**🔹 JOINS in MySQL**

* A **JOIN** is used in MySQL to combine rows from two or more tables based on a related column between them.
* It helps to fetch data that is spread across multiple tables.

**🔹 Types of JOINS in MySQL**

**1. INNER JOIN**

* Returns only the rows that have matching values in both tables.

**Syntax:**

SELECT columns

FROM table1

INNER JOIN table2

ON table1.column = table2.column;

**Example:**

SELECT s.student\_id, s.name, e.course

FROM students s

INNER JOIN enrollments e

ON s.student\_id = e.student\_id;

👉 Shows only students who are enrolled in courses.

**2. LEFT JOIN (or LEFT OUTER JOIN)**

* Returns all rows from the **left table**, and matched rows from the right table.
* If no match exists, NULL values are returned for right table columns.

**Syntax:**

SELECT columns

FROM table1

LEFT JOIN table2

ON table1.column = table2.column;

**Example:**

SELECT s.student\_id, s.name, e.course

FROM students s

LEFT JOIN enrollments e

ON s.student\_id = e.student\_id;

👉 Shows all students, even those not enrolled in any course.

**3. RIGHT JOIN (or RIGHT OUTER JOIN)**

* Returns all rows from the **right table**, and matched rows from the left table.
* If no match exists, NULL values are returned for left table columns.

**Syntax:**

SELECT columns

FROM table1

RIGHT JOIN table2

ON table1.column = table2.column;

**Example:**

SELECT s.student\_id, s.name, e.course

FROM students s

RIGHT JOIN enrollments e

ON s.student\_id = e.student\_id;

👉 Shows all enrollments, even if some do not have a valid student record.

**4. FULL JOIN (FULL OUTER JOIN)**

* Returns all rows when there is a match in **either left or right** table.
* MySQL doesn’t support FULL JOIN directly, but you can simulate it using UNION.

**Syntax (using UNION):**

SELECT columns

FROM table1

LEFT JOIN table2

ON table1.column = table2.column

UNION

SELECT columns

FROM table1

RIGHT JOIN table2

ON table1.column = table2.column;

**5. CROSS JOIN**

* Produces the **Cartesian product** of both tables.
* Every row from table1 is combined with every row from table2.

**Syntax:**

SELECT columns

FROM table1

CROSS JOIN table2;

**Example:**

SELECT s.name, c.course\_name

FROM students s

CROSS JOIN courses c;

👉 Every student with every course.

**6. SELF JOIN**

* A table is joined with itself.
* Useful for comparing rows within the same table.

**Syntax:**

SELECT a.column, b.column

FROM table a, table b

WHERE condition;

**Example:**

SELECT e1.employee\_id, e1.name, e2.name AS manager

FROM employees e1

INNER JOIN employees e2

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

👉 Shows employees along with their managers.

**🔹 Quick Summary**

| **Join Type** | **Returns** |
| --- | --- |
| **INNER JOIN** | Only matching rows |
| **LEFT JOIN** | All from left + matching from right |
| **RIGHT JOIN** | All from right + matching from left |
| **FULL JOIN** | All rows from both tables (matched + unmatched) |
| **CROSS JOIN** | Cartesian product |
| **SELF JOIN** | Join table with itself |