2/7/24

## SQL :→

→ SQL is programming language specifically designed for managing and querying relational databases. It allow users to create, read, update & delete data within a database management system. It is used across different db platform such as MySQL, PostgreSQL, SQLite, Microsoft SQL Server, Oracle and others.

a. Key features :→

i. Data Querying :→ SQL allows users to retrieve data from db using SELECT stmts, which can filter, sort, and aggregate db on specified criteria.

ii. Data Manipulation :- SQL provides commands like INSERT, UPDATE, and DELETE to add, modify and remove rows of data in db tables

iii. Schema Modifications :- SQL enables users to create and alter db schemes, defining tables, views, indexes and relationship between tables.

iv. Data Integrity :- SQL supports constraints (such as primary keys, foreign keys, unique constraints) to enforce rules for data integrity & maintain consistency within the db.

v. Transactions :- SQL supports transactions, allowing multiple SQL operations to be grouped together as a single unit. Transaction ensure atomicity, Consistency, Isolation & Durability (ACID Properties).

## What is Database :→

A db is a collection of data which is also called structured data. It can be accessed or store in a computer system. It can be managed through a Database Management system (DBMS). a software used to manage data. Database refers to related data in a structured form.

## Types of Databases :→

### i. Relational Databases (RDBMS) :→

- Relational db organize data into tables with rows and columns.
- They use structured Query Language (SQL) for querying and managing data.
- Examples include MySQL, PostgreSQL, SQLite Microsoft SQL server, Oracle database, etc.
- Suitable for applications where data relationships and consistency are critical, such as transactional systems and business applications.

### ii. NoSQL Databases :→

- NoSQL (Not only SQL) db are designed to handle large volumes of unstructured or semi-structured data.
- They can be schema-less or have flexible schemas.
- Types include document stores (e.g. MongoDB), key-value stores (e.g Redis) column-family stores (e.g Apache Cassandra) & graph db (e.g Neo4j).

- ideal for applications requiring high availability, scalability, and flexibility, such as web applications, big data analytics, and real time applications

iii. Object Oriented Databases :→

- Object Oriented db store data as object, akin to object oriented programming.
- They are used in applications where complex data structures need to be stored and retrived efficiently
- e.g include db4o, ObjectDB, Versant Object Database etc.
- suitable for applications with complex data models & relationships, such as CAD/CAM, multimedia, and scientific applicaations.

iv. Graph Databases :→

- Graph db store data in graph structures with nodes, edges and properties.
- They are optimized for querying complex relations between data entities
- Examples include Neo4j, OrientDB, Amazon Neptune
- Used in applications such as social network, recommendation system, fraud detection & network management.

v. Column-family Stores :→

- Column-family stores organize data into columns instead of rows, optimizing for read & write efficiency.

- They are suitable for application requiring scalable and distributed data storage.
- Example include Apache Cassandra, HBase, ScyllaDB, etc.
- Used in big data analytics, IoT & other application needing high throughput & availability.

vi. Document Stores :-

- Document stores store data in flexible, JSON-like documents.
- They are schema-less or have a flexible schema, allowing for easy storage & retrieval of semi-structured data.
- Example: MongoDB, Couchbase, CouchDB, etc.
- Ideal for content management, real time analytics & applications where flexibility in data structures is required.

## Why we use Database?

- In content of SQL db are used primarily for structured data management and efficient data manipulation. Here are several key reasons why database are essential in SQL :-

i. Data Organization.
ii. Data Integrity
iii. Data Retrieval
iv. Data Manipulation.
v. Concurrency Control
vi. Scalability & Performance.
vii. Security.
viii. Backup & Recovery.

Q. Difference Between db & DBMS :- →

⊙ Database :-

(i) definition :- A database is an organized collection of structured data that is stored electronically in a computer system. It consists of tables, each containing rows of data organized into columns.

(ii) Purpose :- The primary purpose of a db is to store and manage data efficiently. It provides a structured format for organizing data, making it easier to store, retrieve & manipulate information.

(iii) Components :- A db includes: data itself, the schema structure that defines how the data is organized into tables and possibly indexes for faster data retrieval.

(iv) e.g Examples of db include MySQL, PostgreSQL, Oracle DB, Microsoft SQL Server, SQLite.

v. Scope :- A database refers specifically to the structured collection of data itself, organized into tables.

vi. Functionality :- The database stores data.

- A database is the structured collection of data itself. e.g:- MySQL & PostgreSQL.

# DBMS (Database Management System).

i. Definition :- A DBMS is a software system that provides an interface for managing db and interacting with the data stored within them. It is a software application that allows users and applications to access, manipulate & manage database.

ii. Functionality : DBMS provides a set of tools and functionalities for creating, querying, updating and administering db. It includes capabilities for data definition (creating and modifying schemas), data manipulation and data user permissions and control.

iii. Role :- The DBMS acts as an intermediary between the db itself and the users or applications that need to access the data. It handles tasks such as concurrency control, transaction management, data integrity enforcement and performance optimizations.

iv. e.g. MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, SQLite.

v. Scope :→ it refers to the software system that facilitates the creation, management, & interaction with that database.

vi. Functionality :- It provides tools & utilities to manage, manipulate, and access that data. It includes query language (SQL), administ