DAY -1

**SQL AND ADVANCE SQL:**

**What Is SQL?**

**SQL (Structured Query Language) is the standard language used to interact with relational databases. It allows users to create, read, update, and delete (CRUD) data efficiently.**

**Why SQL Is Important**

* **Universal Language for Databases: Works across MySQL, PostgreSQL, SQL Server, SQLite, and more.**
* **Data Manipulation: Enables precise control over data storage and retrieval.**
* **Backend Integration: Essential for server-side logic and data-driven applications.**
* **Scalability: Powers everything from small apps to enterprise-level systems.**
* **Data Integrity: Enforces rules and constraints to maintain clean, reliable data.**
* **Core SQL Components**

| **Component** | **Description** |
| --- | --- |
| SELECT | Retrieves data from one or more tables |
| INSERT | Adds new data into a table |
| UPDATE | Modifies existing data |
| DELETE | Removes data from a table |
| CREATE TABLE | Defines a new table structure |
| ALTER TABLE | Modifies an existing table |
| DROP TABLE | Deletes a table permanently |
| JOIN | Combines rows from multiple tables based on related columns |
| WHERE | Filters records based on conditions |
| GROUP BY | Aggregates data based on one or more columns |
| ORDER BY | Sorts results in ascending or descending order |

1.Domain driven design: basically tables,classes and contents based on the domain we gonna built the application eg student information system,crime reporting system

Domain Driven Design:

Examples:

Educational Application

           Student, Teacher, Course, Department, Subject

Banking Application

           Account, Customer, Fund Transfer …

Retail Application

           Product, Order, Cart, Payment

 Excercises in sql refreshment:

**--creating table**

create table students(

student\_id int primary key ,

name varchar(200),

course varchar(200),

join\_date date

);

**--insertion**

insert into students values

(1,"Anbu","data engineering","2025-01-01"),

(2,"Dhanu","QA","2025-02-02"),

(3,"Abu","data analytics","2025-03-03");

**--querying**

select\*from students

select name,course from students

select \* from students where join\_date > "2025-01-01";

select \* from students where course="data engineering" and join\_date>"2025-01-01"

select \* from students where join\_date between "2025-01-01" and "2025-03-03"

select \*from students where course in("data engineering","QA")

-----------------------------------------------------------------

**--another variation of like**

select\*from students where name like 'A%'

select\*from students where name like '%u'

select course from students where course like "%engineering%"

-------------------------------------------------------------------------

**--updating**

update students set course="Advanced data engineering" where student\_id=1;

update students

set join\_date="2025-11-11"

where name ="Dhanu"

select\*from students

**--deleting**

delete from students

where join\_date = "2025-03-03"

delete from students where student\_id=1

* **create,read,update,delete** these are the foundation and mandatory functions where all websites applications uses that we do

**explain query for sales analytics database**

**---variration1:**

select employee\_name,sum(sale\_amount) from sales\_Data

group by employee\_name

having sum(sale\_amount)>(select avg(sale\_amount) from sales\_data)

--interpretation : the employees whose total sales amount greater than avg sales amount of total employees

**-- employees who earn more than average salary**

select\*from employees

where salary>(select avg(Salary)from employees)

**Inline query:**

The query statement inside from is called inline query

Subquery: the query in where clause statement is called subquery.

**Example of Inline query:**

select dept\_avg.department,dept\_avg.avg\_salary from(

select department,avg(salary) as avg\_salary

from employees

group by department)

as dept\_avg

**RANKING:**

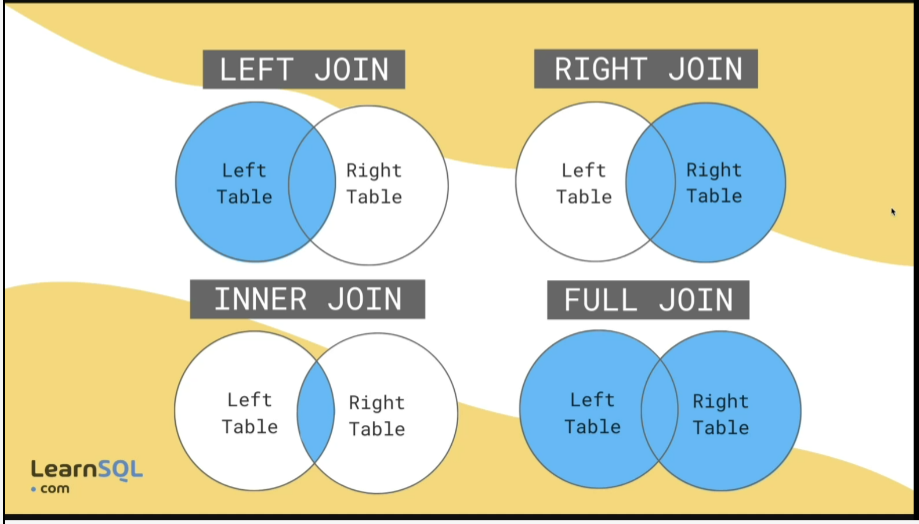
select emp\_name,department,salary,

rank() over (order by salary desc )as salary\_rank

from employees

--we have ranks the top employees based on salary

**JOINS**



**THESE ARE THE TYPES OF JOINS IN SQL**

Examples of inner join:

select customers.customer\_name,o.product\_name,o.order\_amount from orders as o inner join customers using(customer\_id)

we can use all joins by join names except full outer join

for full outer join we need to use two joined tables and union keyword