views, and these are
 - ▶ Data abstraction
 - ▶ Instances and Schemas

Data Abstraction in DBMS

Database systems are made-up of complex data structures. To ease the user interaction with database, the developers hide internal irrelevant details from users. This process of hiding irrelevant details from user is called data abstraction.



Three Levels of data abstraction

Three levels of abstraction



Physical level: This is the lowest level of data abstraction. It describes how data is stored in database. You can get the complex data structure details at this level.



Logical level: This is the middle level of 3-level data abstraction architecture. It describes what data is stored in database.



View level: Highest level of data abstraction. This level describes the user interaction with database system.

Example

- ▶ Let's say we are storing customer information in a customer table. At **physical level**, these records can be described as blocks of storage (bytes, gigabytes, terabytes etc.) in memory. These details are often hidden from the programmers
- ▶ At the **logical level**, these records can be described as fields and attributes along with their data types, their relationship among each other can be logically implemented. The programmers generally work at this level because they are aware of such things about database systems.
- ▶ At **view level**, user just interact with system with the help of GUI and enter the details at the screen, they are not aware of how the data is stored and what data is stored; such details are hidden from them.

Instance and schema in DBMS

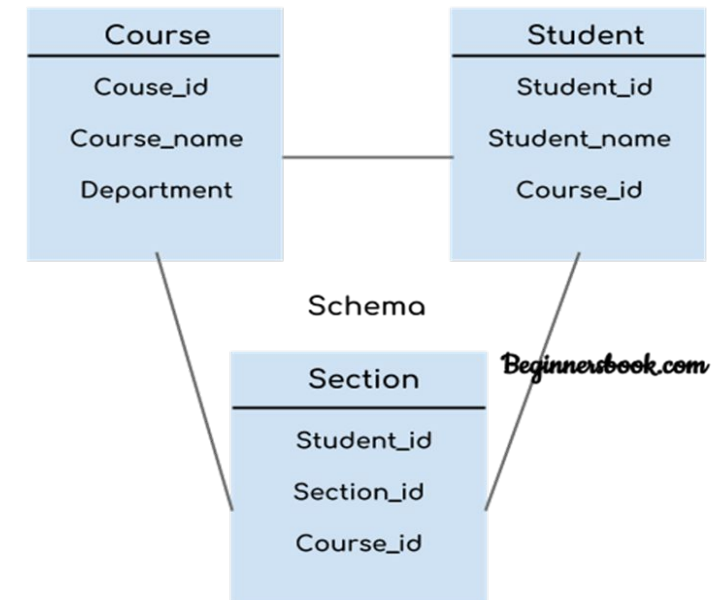
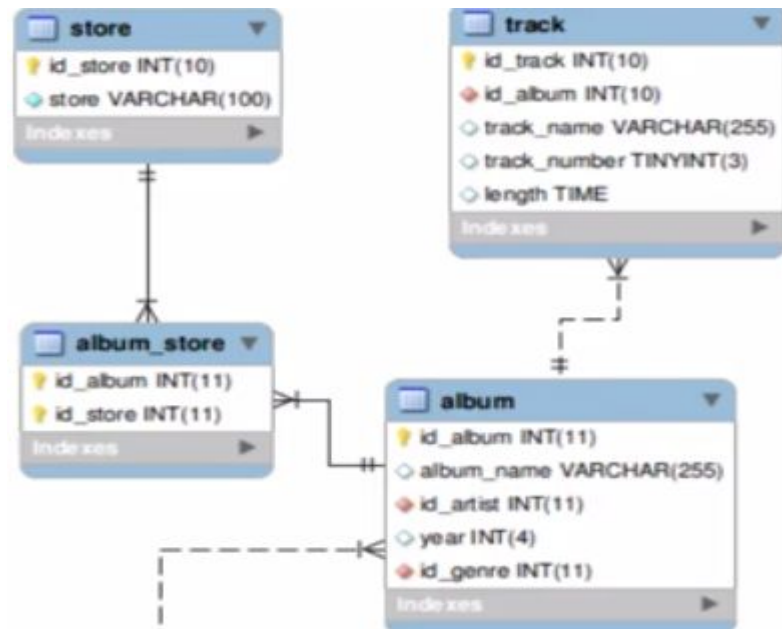
- ▶ Design of a database is called the schema.
- ▶ It is basically a Skeleton structure that represents the logical view of entire Database
- ▶ Schema is of three types:
 - ▶ **Physical schema**
 - ▶ **logical schema**
 - ▶ **view schema.**

