Database Engineering

Lecture #4

DBMS System Concepts and Architecture

Presented By:
Dr. Suvasini Panigrahi
Associate Professor, Department of CSE,
VSSUT, Burla

OUTLINE

- Database Schema and Instance
- Data Abstraction
- Three-Schema Architecture
- Data Independence

Database Schema

- A database schema is the skeleton structure that represents the logical view of the entire database.
- A database schema contains schema objects that may include **tables, fields, relationships, different keys like primary key, foreign key**, etc.
- It defines how the data is organized and how the relations among them are associated.
- It formulates all the constraints that are to be applied on the data.
- It contains a descriptive detail of the database, which can be depicted by means of schema diagram.
- The schema does not contain the data itself. Instead, it gives information about the data storage structures and how they are related to each other.

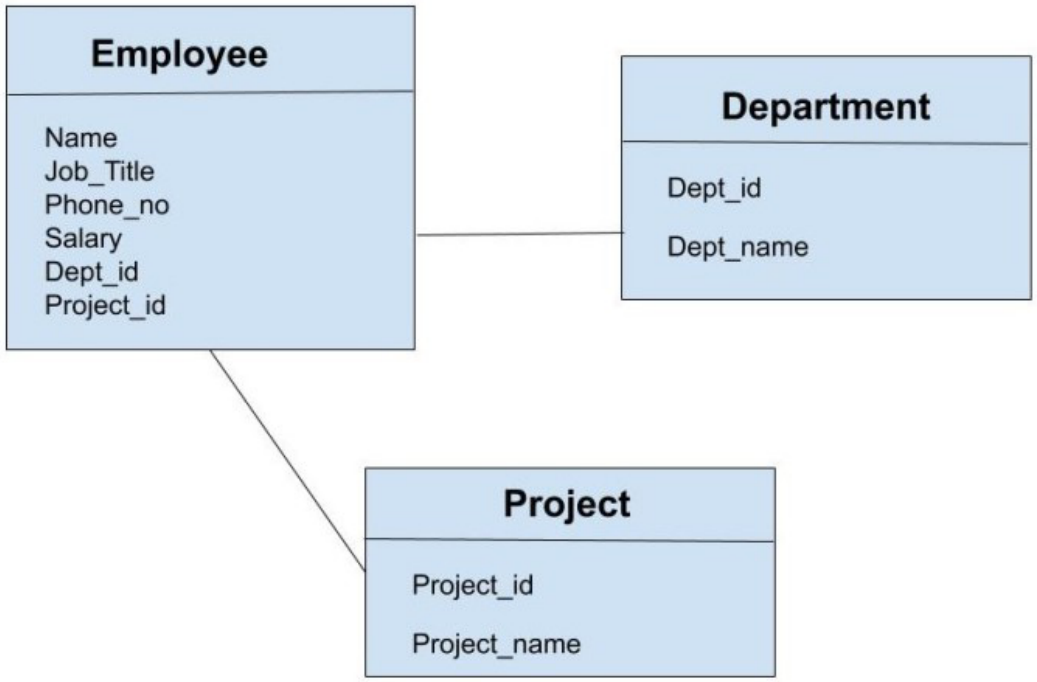
Database Schema and Schema Diagram

- A schema diagram is a diagrammatic representation of the entities and the attributes that will define that database schema.
- A schema diagram only shows us the database design, not the actual data contained in the database.
- A component of the schema is called a schema construct.
- The database designers design the schema in order to help the programmers understand the database and thus, make it useful for the database users.

Database Schema and Schema Diagram

Example: Suppose there are three tables Employee, Department and Project. So, the schema of these three tables can be represented using the schema diagram as follows.

