TABLE JOINS IN MYSQL

SQL joins are essential for combining rows from two or more tables based on a related column, allowing for complex queries that retrieve meaningful data from multiple sources.

INNER JOIN

**Description**: Returns records that have matching values in both tables.

**Syntax** :

**SELECT columns**

**FROM table1**

**INNER JOIN table2**

**ON table1.common\_column = table2.common\_column;**

**Example:**

**SELECT \***

**FROM std\_details**

**INNER JOIN sub\_details**

**ON std\_details.std\_sub = sub\_details.sub\_id;**

**LEFT JOIN (or LEFT OUTER JOIN)**

**Description**: Returns all records from the left table and the matched records from the right table. If there is no match, NULL values are returned for columns from the right table.

Syntax:

**SELECT columns**

**FROM table1**

**LEFT JOIN table2**

**ON table1.common\_column = table2.common\_column;**

Example :

****SELECT \*****

****FROM std\_details****

****LEFT JOIN sub\_details****

****ON std\_details.std\_sub = sub\_details.sub\_id;****

**RIGHT JOIN (or RIGHT OUTER JOIN)**

**Description**: Returns all records from the right table and the matched records from the left table. If there is no match, NULL values are returned for columns from the left table.

Syntax :

**SELECT columns**

**FROM table1**

**RIGHT JOIN table2**

**ON table1.common\_column = table2.common\_column;**

**Example :**

**SELECT \***

**FROM std\_details**

**RIGHT JOIN sub\_details**

**ON std\_details.std\_sub = sub\_details.sub\_id;**

**FULL OUTER JOIN**

**Description**: Combines results of both LEFT and RIGHT joins. It returns all records when there is a match in either left or right table records.

**Syntax**:

**SELECT columns**

**FROM table1**

**FULL OUTER JOIN table2**

**ON table1.common\_column = table2.common\_column;**

**Example :**

**SELECT std\_details.std\_name, sub\_details.sub\_name**

**FROM std\_details**

**LEFT OUTER JOIN sub\_details**

**ON std\_details.std\_sub = sub\_details.sub\_id;**

**SELECT std\_details.std\_name, sub\_details.sub\_name**

**FROM std\_details**

**RIGHT OUTER JOIN sub\_details**

**ON std\_details.std\_sub = sub\_details.sub\_id;**

**CROSS JOIN**

**Description**: Returns the Cartesian product of two tables, meaning every row from the first table is combined with every row from the second.

**Syntax**:

SELECT columns

FROM table1 CROSS JOIN table2;

Example :

**SELECT \***

**FROM std\_details**

**CROSS JOIN std\_fee;**

**SELF JOIN**

**Description**: A regular join but the table is joined with itself.

**Syntax**:

**SELECT A.column\_name, B.column\_name**

**FROM table A, table B**

**WHERE condition;**

Example :

**SELECT A.std\_name, B.std\_name**

**FROM std\_details A, std\_details B**

**WHERE A.id = B.id;**