**What Is SQL?**

SQL is a **standard language** used to interact with **Relational Database Management Systems (RDBMS)**. It allows you to:

* Create and manage databases
* Insert, update, delete, and retrieve data
* Define access permissions
* Build views, procedures, and triggers

SQL is supported by systems like **MySQL, Oracle, SQL Server, PostgreSQL**, and **SQLite**.

**2. SQL Data Types**

SQL supports various data types to store different kinds of data:

|  |  |
| --- | --- |
| **Category** | **Examples** |
| Numeric | INT, FLOAT, DECIMAL |
| Character | CHAR, VARCHAR, TEXT |
| Date/Time | DATE, TIME, DATETIME |
| Boolean | BOOLEAN |
| Binary | BLOB |

**3. SQL Commands**

SQL commands are grouped into five categories:

**a. DDL (Data Definition Language)**

Used to define and modify database structure:

* CREATE: Create database or table
* ALTER: Modify table structure
* DROP: Delete table or database
* TRUNCATE: Remove all rows from a table

**b. DML (Data Manipulation Language)**

Used to manage data within tables:

* INSERT: Add new records
* UPDATE: Modify existing records
* DELETE: Remove records

**c. DQL (Data Query Language)**

Used to query data:

* SELECT: Retrieve data from tables

**d. DCL (Data Control Language)**

Used to control access:

* GRANT: Give user privileges
* REVOKE: Remove privileges

**e. TCL (Transaction Control Language)**

Used to manage transactions:

* COMMIT: Save changes
* ROLLBACK: Undo changes
* SAVEPOINT: Set a point to rollback to

**4. SQL Clauses and Keywords**

|  |  |
| --- | --- |
| **Clause** | **Purpose** |
| WHERE | Filter rows based on condition |
| GROUP BY | Group rows for aggregation |
| HAVING | Filter groups |
| ORDER BY | Sort results |
| LIMIT / TOP | Restrict number of rows |
| JOIN | Combine rows from multiple tables |

**5. SQL Constraints**

Constraints enforce rules on data in tables:

|  |  |
| --- | --- |
| **Constraint** | **Description** |
| NOT NULL | Prevents null values |
| UNIQUE | Ensures all values are unique |
| PRIMARY KEY | Uniquely identifies each row |
| FOREIGN KEY | Links to another table’s primary key |
| CHECK | Validates data with a condition |
| DEFAULT | Sets a default value |

**6. SQL Joins**

Joins combine data from multiple tables:

|  |  |
| --- | --- |
| **Join Type** | **Description** |
| INNER JOIN | Matches rows in both tables |
| LEFT JOIN | All rows from left + matched right |
| RIGHT JOIN | All rows from right + matched left |
| FULL JOIN | All rows from both tables |
| CROSS JOIN | Cartesian product of both tables |

**7. SQL Functions**

SQL includes built-in functions for calculations and data manipulation:

**Aggregate Functions:**

* COUNT(), SUM(), AVG(), MIN(), MAX()

**String Functions:**

* CONCAT(), SUBSTRING(), UPPER(), LOWER()

**Date Functions:**

* NOW(), CURDATE(), DATEDIFF(), DATE\_ADD()

**8. Views, Stored Procedures, and Triggers**

**Views:**

Virtual tables based on SQL queries.

CREATE VIEW active\_users AS

SELECT \* FROM users WHERE status = 'active';

**Stored Procedures:**

Reusable blocks of SQL code.

CREATE PROCEDURE GetUser(IN userId INT)

BEGIN

SELECT \* FROM users WHERE id = userId;

END;

**Triggers:**

Automatic actions based on events.

CREATE TRIGGER log\_insert

AFTER INSERT ON orders

FOR EACH ROW

BEGIN

INSERT INTO logs (action) VALUES ('New order added');

END;

**9. Example SQL Queries**

**Create Table:**

CREATE TABLE Employee (

ID INT PRIMARY KEY,

Name VARCHAR(100),

Salary DECIMAL(10,2)

);

**Insert Data:**

INSERT INTO Employee VALUES (1, 'Ravi', 50000.00);

**Select Data:**

SELECT \* FROM Employee WHERE Salary > 40000;

**Update Data:**

UPDATE Employee SET Salary = 55000 WHERE ID = 1;

**Delete Data:**

DELETE FROM Employee WHERE ID = 1;

**10. SQL Best Practices**

* Use parameterized queries to prevent SQL injection
* Normalize data to reduce redundancy
* Index frequently queried columns
* Use transactions for critical operations
* Avoid SELECT \* in production queries