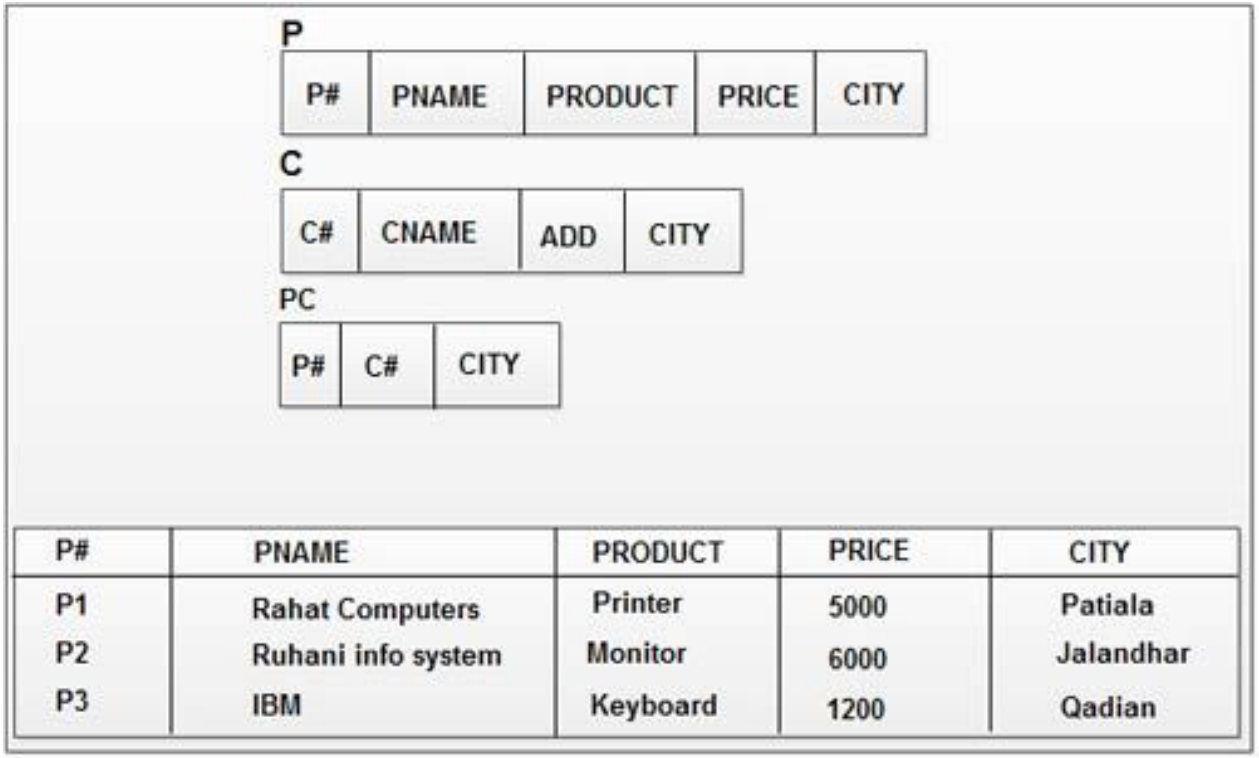
In this figure, Employee and Department tables are related and likewise, the Employee and Project tables are also related.



Schema

Database Instance/State/Occurrences/ Snapshot

- Database instance or snapshot is the actual content of the database at a particular point in time.



P#	PNAME	PRODUCT	PRICE	CITY
P1	Rahat Computers	Printer	5000	Patiala
P2	Ruhani info system	Monitor	6000	Jalandhar
P3	IBM	Keyboard	1200	Qadian

