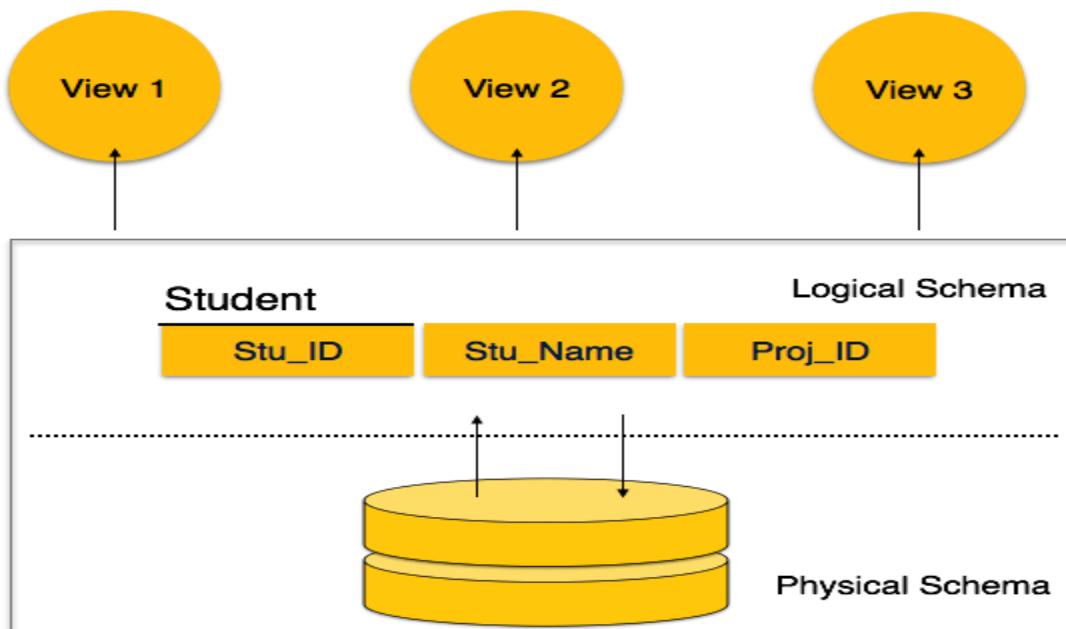


## Database Schema

The overall design of the database is called the database schema.

A database schema is the skeleton structure that represents the logical view of the entire database. 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.

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.



Database systems have several schemas, partitioned according to the levels of abstraction.

- **Physical Database Schema (level)** – The lowest level of abstraction. This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
- **Logical Database Schema (level)** – The next higher level of abstraction. This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints. The logical level thus describes the entire database in terms of a small number of relatively simple structures.
- **View level** - The highest level of abstraction describes only part of the entire database. A database may also have several schemas at the view level, sometimes called subschemas that describe different views of the database.

## Database Instance

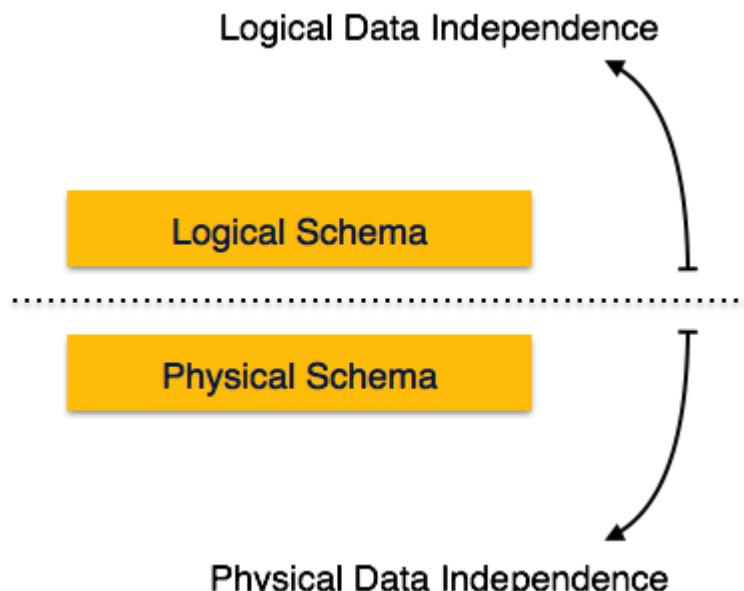
The collection of information stored in the database at a particular moment is called an instance of the database.

It is important that we distinguish these two terms individually. Database schema is the skeleton of database. It is designed when the database doesn't exist at all. Once the database is operational, it is very difficult to make any changes to it. A database schema does not contain any data or information.

A database instance is a state of operational database with data at any given time. It contains a snapshot of the database. Database instances tend to change with time. A DBMS ensures that its every instance (state) is in a valid state, by diligently following all the validations, constraints, and conditions that the database designers have imposed.

## Data Independence

A database system normally contains a lot of data in addition to users' data. For example, it stores data about data, known as metadata, to locate and retrieve data easily. It is rather difficult to modify or update a set of metadata once it is stored in the database. But as a DBMS expands, it needs to change over time to satisfy the requirements of the users. If the entire data is dependent, it would become a tedious and highly complex job.



Metadata itself follows a layered architecture, so that when we change data at one layer, it does not affect the data at another level. This data is independent but mapped to each other.

## Logical Data Independence

Logical data is data about database, that is, it stores information about how data is managed inside. For example, a table (relation) stored in the database and all its constraints, applied on that relation.

Logical data independence is a kind of mechanism, which liberalizes itself from actual data stored on the disk. If we do some changes on table format, it should not change the data residing on the disk.

## Physical Data Independence

All the schemas are logical, and the actual data is stored in bit format on the disk. Physical data independence is the power to change the physical data without impacting the schema or logical data.

For example, in case we want to change or upgrade the storage system itself – suppose we want to replace hard-disks with SSD – it should not have any impact on the logical data or schemas.

## Database Users and User Interfaces

People who work with a database can be categorized as database users or database administrators.

There are four different types of database-system users.

1. Naïve users (End users)- are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously. The typical user interface for end user is a forms interface, where the user can fill in appropriate fields of the form.
2. Application programmers are computer professionals who write application programs. Application programmers can choose from many tools like RAD and fourth generation languages.
3. Sophisticated users interact with the system without writing programs.
4. Specialized users are sophisticated users who write specialized database applications.
5. Database Administrator- A person who has such central control over the system is called a database administrator. The functions of a DBA include:
  - Schema definition.
  - Storage structure and access-method definition.
  - Schema and physical-organization modification.
  - Granting of authorization for data access.
  - Routine maintenance.

**DATABASE MANAGEMENT SYSTEM**