Cont'd

- ▶ The design of a database at physical level is called **physical schema**, how the data stored in blocks of storage is described at this level.
- ▶ Design of database at logical level is called **logical schema**, programmers and database administrators work at this level, at this level data can be described as certain types of data records gets stored in data structures, however the internal details such as implementation of data structure is hidden at this level (available at physical level).
- ▶ Design of database at view level is called **view schema**. This generally describes end user interaction with database systems.

For example:

- In the following diagram, we have a schema that shows the relationship between three tables: Course, Student and Section. The diagram only shows the design of the database, it doesn't show the data present in those tables. Schema is only a structural view(design) of a database as shown in the diagram below.



DBMS Instance

- ▶ The data stored in database at a particular moment of time is called instance of database. Database schema defines the variable declarations in tables that belong to a particular database; the value of these variables at a moment of time is called the instance of that database.
- ▶ The schema is sometimes called the **intension** of the database, while an instance is called an **extension** (or **state**) of the database.

For example

let's say we have a single table student in the database, today the table has 100 records, so today the instance of the database has 100 records. Let's say we are going to add another 100 records in this table by tomorrow so the instance of database tomorrow will have 200 records in table. In short, at a particular moment the data stored in database is called the instance, that changes over time when we add or delete data from the database.

Data models

1. HIGH-LEVEL CONCEPTUAL MODELS OR DATA
2. REPRESENTATIONAL IMPLEMENTATION MODELS OR DATA
3. LOW-LEVEL OR PHYSICAL DATA MODELS

Data Models

- ▶ A collection of concepts that can be used to describe the structure of a database—provides the necessary means to achieve this abstraction.
- ▶ By structure of a database, we mean the data types, relationships, and constraints that apply to the data.
- ▶ Most data models also include a set of basic operations for specifying retrievals and updates on the database.

Categories of Data Models

1. **High-level or conceptual data models:**
 - ▶ Provides concepts that are close to the way many users perceive data.
 - ▶ Use concepts such as entities, attributes and relationships.
 - ▶ **Entities:** represent real world object or concept.
 - ▶ **Attributes:** further describe an entity.
 - ▶ **Relationships:** association among 2 or more entities.

Low-level or physical data model

2. Low-level or physical data model:

- ▶ Describe how data is stored in the computer.
- ▶ Access path (structure for efficient searching of database records]

3. Representation or implementation data model:

- ▶ Represent data using record structures [record-based data models].

- ▶ **Database Schema:** Description of a database.
- ▶ **Schema Diagram:** Displayed schema.
- ▶ **Schema Construct:** each object within the schema.
- ▶ **Database State:** The data in the database at a particular moment.

DBMS Architecture



Database architecture uses programming languages to design a particular type of software for businesses or organizations.



Database architecture focuses on the design, development, implementation and maintenance of computer programs that store and organize information for businesses, agencies and institutions.



A database architect develops and implements software to meet the needs of users.

Types of DBMS Architecture

- ▶ **The logical DBMS architecture:** The logical architecture deals with the way data is stored and presented to users.
- ▶ **The physical DBMS architecture:** The physical architecture is concerned with the software components that make up a DBMS.
- ▶ Note: we are only concerned with the logical architecture

Three-Schema Architecture

The main objective of this architecture is to have an effective separation between the user interface and the physical database. So, the user never has to be concerned regarding the internal storage of the database and it has a simplified interaction with the database system.

