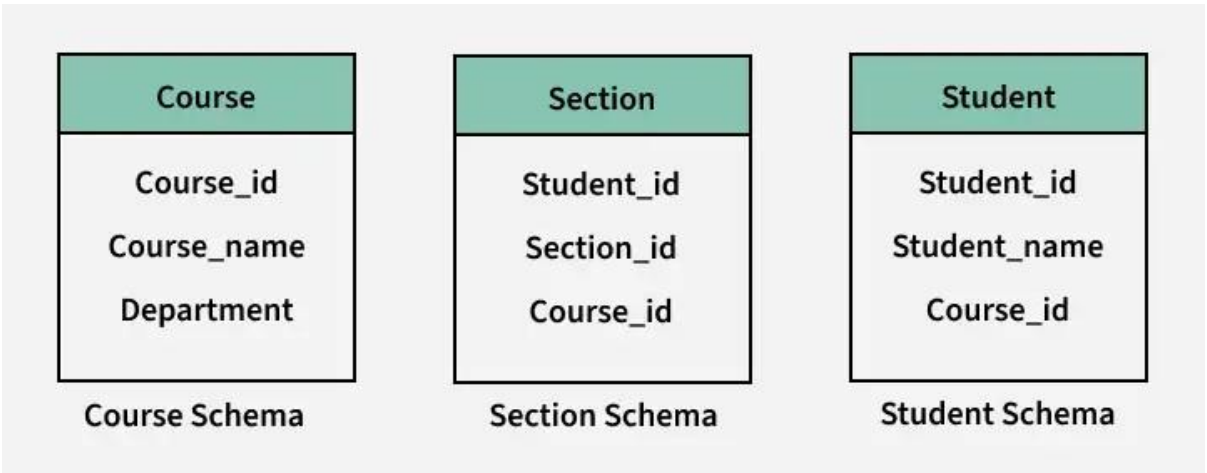
Database Schemas

Last Updated : 13 Jan, 2025

A database schema defines the structure and organization of data within a database. It outlines how data is logically stored, including the relationships between different tables and other database objects. The schema serves as a blueprint for how data is stored, accessed, and manipulated, ensuring consistency and integrity throughout the system. In this article, we will explore the concept of database schema, its types, and how it plays a crucial role in designing efficient and scalable databases

What is Schema?

A schema is the blueprint or structure that defines how data is organized and stored in a database. It outlines the tables, fields, relationships, views, indexes, and other elements within the database. The schema defines the logical view of the entire database and specifies the rules that govern the data, including its types, constraints, and relationships.



