Database Management System
Lecture: 4
DBMS Architecture and Data Independence
(Three-Schema Architecture)

Databases change over time as information is inserted and deleted.

Instances and Schemas:

The collection of information stored in the database at a particular moment is called an instance of the database.

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

Differences between Schema and Instance:

- A schema is the design representation of a database whereas instance is the snapshot of a database at a particular moment.
- Instance changes very frequently whenever data is removed or added in the database. As against, the changes in schema occurs rarely.

e.g. Schema and instance can be easily perceived by analogy to a program. At the time of writing a program in a programming language, the variables of that program is declared at first, this is analogous to the schema definition. Additionally, each variable in a program must have some values associated at a particular time, this is similar to an instance.

Three important characteristics of the database approach are:

1. insulation of programs and data.
2. support of multiple user views
3. use of a catalog to store the database description (schema).

Three-Schema Architecture (ANSI/SPARC Architecture):

The **three schema architecture** is used to describe the structure of a specific database system. The goal of the three-schema architecture is to separate the user applications and the physical database.

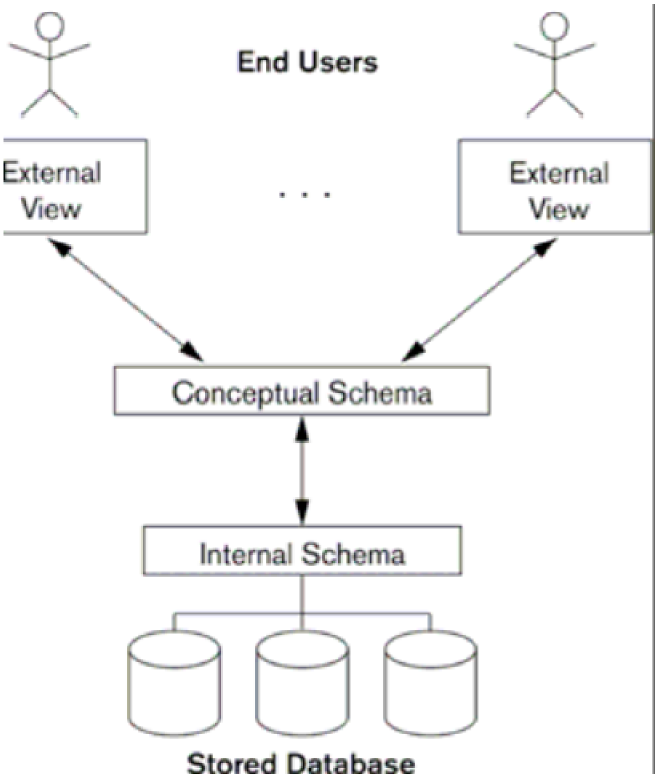

Fig: The Three Schema Architecture

In this architecture, schemas can be defined at the following three levels:

1. **The internal level (physical schema)** has an internal schema, which describes the physical storage structure of the database.
    - The physical level is used to describe complex low level data structures in detail.
    - The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.

2. **The Conceptual level (conceptual schema)** has a conceptual schema, which describes the structure of the whole database for a community of users.
    - The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operation and constraints.
    - In the conceptual level, internal details such as an implementation of the data structure are hidden.
    - Programmers and database administrators work at this level.

3. **The External Level or view 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 rest of the database from that user group.
    - The view schema describes the end user interaction with database systems.

Data Independence:

Data independence can be **defined as the capacity to change the schema at one level of a database without having to change the schema at the next higher level.**

Two types of data independence:

1. **Physical Data Independence:**
    - Physical Data Independence is defined as the ability to make changes in the structure of the lowest level of the Database Management System (DBMS) without affecting the higher level schemas.
    - The schema in the database are logical, but the actual data is stored is in bit format on the disk. Physical data independence allows to change the physical data without disturbing the schema or logical data.
    - e.g. if user want to upgrade the storage system it means he/she wants to replace hard-disks with SSD; then this change should not disturb or change the logical data or schemas.
2. **Logical Data independence:**
    - is the capacity to change the conceptual schema without having to change external schemas or application programs.
    - Due to logical independence, any of the below change will not affect the external layer:
        i. Add/Modify/Delete a new attribute, entity or relationship is possible without a rewrite of existing application programs.
        ii. Merging two records into one.
        iii. Breaking an existing record into two or more records.
    - Comparatively it is difficult to achieve logical data independence.