Three Level Architecture

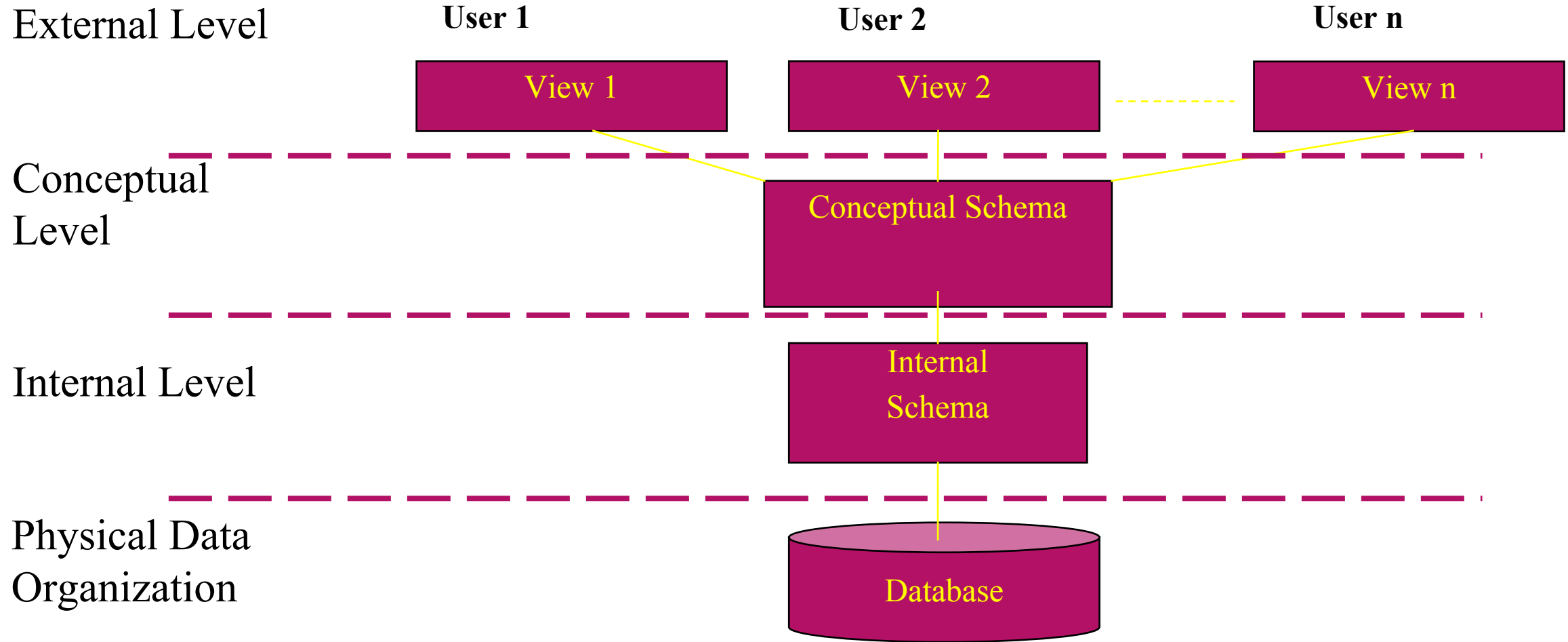
**External/view
Level**

**Conceptual/Logical
Level**

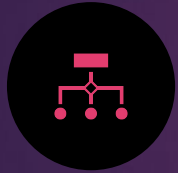
**Internal/Physical
Level**

Three Level Architecture

21



External Schema



This level is closest to the users and is concerned with the way in which the data is viewed by individual users.



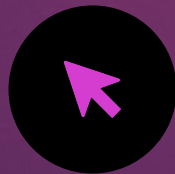
Most of the users are not concerned with all the information contained in the database.



Instead, they need only a part of the database relevant to them. The system provides many views for the same database.



Includes Virtual/calculated data: that is not actually stored in the database but is created when needed e.g., age, statistical data etc.



Also, it should not allow the user to get or retrieve data which is not allowed to the user, so the user view should both be a facilitator and also a barrier for proper utilization of the database system.

Employee Data
First Name: Rana Last Name: Aslam Date of Birth: 12 Sep 1970

Saleema



Workers
Name: R. Aslam Age: 25y,10d Dept: Sales

Saleem

External Layer

Lower Layers

Logical / Conceptual View

This level of abstraction describes what data are stored in the database. It also describes the relationships existing among data.