Schemas

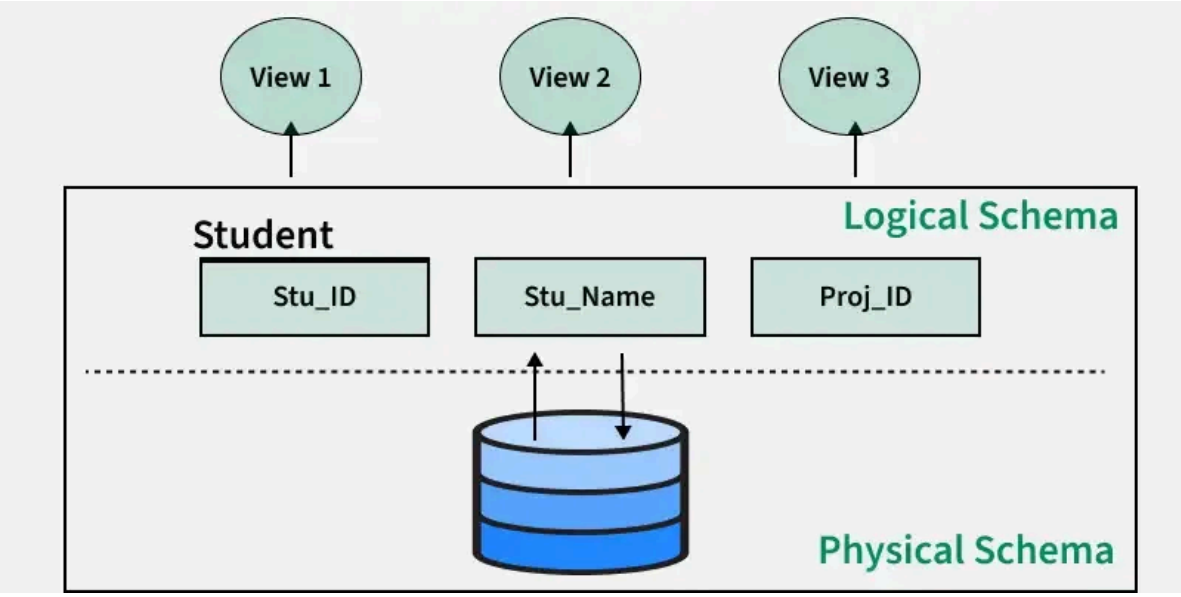
Database Schema

A database schema is the design or structure of a database that defines how data is organized and how different data elements relate to each other. It acts as a blueprint, outlining tables, fields, relationships, and rules that govern the data.

Key points about a database schema:

- It defines how data is logically organized, including tables, fields, and relationships.
- It outlines the relationships between entities, such as primary and foreign keys.
- It helps resolve issues with unstructured data by organizing it in a clear, structured way.
- Database schemas guide how data is accessed, modified, and maintained.

In simple terms, the schema provides the framework that makes it easier to understand, manage, and use data in a database. It's created by database designers to ensure the data is consistent and efficiently organized.



Types of Database Schemas

Types of Database Schemas

Physical Database Schema

- A physical schema defines how data is stored in the storage system, including the arrangement of files, indices and other storage structures. It specifies the actual code and syntax needed to create the database structure. Essentially, it determines where and how the data is stored in the physical storage medium.

- The database administrator decides the storage locations and organization of data within the storage blocks. This schema represents the lowest level of abstraction

Logical Database Schema

- A logical database schema defines the logical structure of the data, including tables, views, relationships, and integrity constraints. It describes how data is organized in tables and how the attributes of these tables are connected. The logical schema ensures that the data is stored in an organized manner, while maintaining data integrity.

- Using Entity-Relationship (ER) modeling, the logical schema outlines the relationships between different data components. It also defines integrity constraints to ensure the quality of data during insertion and updates.

- This schema represents a higher level of abstraction compared to the physical schema, focusing on logical constraints and how the data is structured, without dealing with the physical storage details.

View Database Schema

- The view schema is the highest level of abstraction in a database, focusing on how users interact with the database. It defines the interface through which users can access and manipulate data, without needing to understand the underlying storage mechanisms.

- A database can have multiple view schemas, also known as subschemas, each providing a different perspective of the data. These schemas describe only a part of the database.

Creating Database Schema

For creating a schema, the statement "CREATE SCHEMA" is used in every database. But different databases have different meanings for this. Below we'll be looking at some statements for creating a database schema in different database systems:

1. MySQL: In MySQL, we use the "CREATE SCHEMA" statement for creating the database, because, in MySQL CREATE SCHEMA and CREATE DATABASE, both statements are similar.

2. SQL Server: In SQL Server, we use the "CREATE SCHEMA" statement for creating a new schema.

3. Oracle Database: In Oracle Database, we use "CREATE USER" for creating a new schema, because in the Oracle database, a schema is already created with each database user. The statement "CREATE SCHEMA" does not create a schema, instead, it populates the schema with tables & views and also allows one to access those objects without needing multiple SQL statements for multiple transactions.

Database Schema Designs

There are many ways to structure a database and we should use the best-suited schema design for creating our database because ineffective schema designs are difficult to manage & consume extra memory and resources.