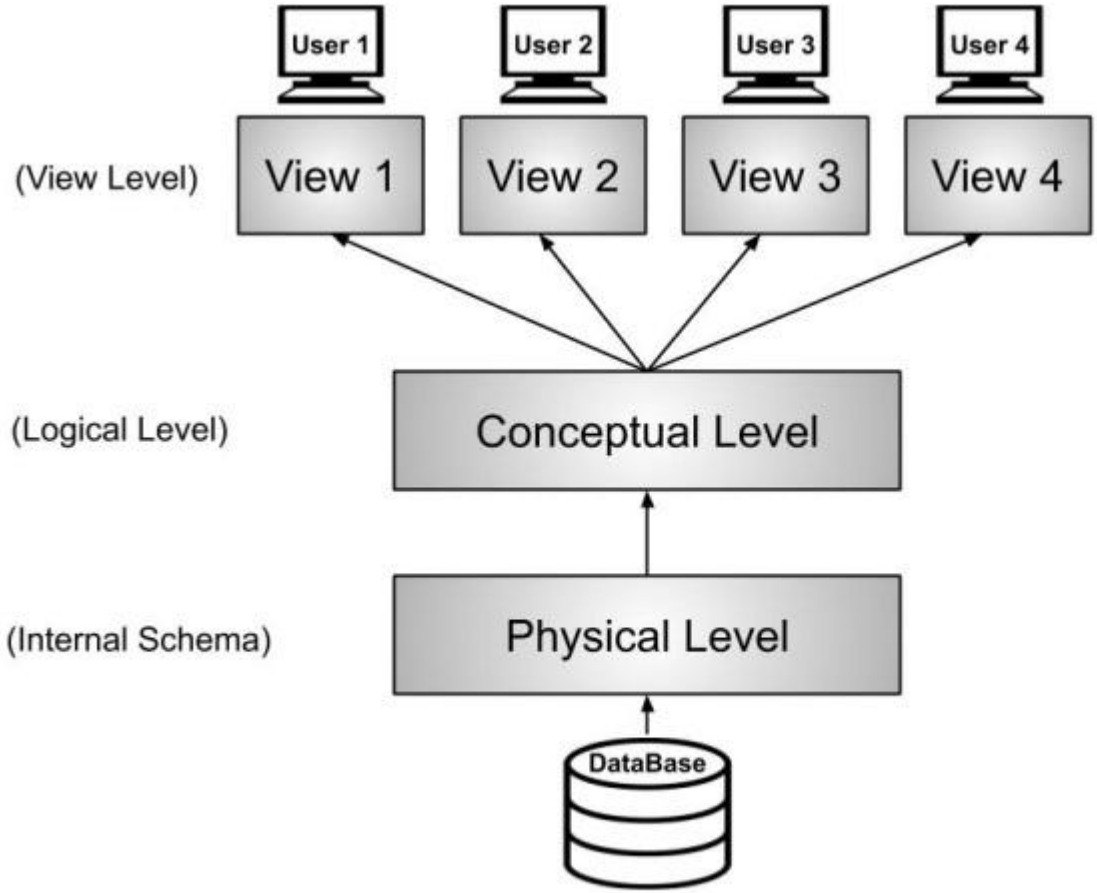
Data Abstraction

- Database systems comprise of complex data structures.
- In order to make the database system efficient in terms of retrieval of data, and reduce complexity in terms of its usability for users, developers use the concept of data abstraction so as to hide the irrelevant details from the users.
- Data Abstraction refers to the process of hiding irrelevant details from the users. The main purpose of data abstraction is to hide the unwanted data and provide an abstract view of the data.
- The data abstraction is divided into three levels in order to achieve Data Independence.
- Based on these three levels, the DBMS architecture is designed.
- The goal of the three-schema architecture is to separate the user applications from the physical database.

Level of Data Abstraction in DBMS

- There are mainly 3 levels of data abstraction in the three-schema architecture of DBMS:
  1. Physical or Internal Level
  2. Logical or Conceptual Level
  3. View or External Level

Level of Data Abstraction in DBMS



Levels of Data Abstraction

Three-Schema Architecture

- Architecture of a DBMS refers to the design of the major components of a DBMS and how they are related to each other.
- There are three levels of the schema.
- The three levels of the database schema are defined according to the three levels of data abstraction.
  - View Schema
  - Logical Schema
  - Physical Schema
  - The three-schema architecture support the following DBMS characteristics:
    - Program Data Independence and Program Operation Independence
    - Multiple User Views

Program Data Independence and Program Operation Independence

- Program Data Independence: In database approach, if any changes are made to the data structure, then the database access programs do not require changes in most cases. This is called program-data independence.
- Program Operation Independence: User application programs can operate on the data by invoking some specific operations regardless how the operations are implemented. This is called program-operation independence.
- Allowing of program-data independence and program-operation independence is called data abstraction.

Multiple User Views

- A database typically has different types of users, each of whom may require different perspective or view of the database.
- DBMS provides different views for the different types of users.

Levels of Data Abstraction

1. Physical or Internal Level:

- Internal level has an internal schema that describes the complex details of data storage and access paths for the data in the database
- This is the lowest level of data abstraction in the database management system.
- It is the layer that tells us how the data is actually stored in the database.
- It also defines methods to access the data in the database.
- It defines complex data structures in detail, so it is very complex to understand. That is why it is kept hidden from the end users.

Levels of Data Abstraction

2. Logical or Conceptual Level

- The logical or conceptual level has a conceptual schema that describes the logical structure of the whole database for a group of users.
- It is the intermediate or next level of data abstraction.
- The conceptual schema hides the details of physical storage structures and concentrates on describing the database as a group of entities, attributes along with their data types, relationships, operations and constraints.
- It describes the structure of the entire database in the form of a list of tables.
- The logical level or conceptual level is less complex than the physical level.

Levels of Data Abstraction

3. View or External Level

- View or External Level includes a number of external schemas or user views.
- Each external schema describes the part of the database that a particular user group is interested in and hides the remaining data from that user group.
- This schema provides the highest level of data abstraction.

- External level is meant for the end-user interaction. At this level, end users can access the data based on their queries.
- Users can just view the data and interact with the database. However, the storage and implementation details are hidden from them.

Mappings in the Three-Schema Architecture

- In the three-schema architecture, each user group refers to its own external schema.
- The DBMS must transform a request specified on an external schema into a request against the conceptual schema, and then into a request on the internal schema for processing over the stored database.
- If the request is a database retrieval, the data extracted from the stored database must be reformatted to match the user's external view.
- The processes of transforming requests and results between levels are called **mappings**.