At this level, the database is described logically in terms of simple data-structures.



The users of this level are not concerned with how these logical data structures will be implemented at the physical level, rather they just are concerned about what information is to be kept in the database.

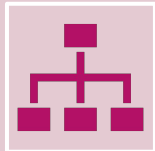


Describes what data are stored in the database (i.e., all the records and relationships included in the database).

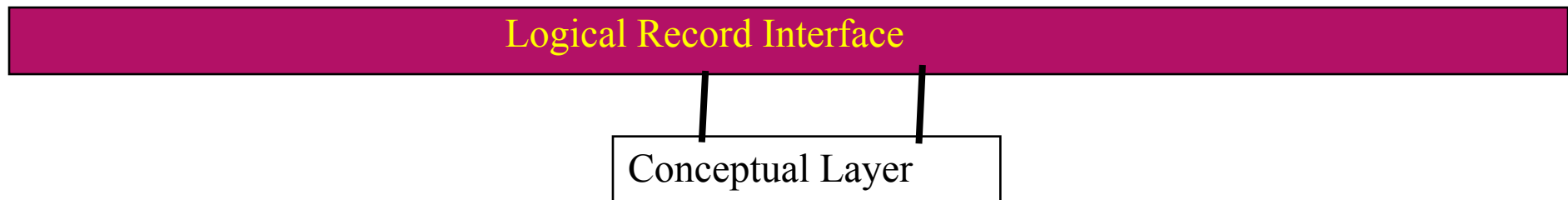
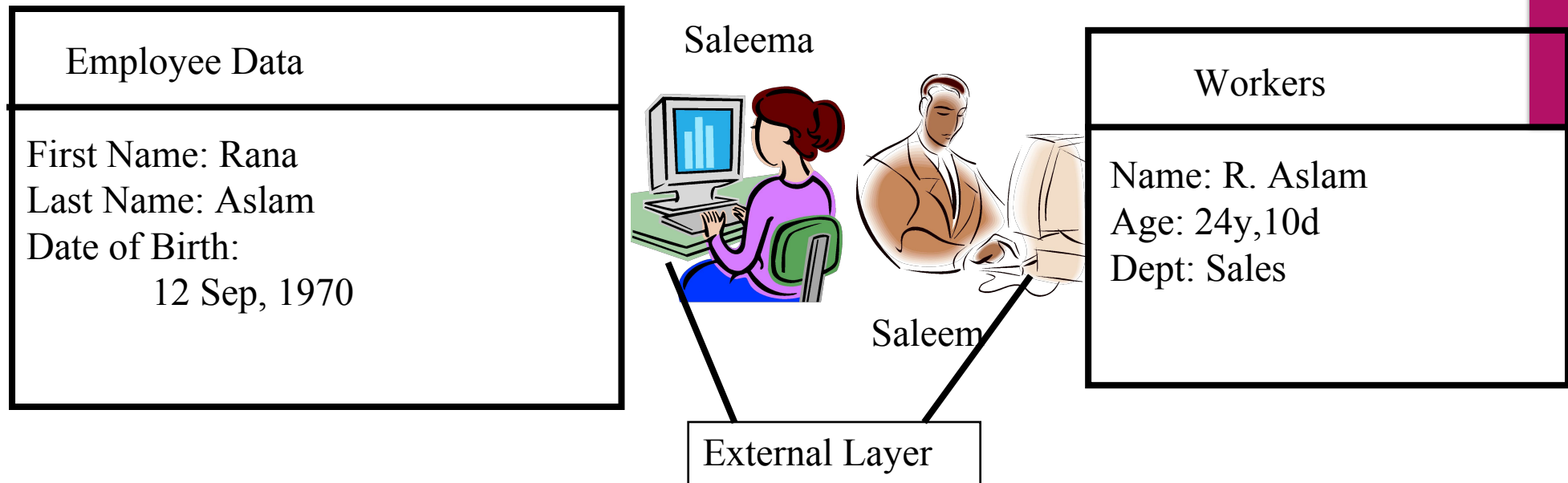
Logical / Conceptual View



The conceptual view is a representation of the entire information content of the database in a form that is somewhat abstract in comparison with the way in which the data is physically stored.