Schema design mostly depends on the application's requirements. Here we have some effective schema designs to create our applications, let’s take a look at the schema designs:

1. Flat Model
2. Hierarchical Model
3. Network Model
4. Relational Model
5. Star Schema
6. Snowflake Schema

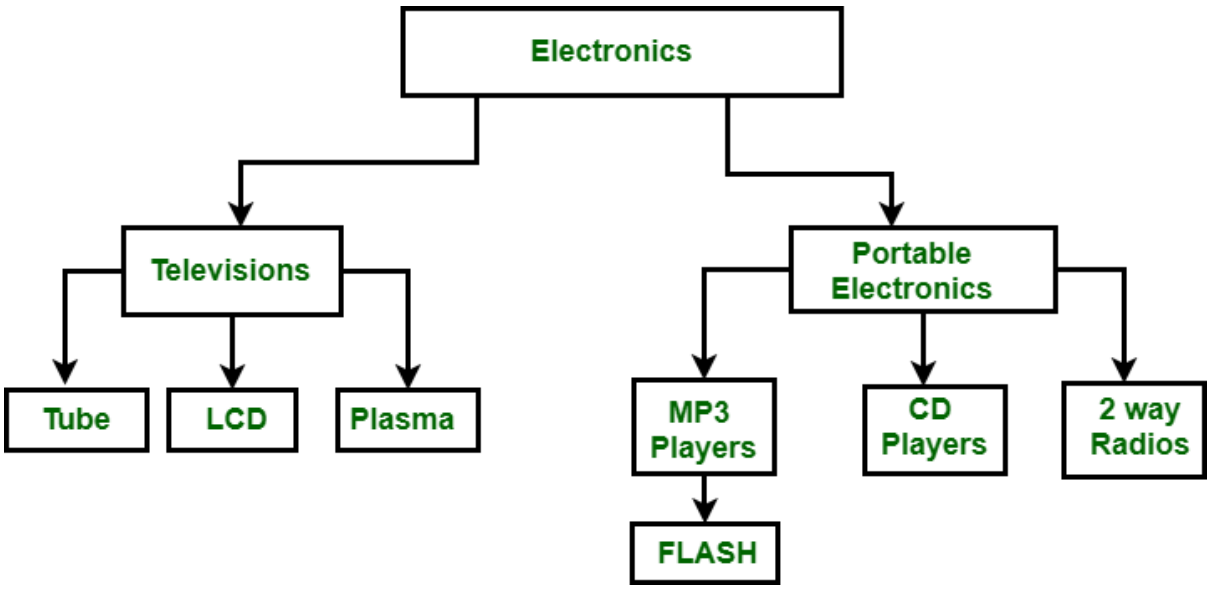
Flat Model

A flat model schema is a 2-D array in which every column contains the same type of data/information and the elements with rows are related to each other. It is just like a table or a spreadsheet. This schema is better for small applications that do not contain complex data.

Flat Model

Hierarchical Model

Data is arranged using parent-child relationships and a tree-like structure in the Hierarchical Database Model. Because each record consists of several children and one parent, it can be used to illustrate one-to-many relationships in diagrams such as organizational charts. A hierarchical database structure is great for storing nested data.

