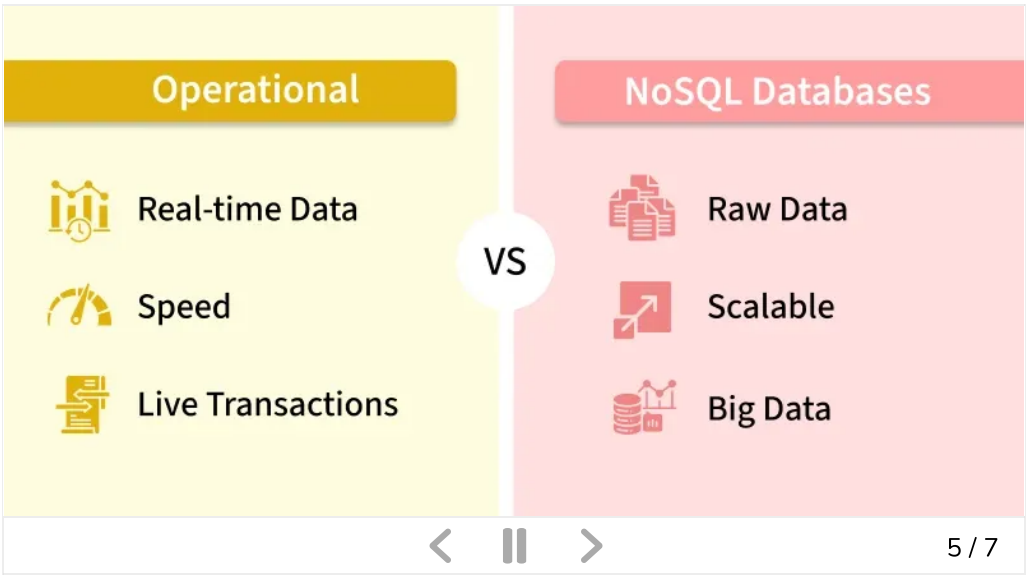
Types of Databases

Last Updated : 07 Aug, 2025

Databases are essential for storing, retrieving, and managing structured or unstructured data efficiently.

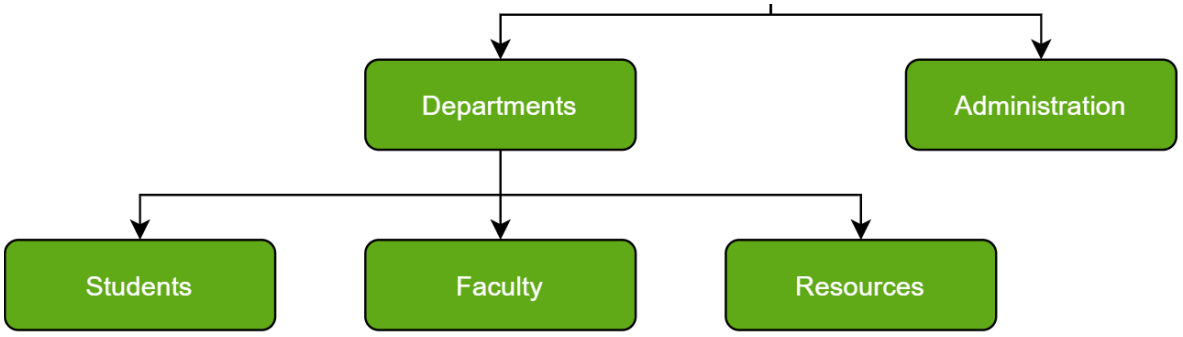
- They form the backbone of modern applications—from mobile apps to enterprise systems.
- The choice of database impacts performance, scalability, consistency, and data integrity.
- Databases are classified based on structure, data model, storage method, and use case.

Understanding the types of databases helps in selecting the right system for specific application needs.



1. Hierarchical Databases

Hierarchical databases organize data in a **tree-like structure,** where each parent record can have multiple child records**.** This model works well for scenarios where data follows a predefined hierarchical relationship, where data is arranged in levels or ranks.