The conceptual view is defined by means of the conceptual schema, which includes the definition of each of the various types of conceptual records and the mapping between the conceptual schema and the internal schema.



<u>Name</u>	<u>DoB</u>	<u>Deps</u>	<u>Depld</u>
Rana Aslam	12/09/70	5	D001
Marya Wasti	29/02/80	0	D005

Internal View

Concerns about the physical implementation of the database

Lowest level of abstraction.

Describes how the data are physically stored.

Internal view – internal schema (not only defines the various types of stored record but also specifies what indexes exists, how files are represented, etc.)

The internal level is closest to physical storage.

Physical Level



Generally same as Internal



Actual representation of data on the storage device



In the binary format



OS responsibility

Schemas, Mappings, and Instances

- ▶ The overall description of the database is called the **database schema**. There are three different types of schema in the database, and these are defined according to the levels of abstraction of the three-level.
- ▶ At the highest level, we have multiple **external schemas** (also called **subschemas**) that correspond to different views of the data.
- ▶ At the conceptual level, we have the **conceptual schema**, which describes all the entities, attributes, and relationships together with integrity constraints.
- ▶ At the lowest level of abstraction, we have the **internal schema**, which is a complete description of the internal model, containing the definitions of stored records, the methods of representation, the data fields, and the indexes and storage structures used. There is only one conceptual schema and one internal schema per database.

Instances, Intension and extension

- ▶ The data in the database at any point in time is called a **database instance**. Therefore, many database instances can correspond to the same database schema.
- ▶ An instance of the database is a snapshot of the actual content at any given time, including the data itself and its relationship to other data in the database.
- ▶ The schema is sometimes called the **intension** of the database, while an instance is called an **extension** (or **state**) of the database.