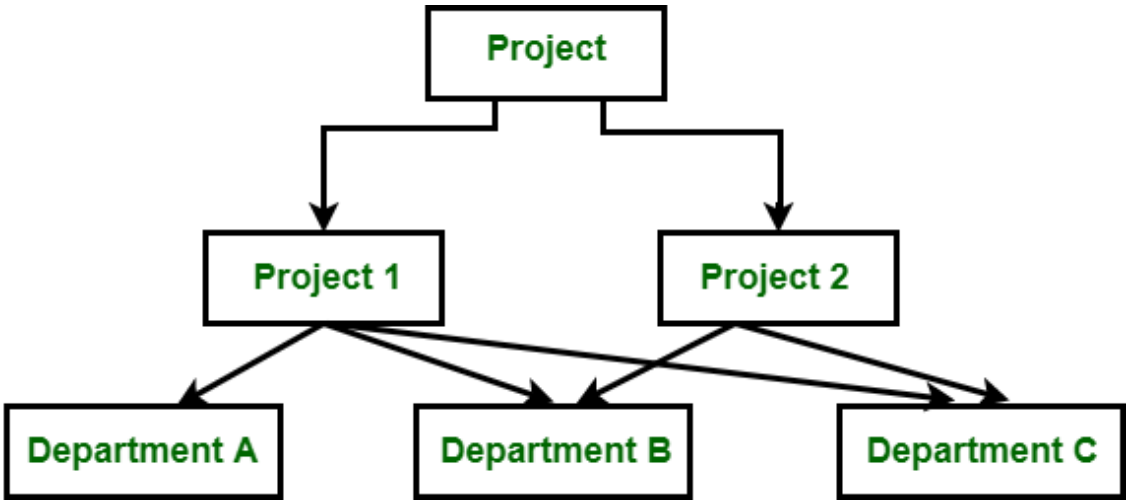


Designing Hierarchical Model

Network Model

The network model is similar to the hierarchical model in that it represents data using nodes (entities) and edges (relationships). However, unlike the hierarchical model, which enforces a strict parent-child relationship, the network model allows for more flexible many-to-many relationships. This flexibility means that a node can have multiple parent nodes and child nodes, making the structure more dynamic.

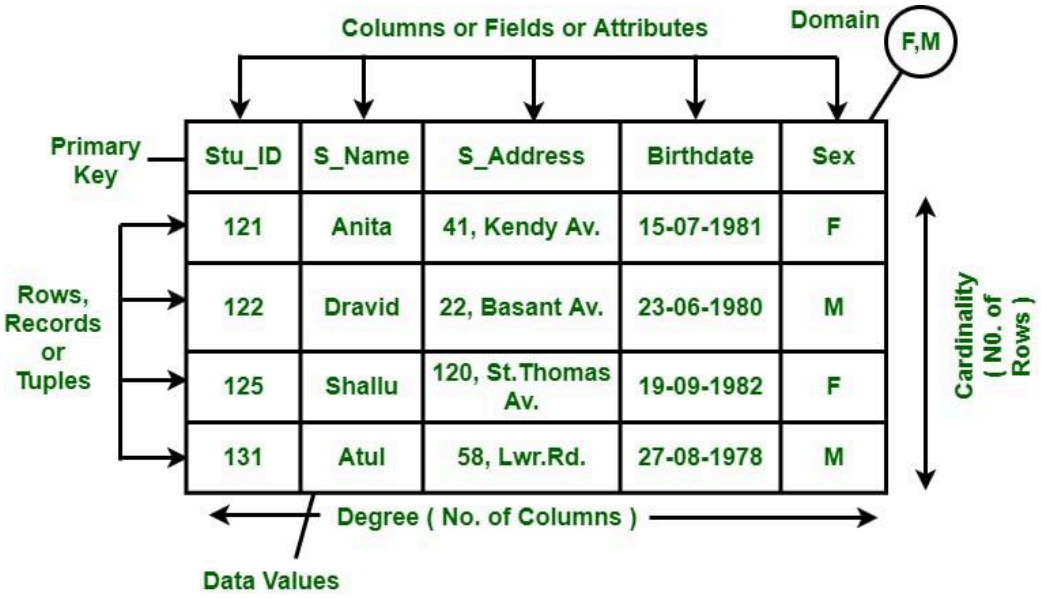
The network model can contain cycles which is a situation where a path exists that allows you to start and end at the same node. These cycles enable more complex relationships and allow for greater data interconnectivity.