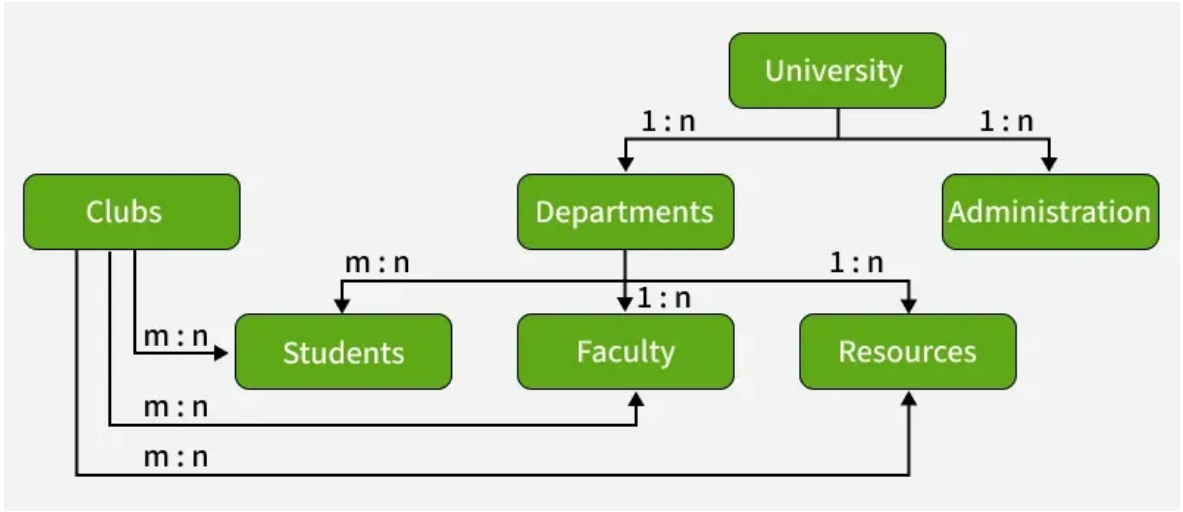
Hierarchical Database Example

For example, in a university, "University" is at the top level, while "Departments" and "Administration" are at lower levels, even though they are distinct entities. This structure can also be viewed as a **parent-child relationship**, where each parent record can have multiple child records, but a child record can only have one parent. As more data are added, the structure expands like a tree.

- **Example**: IBM's Information Management System (IMS) is a well-known hierarchical database.

2. Network Databases

A network databases build on the hierarchical model but allow child records to be linked to multiple parent records, creating a web-like structure of interconnected data. This results in a more flexible structure, often referred to as a graph model, where entities can be connected in many different ways.

