### Introduction to Joins

Joins are used in SQL to combine rows from two or more tables based on related columns between them. It allows you to retrieve data from multiple tables in a single query.

### Types of Joins

#### Inner Join

An inner join returns only the rows that have matching values in both tables.

SELECT \*

FROM table1

INNER JOIN table2 ON table1.column\_name = table2.column\_name;

#### Left Join (or Left Outer Join)

A left join returns all rows from the left table and the matched rows from the right table. If there's no match, NULL values are returned for the right table columns.

SELECT \*

FROM table1

LEFT JOIN table2 ON table1.column\_name = table2.column\_name;

#### Right Join (or Right Outer Join)

A right join returns all rows from the right table and the matched rows from the left table. If there's no match, NULL values are returned for the left table columns.

SELECT \*

FROM table1

RIGHT JOIN table2 ON table1.column\_name = table2.column\_name;

#### Full Join (or Full Outer Join)

A full join returns all rows when there is a match in either left or right table. If there's no match, NULL values are returned for the unmatched side.

SELECT \*

FROM table1

FULL JOIN table2 ON table1.column\_name = table2.column\_name;

#### Cross Join

A cross join returns the Cartesian product of two tables, resulting in a combination of every row from the first table with every row from the second table.

SELECT \*

FROM table1

CROSS JOIN table2;

#### Natural Join

A natural join performs a join based on all columns in the two tables that have the same name. It automatically matches the columns with the same name.

SELECT \*

FROM table1

NATURAL JOIN table2;

### Union & Union ALL

The UNION operator is used to combine the result sets of two or more SELECT statements into a single result set. It removes duplicate rows.

SELECT column1 FROM table1

UNION

SELECT column1 FROM table2;

The UNION ALL operator is similar to UNION, but it includes all rows, including duplicates.

SELECT column1 FROM table1

UNION ALL

SELECT column1 FROM table2;

### Self Join

A self join is a regular join but the table is joined with itself. It is useful when you want to combine rows from the same table based on a related column.

SELECT e1.employee\_name, e2.employee\_name

FROM employee e1

JOIN employee e2 ON e1.manager\_id = e2.employee\_id;

**-- Joins - inner join , left join , right join and full join**

**-- T1 t2**

**-- stid name age stid tid subject**

**-- 1 abc 12 2 3 maths**

**-- 2 bdc 23 4 3 maths**

**-- 3 ojn 25 5 4 science**

**-- 4 ljn 26**

**-- right join - stid**

**-- stid name age stid tid subject**

**-- 2 bdc 23 2 3 maths**

**-- 4 ljn 26 4 3 maths**

**-- nul nul nul 5 4 science**

**--**

select \* from actor;

select \* from actor\_award;

select \* from actor inner join actor\_award on actor.actor\_id = actor\_award.actor\_id;

**-- Display all the cities with there postal code in India ?**

select \* from country; **-- country data**

select \* from city; **-- city data**

select \* from address; **-- postal codes**

SELECT

postal\_code, city, country

FROM

address

INNER JOIN

city ON address.city\_id = city.city\_id

INNER JOIN

country ON city.country\_id = country.country\_id

WHERE

country = 'India';

**-- Display the names of actors and the names of the films they have acted in.**

select \* from actor; **-- names of actor**

select \* from film\_actor; **-- common connection**

select \* from film; **-- title**

SELECT

concat(a.first\_name, " ", a.last\_name) as Name, title

FROM

actor a

INNER JOIN

film\_actor ON a.actor\_id = film\_actor.actor\_id

INNER JOIN

film ON film.film\_id = film\_actor.film\_id;

**-- Display all the actors and total number of films they have acted in.**

SELECT

concat(first\_name, " ", last\_name) as Name, count(title) as count\_of\_movies

FROM

actor

left JOIN

film\_actor ON actor.actor\_id = film\_actor.actor\_id

left JOIN

film ON film.film\_id = film\_actor.film\_id

group by Name

order by count\_of\_movies;