Designing Network Model

Relational Model

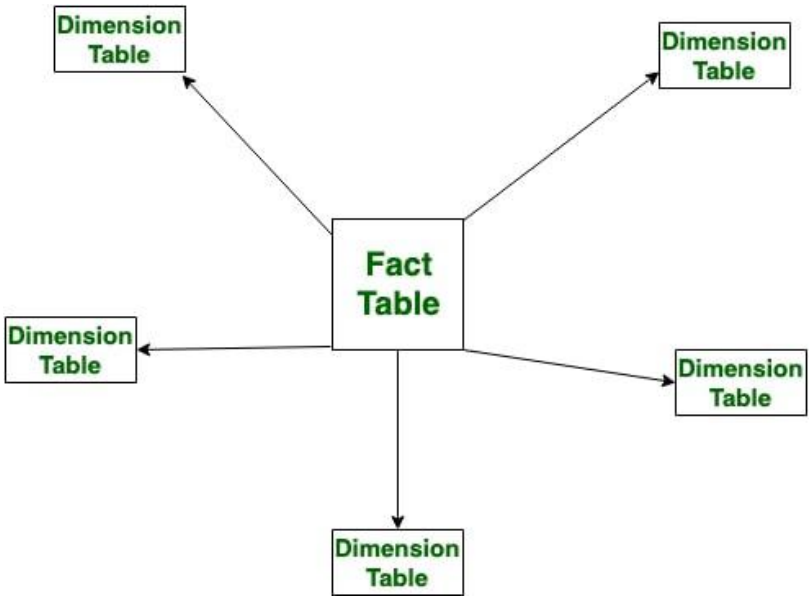
The relational model is mainly used for relational databases, where the data is stored as relations of the table. This relational model schema is better for object-oriented programming.



Designing Relational Model

Star Schema

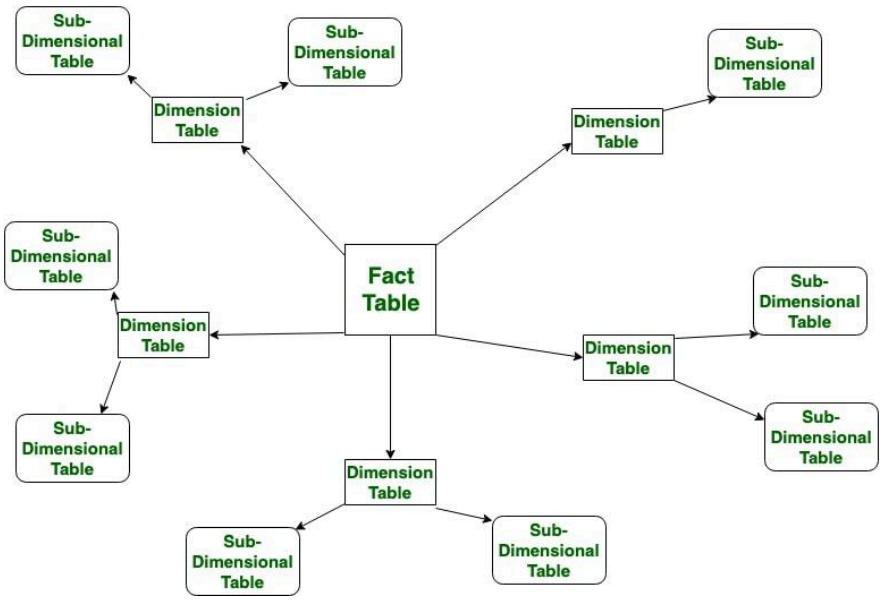
Star schema is better for storing and analyzing large amounts of data. It has a fact table at its center & multiple dimension tables connected to it just like a star, where the fact table contains the numerical data that run business processes and the dimension table contains data related to dimensions such as product, time, people, etc. or we can say, this table contains the description of the fact table. The star schema allows us to structure the data of RDBMS.



Designing Star Schema

Snowflake Schema

Just like star schema, the snowflake schema also has a fact table at its center and multiple dimension tables connected to it, but the main difference in both models is that in snowflake schema – dimension tables are further normalized into multiple related tables. The snowflake schema is used for analyzing large amounts of data.