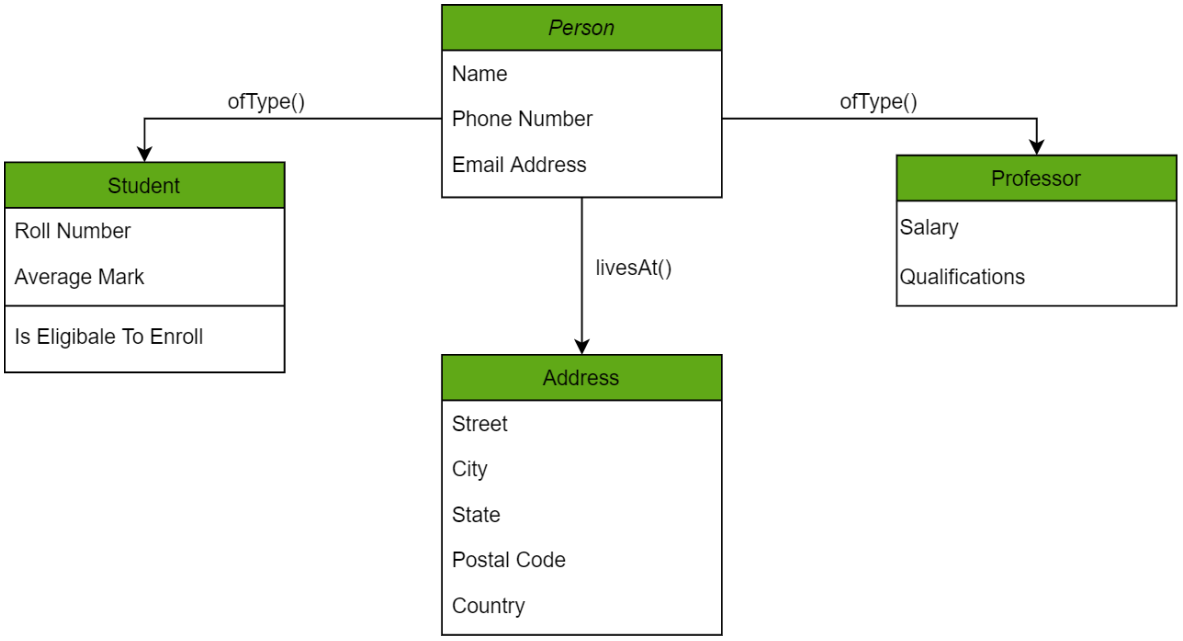


For example: **club<---->students**. In a University database, a student can join multiple clubs and a club can have multiple students. This model is

- **Example**: The **Integrated Data Store (IDS)** is a well-known example of a network database.

3. Object-Oriented Databases

Object-oriented databases are based on the principles of object-oriented programming **(OOP)**, where data is stored as objects. These objects include attributes (data) and methods (functions), making them easily referenced and manipulated. These databases are designed to handle complex data structures such as multimedia, graphics, and large files.



Object-Oriented Example

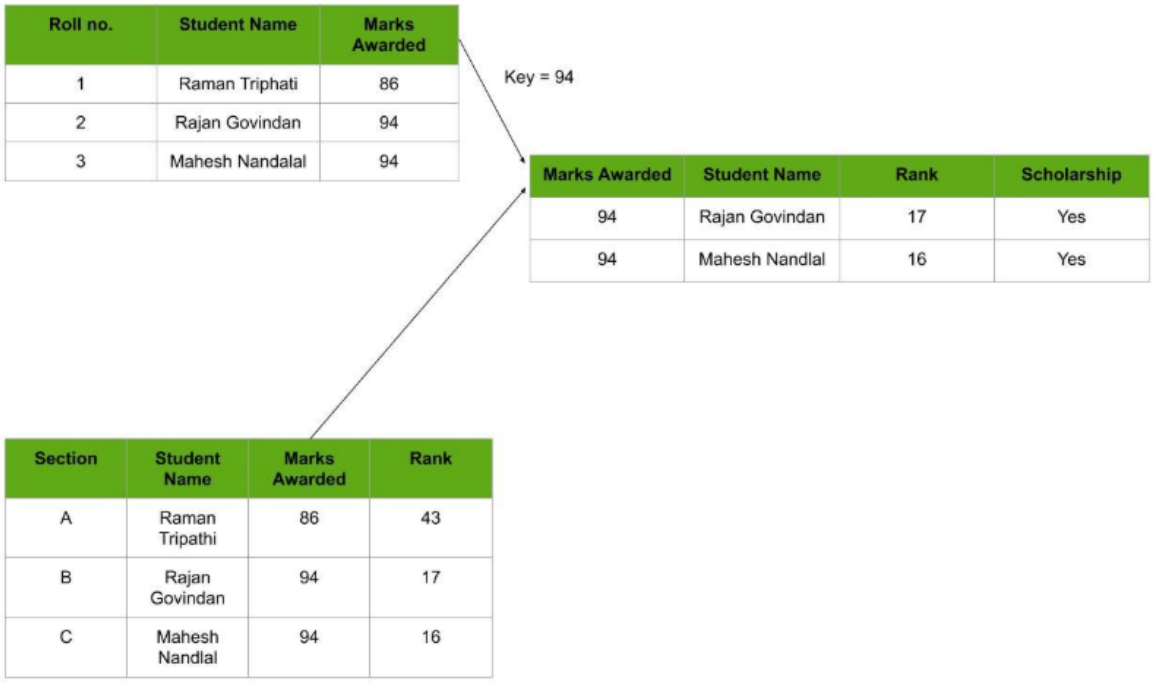
For instance, a "**Person**" object in the database could include attributes like `Name` and `Address` and methods like `getLatestAddress()` to retrieve information. This approach reduces the workload on the database by allowing objects to be reused and linked directly, streamlining data access and **manipulation**. Each object behaves as an instance of thedatabase model, enabling efficient operations.