Also, a part of 3-level
architecture

External/conceptual
mapping

Conceptual/Internal

Inter Schema Mapping

Mapping



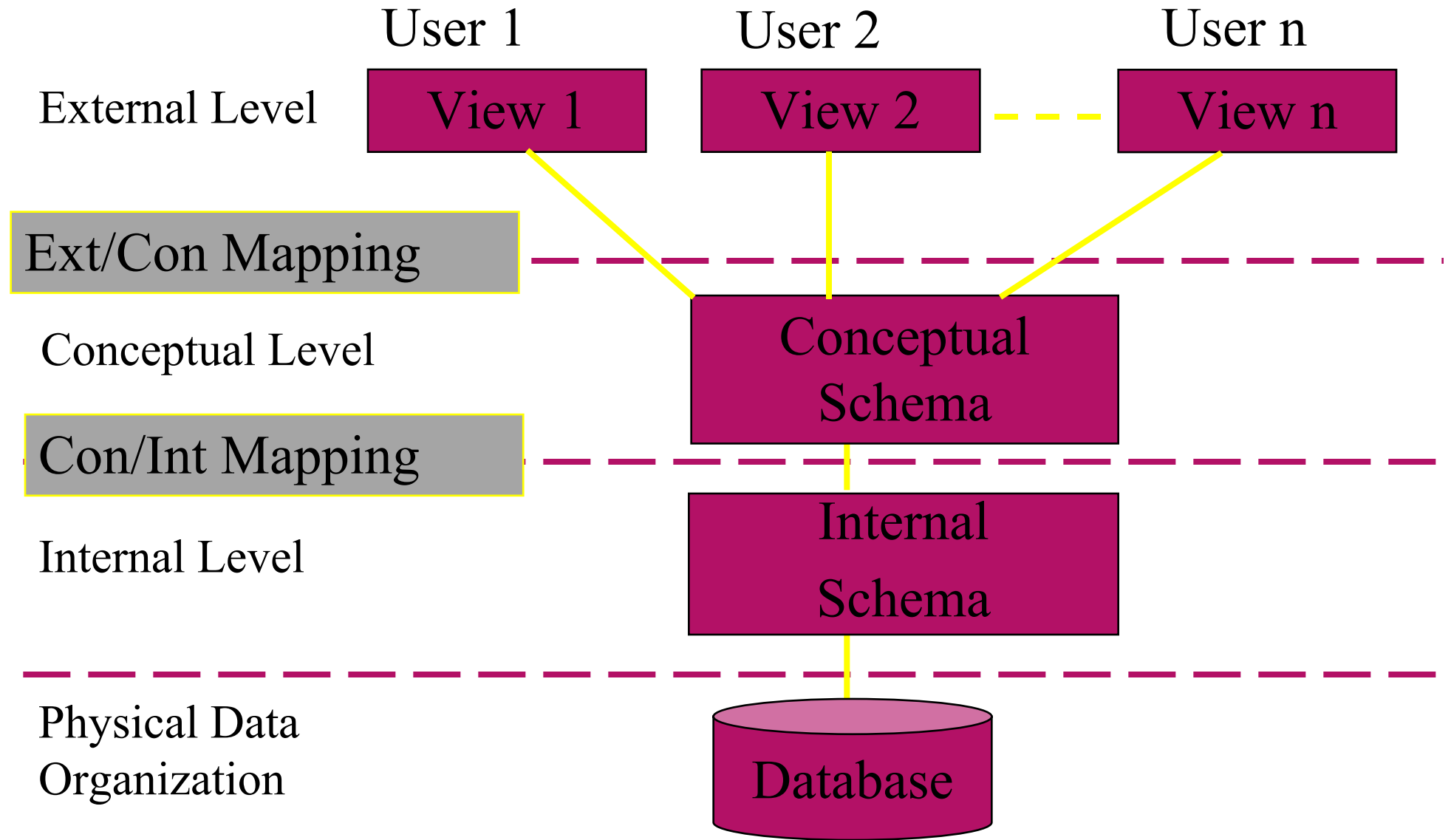
External – Conceptual Mapping

Specifies mapping between objects in the external view to those in the logical model



Conceptual – Internal Mapping

Specifies mapping between objects in the logical model to those in the physical model-data independence



First Name: Rana
Last Name: Aslam
Date of Birth:
12 Sep, 1970

Saleema



Name: R. Aslam
Age: 24y,10d
Dept: Sales

<u>Name</u>	<u>DoB</u>	<u>Deps</u>	<u>Depld</u>
Rana Aslam	12/09/70	5	D001
Marya	29/02/80	0	D005

BH RH Rana Aslam 120970 5 D001 RH Marya ...

0111001101001110010100101001010100101010010101.....

Data Independence and Its Types

The ability to modify a schema definition in one level without affecting a scheme definition in the next higher level is called **DATA INDEPENDENCE**. It has two types:

- ▶ **Physical Data Independence**
- ▶ **Logical Data Independence**



Physical Data Independence

01

It refers to the ability to modify the scheme followed at the physical level without affecting the scheme followed at the conceptual level.

02

The application programs remain the same even though the scheme at the physical level gets modified.

03

Modifications at the physical level are occasionally necessary in order to improve performance of the system.

Physical Data Independence

- ▶ Changes to the internal schema, such as using different file organizations or storage structures, using different storage devices, modifying indexes, or hashing algorithms, should be possible without having to change the conceptual or external schemas.

Logical Data Independence



It refers to the ability to modify the conceptual scheme without causing any changes in the schemes followed at view levels.



The logical data independence ensures that the application programs remain the same.



Modifications at the conceptual level are necessary whenever logical structures of the database get altered because of some unavoidable reasons.

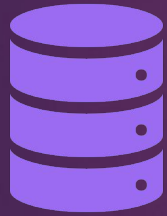
Logical Data Independence

- ▶ Changes to the conceptual schema, such as the addition or removal of new entities, attributes, or relationships, should be possible without having to change existing external schemas or having to rewrite application programs.

Physical & Logical Data Independence

- ▶ It is more difficult to achieve logical data independence than the physical data independence.
- ▶ The reason being that the application programs are heavily dependent on the logical structure of the database.

Next lecture



DB Query Language



Data models



Data Flow Diagrams