Designing Snowflake Schema

Difference between Logical and Physical Database Schema

Physical Schema	Logical Schema
Physical schema describes the way of storage of data in the disk.	Logical schema provides the conceptual view that defines the relationship between the data entities.
Having Low level of abstraction.	Having a high level of abstraction.
The design of database is independent to any database management system.	The design of a database must work with a specific database management system or hardware platform.
Changes in Physical schema effects the logical schema	Any changes made in logical schema have minimal effect in the physical schema
Physical schema does not include attributes.	Logical schema includes attributes.

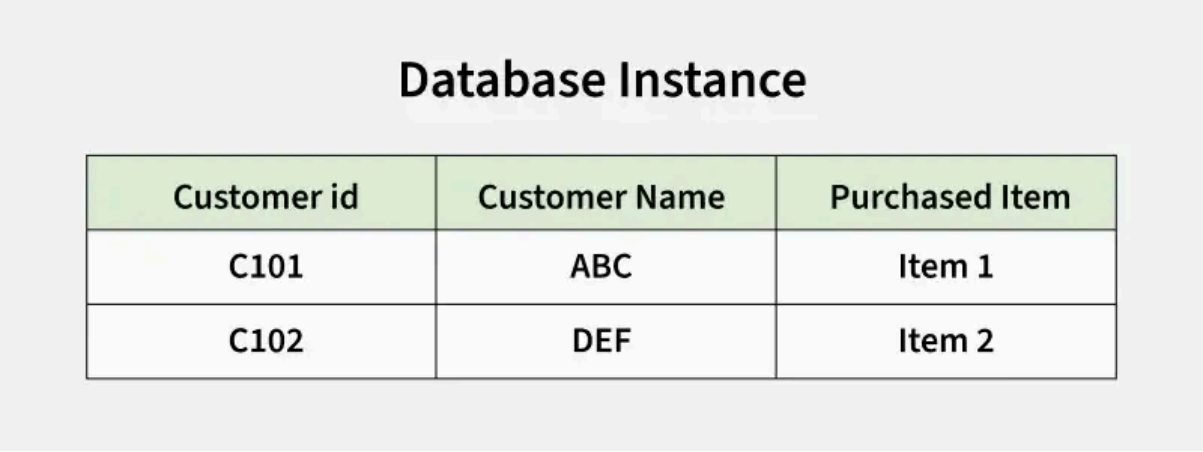
Physical Schema	Logical Schema
Physical schema contains the attributes and their data types.	Logical schema does not contain any attributes or data types.
Examples: Data definition language(DDL), storage structures, indexes.	**Examples:** Entity Relationship diagram, Unified Modeling Language, class diagram.

Advantages of Database Schema

- **Providing Consistency of data:** Database schema ensures the data consistency and prevents the duplicates.
- **Maintaining Scalability:** Well designed database schema helps in maintaining addition of new tables in database along with that it helps in handling large amounts of data in growing tables.
- **Performance Improvement:** Database schema helps in faster data retrieval which is able to reduce operation time on the database tables.
- **Easy Maintenance:** Database schema helps in maintaining the entire database without affecting the rest of the database
- **Security of Data:** Database schema helps in storing the sensitive data and allows only authorized access to the database.

Database Instance

A database instance is a snapshot of a database at a specific moment in time, containing all the properties described by a database schema as data values. Unlike database schemas, which are considered the "blueprint" of a database, instances can change over time whereas it is very difficult to modify the schema because the schema represents the fundamental structure of the database. Database instance does not hold any information related to the saved data in database.



Instance

Database schema versus database instance

Aspect	Database Schema	Database Instance
Definition	Blueprint or design of the database structure	Actual data stored in the database at a given time
Nature	Static (does not change frequently)	Dynamic (changes with every data modification)
Represents	Structure (tables, columns, data types, relationships)	State of the data in the database
Example	Table definitions, data types, constraints	Actual rows of data in the tables
Change Frequency	Changes infrequently (e.g., during schema design changes)	Changes frequently with transactions

Conclusion

In conclusion, **database schemas** play a crucial role in organizing, structuring, and managing data within a database. They define how data is logically and physically stored, ensuring consistency, integrity, and efficient access. Understanding these schemas helps database