Chapter 1: Introduction to Databases

Prepared by: Sanskar Debnath

What is a Database?

A database is an **electronically stored, systematic collection of data**. It can contain any type of data, including words, numbers, images, videos, and files. We can use software called a database management system (DBMS) to store, retrieve, and edit data. In computer systems, the word **database** can also refer to any DBMS, to the database system, or to an application associated with the database.

What are the different database types explain them with an example.

There are mainly two types of database are available

- **Relational database**
- **Non relational database**

Relational Database: A relational databases organize data into tables with rows (records) and columns (fields), and establish relationships between these tables. RD or relational database typically use SQL (Structured Query Language) for querying and manipulating data.

Examples:

- MySQL: A popular open-source relational database management system (RDBMS). It's widely used in web applications.
- PostgreSQL: Another powerful open-source RDBMS known for its reliability and extensibility.
- Oracle: A leading commercial RDBMS, often used in large enterprise applications.
- Microsoft SQL Server: A commercial RDBMS from Microsoft, commonly used in Windows-based environments.
- S**QLite:** A lightweight, file-based database that requires no separate server process. It's often used for embedded systems, mobile apps, and small-scale applications.
- **MariaDB:** A fork of MySQL, aiming to maintain compatibility while offering improvements in performance and features. It's a popular choice for web applications and general-purpose database tasks.
- **Amazon Aurora:** A cloud-native relational database service offered by Amazon Web Services (AWS). It's designed for high performance and availability, compatible with MySQL and PostgreSQL.

Non-Relational Databases (NoSQL): Non-relational databases offer more flexibility in data modeling and don't rely on the traditional table-based structure. They are designed to handle various data types and large volumes of data. They can be schema-less or have flexible schemas, accommodating unstructured or semi-structured data. They use different query languages or APIs, often specific to the database type.

Examples:

- **Document Databases:** Store data as documents (like JSON or XML), ideal for content management and web applications. **Example:** MongoDB
- **Key-Value Stores:** Store data as key-value pairs, optimized for fast retrieval, often used for caching and session management. **Example:** Redis
- **Column-Family Stores:** Store data in columns grouped into families, suitable for analytical applications and large datasets. **Example:** Cassandra
- **Graph Databases:** Store data as nodes and edges, ideal for representing relationships and networks, used in social networks and recommendation engines. **Example:** Neo4j