DATABASE

1. What is data?

A:Data is a collection facts that can be form of numbers, words, images etc…

1. What is a database?

A:Database is a systematic collection of data that supports storage and manipulation of that data in easy way.

1. what is DBMS?

A: A Database Management System(DBMS) is software that provides an interface for users and applications to interact with a database .it allows users to store, retrieve, and manage data in database.

There are different kinds of databases stores data in different ways

Types of Databases

**Relational Databases:** Stores data in tabular form.

1. MySQL: An easy-to-use database that organizes data into tables and uses SQL to manage it.
2. PostgreSQL: A powerful database that uses SQL, known for handling complex queries and large datasets.
3. Oracle Database: A robust commercial database designed for large businesses, supporting complex data and high-security needs.
4. Microsoft SQL Server: A database by Microsoft that integrates well with other Microsoft products and supports complex data management.
5. SQLite: A lightweight, self-contained database often used in mobile apps and small projects.

# Non-Relational Databases:

NoSQL databases are designed for handling large volumes of unstructured or semi- structured data. They offer flexibility in data modeling and can scale horizontally. There are several types of NoSQL databases:

# Document Stores

These databases store data as JSON, BSON, or XML documents, making them suitable for handling complex data structures.

Examples: MongoDB, CouchDB.

# Key-Value Stores

Data is stored as key-value pairs, which allows for rapid data retrieval. Examples: Redis, DynamoDB.

# Column Stores

Data is stored in columns rather than rows, optimizing them for read-heavy operations. Examples: Apache Cassandra, HBase.

# Graph Databases

Graph databases use nodes and edges to represent and store data, making them ideal for applications requiring complex relationships.

Examples: Neo4j, Amazon Neptune.

# Object-Oriented Databases

Object-oriented databases store data in the form of objects, similar to object-oriented

programming languages. They are suitable for applications with complex data relationships. Examples:db4o, ObjectDB.

# NewSQL Databases

NewSQL databases aim to provide the scalability of NoSQL systems while maintaining the ACID properties of traditional relational databases.

Examples: Google Spanner, CockroachDB.

# Time-Series Databases

Optimized for handling time-stamped data, time-series databases are commonly used in monitoring, IoT, and financial applications.

Examples: InfluxDB, TimescaleDB.

# Graph Databases

Graph databases store data in graph structures consisting of nodes, edges, and properties. They are ideal for applications that involve complex data relationships, such as social networks and recommendation systems.

Examples: Neo4j, ArangoDB.

# Hierarchical Databases

Hierarchical databases organize data in a tree-like structure. Each child record has only one parent, making it easy to navigate the hierarchy.

Examples: IBM Information Management System (IMS).

# Network Databases

Network databases use a flexible graph structure with multiple parent-child relationships, allowing for more complex data models compared to hierarchical databases.

Examples: Integrated Data Store (IDS), IDMS (Integrated Database Management System).

# In-Memory Databases

In-memory databases store data in the main memory (RAM) rather than on disk, providing faster read and write operations.

Examples: Redis, SAP HANA.

# Embedded Databases

Embedded databases are integrated within applications, providing database services directly without requiring a separate server.

Examples:SQLite, Apache Derby.

# Multimodel Databases

Multimodel databases support multiple data models, such as relational, document, key- value, and graph, within a single database engine.

Examples: ArangoDB, OrientDB.

# Spatial Databases

Spatial databases are optimized for storing and querying spatial data, including geographic information systems (GIS).

Examples: PostGIS (extension of PostgreSQL), Oracle Spatial.

Each type of database has its strengths and is suited to specific applications and data management needs. Choosing the right type of database depends on the requirements of the project, including data complexity, scalability, and the nature of the queries.