A practical example of this model is the **Berkeley DB software library**, which is designed for fast and efficient query responses in embedded systems. Object-oriented databases are especially useful for applications involving complex data types or multimedia content.

4. Relational Databases

information has a relationship with every other piece of information. This is on account of every data value in the database having a **unique identity** in the form of a record. Note that all the data are tabulated in this model.

Therefore, every row of data in the database is linked with another row using a primary key. Similarly, every table is linked with another table using a **foreign key**. Refer to the diagram below and notice how the concept of '**Keys**' is used to link two tables.



Relational Database Example

Due to this introduction of tables to organize data, it has become exceedingly popular. In consequence, they are widely integrated into Web-App interfaces to serve as ideal repositories for user data.

What makes it further interesting is the ease in mastering it, since the language used to interact with the database is simple (SQL in this case) and easy to comprehend. In Relational databases, scaling and traversing through data is quite a **lightweight task** in comparison to Hierarchical Databases.

- **Example**: MySQL, PostgreSQL, and Oracle Database are some popular relational databases.

5. Cloud Databases

are widely used for applications requiring dynamic workloads, as they eliminate the need for on-premises infrastructure.

Common cloud services for accessing and managing databases include SaaS **(Software as a Service)** and PaaS **(Platform as a Service)**, which simplify database operations for businesses. Popular cloud platforms offering database services include:

- Amazon Web Services (AWS)
- Google Cloud Platform (GCP)
- Microsoft Azure
- ScienceSoft, etc.

6. Centralized Databases

A centralized database is a database stored and managed at a **single location**, such as a **central server** or data center. It ensures higher security and consistency as all data are maintained in one place, making it easier to control and manage.

Users can access the database remotely to fetch or update information. Centralized databases are commonly used in enterprise systems where data consistency and security are critical. However, scalability and performance limitations should be carefully considered.

7. Personal Databases

A personal database is a **small-scale database** designed for a single user, typically used on personal computers or mobile devices. These databases are ideal for managing individual data like contacts, budgets, notes, or schedules. They are lightweight, easy to use, and require minimal database administration, making them accessible for non-technical users. Examples are:

- **Microsoft Access**: A simple database solution for personal or small business needs.
- **SQLite**: A lightweight, self-contained database commonly used in mobile and desktop applications.

data for daily operations within organizations and businesses. It allows users to create, update, and delete data efficiently, ensuring that the database reflects current activities and transactions.

These databases handle live transactions and provide quick access to up-to-date data. **SAP HANA** is an example of an operational database used for high-speed transactions and analytics.

9. NoSQL Databases

A NoSQL database (short for "non-SQL" or "non-relational") provides a mechanism for storing and retrieving data that does not rely on traditional table-based relational models. Instead, it uses flexible data models like key-value pairs, documents, column families, or graphs, making it ideal for handling unstructured, semi-structured, and structured data.

NoSQL databases are known for their simplicity of design, horizontal scalability (adding more servers for scaling), and high availability. Unlike relational databases, their data structures allow faster operations in certain use cases. MongoDB, for instance, is a widely used document-based NoSQL database.

Real World Application of Database

Databases are used in most modern applications, whether the database is on our personal phone, computer or the internet. An operational database system will store much of the data an application needs to function, keeping the data organized and allowing users to access the data.

application would access and store in the operational database system, such as:

- **Customer data:** like usernames, email addresses, and preferences, and etc.
- **Business data:** like product properties, prices, reviews and ratings, , and etc.
- **Relationship data:** a customer can view multiple products, and vice versa.

Next Article

Introduction of DBMS (Database Management System)

Similar Reads

Corporate & Communications Address:
A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)

Registered Address:
K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305