Mappings in the Three-Schema Architecture

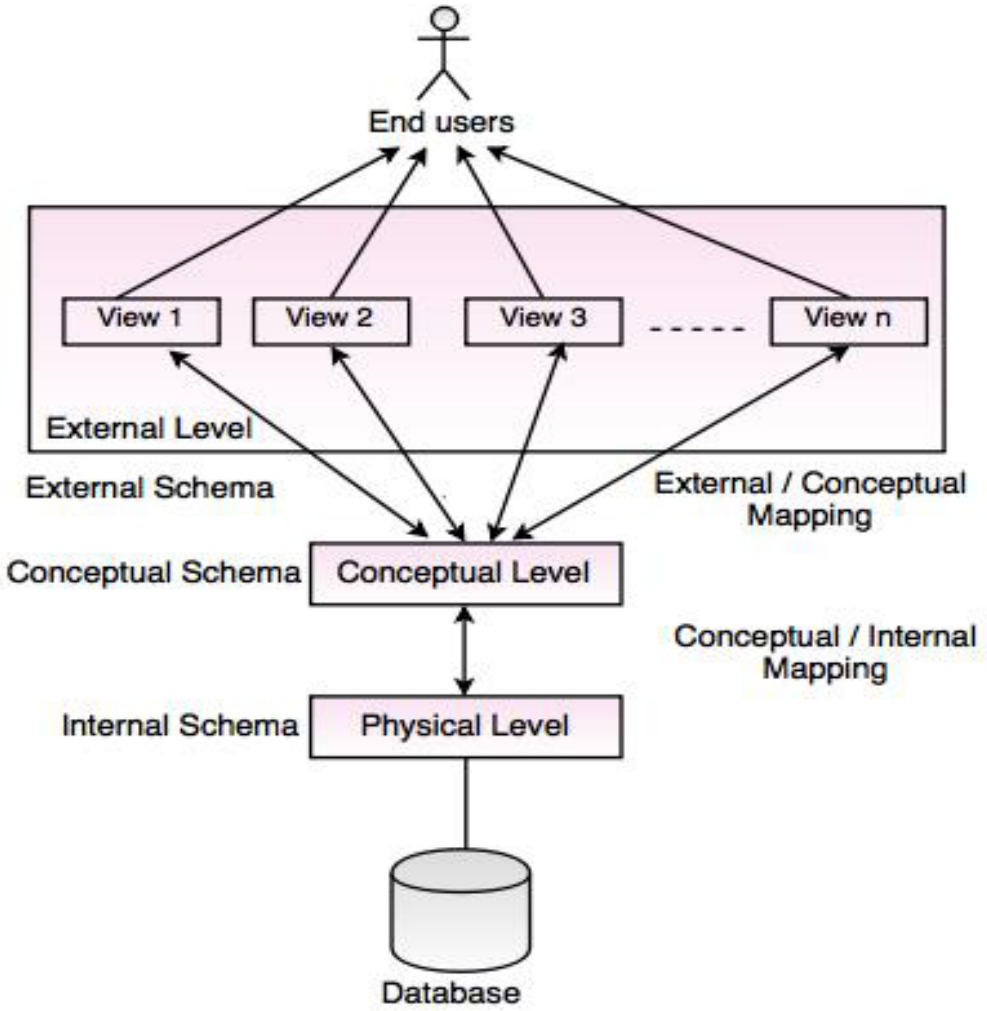


Fig. Three Level Architechture of DBMS

Mappings in the Three-Schema Architecture

- There are basically two types of mapping in the database architecture:
  1) Conceptual/ Internal Mapping
  2) External / Conceptual Mapping
- **Conceptual/ Internal Mapping**
  - The conceptual/internal mapping lies between the conceptual level and the internal level.
  - Its role is to define the correspondence between the records and fields of the conceptual level to the files and data structures of the internal level.
- **External/ Conceptual Mapping**
  - The external/conceptual mapping lies between the external level and the conceptual level.
  - Its role is to define the correspondence between a particular external view and the conceptual view.

