In MySQL, **joining tables** is used to combine data from multiple tables based on a related column. Here’s a guide to the different types of joins:

**1. INNER JOIN**

Returns rows where there is a match in both tables.

**Syntax:**

sql

Copy code

SELECT columns

FROM table1

INNER JOIN table2

ON table1.column = table2.column;

**Example:**

sql

Copy code

SELECT employees.name, departments.department\_name

FROM employees

INNER JOIN departments

ON employees.department\_id = departments.id;

This fetches employees and their corresponding department names where there is a match.

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

Returns all rows from the left table and the matching rows from the right table. If no match, NULL values are returned.

**Syntax:**

sql

Copy code

SELECT columns

FROM table1

LEFT JOIN table2

ON table1.column = table2.column;

**Example:**

sql

Copy code

SELECT employees.name, departments.department\_name

FROM employees

LEFT JOIN departments

ON employees.department\_id = departments.id;

This retrieves all employees and their departments. If an employee isn’t assigned a department, NULL is shown for the department.

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

Returns all rows from the right table and the matching rows from the left table. If no match, NULL values are returned.

**Syntax:**

sql

Copy code

SELECT columns

FROM table1

RIGHT JOIN table2

ON table1.column = table2.column;

**Example:**

sql

Copy code

SELECT employees.name, departments.department\_name

FROM employees

RIGHT JOIN departments

ON employees.department\_id = departments.id;

This retrieves all departments and their employees. If a department has no employees, NULL is shown for the employee.

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

Returns all rows when there is a match in either table. If there’s no match, NULL is returned for missing matches.

⚠️ **Note:** MySQL does not support FULL JOIN directly. You can simulate it using UNION.

**Syntax:**

sql

Copy code

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**

Returns the Cartesian product of the two tables (every row of table1 paired with every row of table2).

**Syntax:**

sql

Copy code

SELECT columns

FROM table1

CROSS JOIN table2;

**Example:**

sql

Copy code

SELECT employees.name, projects.project\_name

FROM employees

CROSS JOIN projects;

This pairs every employee with every project.

**Use Case Example**

Imagine these tables:

**employees table:**

| **id** | **name** | **department\_id** |
| --- | --- | --- |
| 1 | Alice | 1 |
| 2 | Bob | NULL |
| 3 | Charlie | 2 |

**departments table:**

| **id** | **department\_name** |
| --- | --- |
| 1 | HR |
| 2 | IT |
| 3 | Finance |

* **INNER JOIN**:

sql

Copy code

SELECT employees.name, departments.department\_name

FROM employees

INNER JOIN departments

ON employees.department\_id = departments.id;

**Result:**

| **name** | **department\_name** |
| --- | --- |
| Alice | HR |
| Charlie | IT |

* **LEFT JOIN**:

sql

Copy code

SELECT employees.name, departments.department\_name

FROM employees

LEFT JOIN departments

ON employees.department\_id = departments.id;

**Result:**

| **name** | **department\_name** |
| --- | --- |
| Alice | HR |
| Bob | NULL |
| Charlie | IT |