

DAY -7

SQL

SQL (Structured Query Language) is used to:

- Store data
- Retrieve data
- Update data
- Delete data

A **database** is an organized collection of data.

Example:

- Student records
- Employee details
- Bank transactions
- Product inventory

What is a Table, Row, and Column?

- **Table** → Collection of related data
- **Row (Record)** → One complete entry
- **Column (Field)** → One type of data

Example:

- Table → STUDENTS
- Row → (1, Asha, 20, CSE)
- Column → name

DDL (Data Definition Language)

Used to define database structure.

Command	Use
CREATE	Create table/database
ALTER	Modify table
DROP	Delete table
TRUNCATE	Remove all records

DML (Data Manipulation Language)

Used to manage data inside tables.

Command Use

INSERT	Add data
UPDATE	Modify data
DELETE	Remove data
SELECT	Retrieve data

. DCL (Data Control Language)

Used for permissions.

Command Use

GRANT	Give access
REVOKE	Remove access

TCL (Transaction Control Language)

Used to manage transactions.

Command Use

COMMIT	Save changes
ROLLBACK	Undo changes

Command Use

SAVEPOINT Set checkpoint

Primary Key & Foreign Key

Primary Key

- Uniquely identifies a row
- Cannot be NULL
- Must be unique

Constraints in SQL

Constraints ensure **data integrity**.

Constraint Purpose

NOT NULL No empty values

UNIQUE No duplicates

PRIMARY KEY Unique + Not null

FOREIGN KEY Table relation

CHECK Condition

Aggregate Functions

Used to perform calculations.

Function Use

COUNT() Number of rows

SUM() Total

AVG() Average

MAX() Maximum

MIN() Minimum

Where SQL Is Used

- Web applications

- Banking systems
- E-commerce websites
- Enterprise software
- Data analytics

JWT

What is JWT?

JWT (JSON Web Token) is a compact and secure way to **transfer information between two parties** as a JSON object.

It is mainly used for:

- **Authentication** (who you are)
- **Authorization** (what you are allowed to access)

Traditional login systems use **sessions**, which store user data on the server.
JWT avoids this by storing information **inside the token**.

Benefits:

- Stateless (no server-side storage)
- Scalable
- Secure (digitally signed)
- Works well with REST APIs

How JWT Works (Step-by-Step)

1. User logs in with credentials
2. Server verifies credentials
3. Server generates a JWT
4. JWT is sent to the client
5. Client sends JWT with every request
6. Server verifies the token before responding

Where JWT is Stored (Client Side)

Common storage options:

- HTTP-only cookies (more secure)
- LocalStorage / SessionStorage

JWT vs Session-Based Authentication

JWT	Sessions
Stateless	Stateful
Token stored on client	Session stored on server
Scalable	Less scalable
Used in APIs	Used in traditional apps

Token Expiration & Refresh Tokens

Access Token

- Short lifespan (minutes)

Refresh Token

- Used to generate new access tokens
- Longer lifespan

Improves security