Data Independence

- Data independence refers to the characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.
- There are two types of data independence:
  1. Logical Data Independence
  2. Physical Data Independence

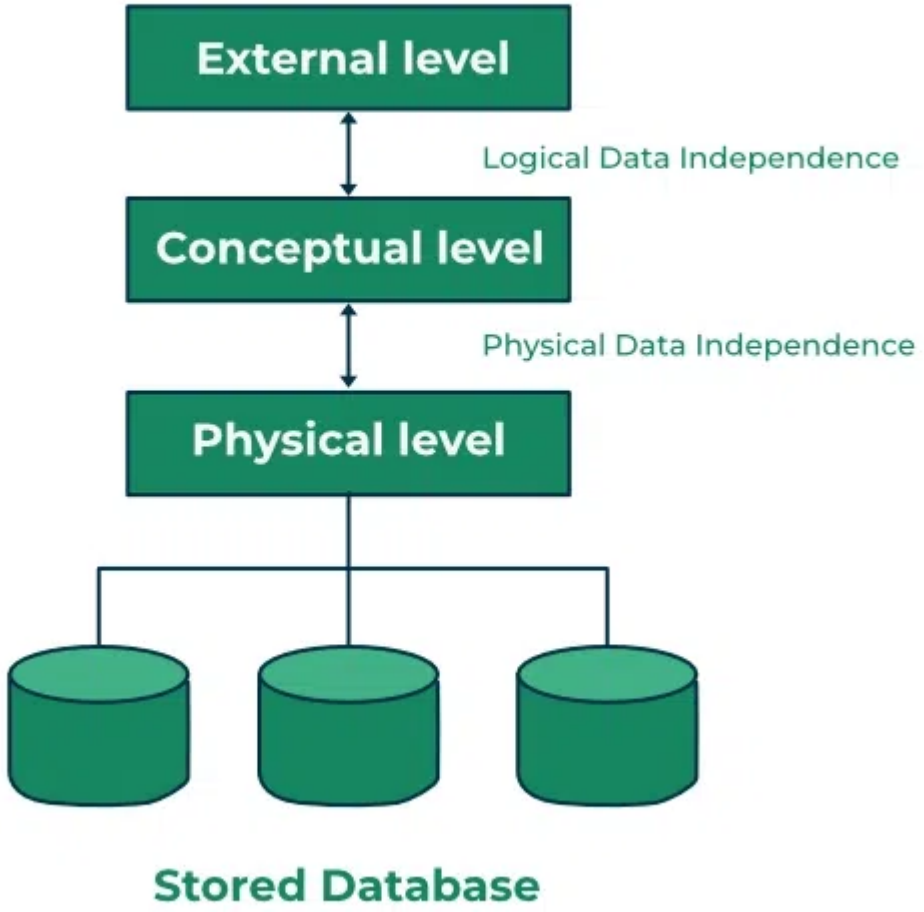
Logical Data Independence

- Logical data independence refers to the characteristic of being able to change the conceptual schema without having to change the external schema.
- Logical data independence is used to separate the external level from the conceptual view.
- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- These changes may include insertion or deletion of attributes, altering table structures or relationships between entities, etc.

Physical Data Independence

- Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.
- Physical data independence is used to separate conceptual levels from the internal levels.
- These alterations or modifications to the physical schema may include:
  - Utilizing new storage devices.
  - Modifying data structures used for storage.
  - Altering indexes or using alternative file organization techniques, etc.

Data Independence in DBMS



Data Independence

- Why is Logical Data Independence harder to achieve compared to Physical Data Independence?
  - Logical Data Independence is harder to achieve as the application programs are heavily dependent on the logical structure of the data they access, hence, a change in the conceptual level might require the change of the corresponding external schema(s).
  - When it comes to Physical Data Independence, change in the physical storage structure or modifying the file organization or use of new storage device, etc. may not require the change at the higher logical level.

Database Schema vs. Database Instance

Database Schema	Database Instance
It defines the basic structure of the database i.e., how the data will be stored in the database.	It is the data stored in the database at a particular instance of time.
Schema is same for whole database.	Data in instances can be changed using insertion, deletion and update operations.
Does not change Frequently.	It changes Frequently.
Schema is also called intension.	Instance is also called extension of the schema.

Difference between Physical Data Independence and Logical Data Independence

Physical Data Independence	Logical Data Independence
It mainly concern about how the data is stored into the system.	It mainly concerned about the structure or the changing data definition.
As compared to the logical independence it is easy to achieve physical data independence.	As compared to the physical independence it is not easy to achieve logical data independence.
Any change at the physical level, does not require to change at the application level.	The change in the logical level requires a change at the application level.
The modifications made at the internal level may or may not be needed to improve the performance of the structure.	The modifications made at the logical level is significant whenever the logical structure of the database is to be changed.
It is concerned with the internal schema.	It is concerned with the conceptual schema.
Example: Change in compression techniques, Hashing algorithms and storage devices etc.	Example: Add/Modify or Delete a new attribute

ANY DOUBTS?

THANK YOU!