SQL joins are powerful tools that allow you to combine rows from two or more tables based on a related column. Here's a breakdown of the **main types of SQL joins** with examples to help you visualize how they work:

🔗 1. INNER JOIN

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

**Example:**

SELECT employees.name, departments.dept\_name FROM employees INNER JOIN departments ON employees.dept\_id = departments.id;

**Result:** Only employees who belong to a department will be shown.

🧭 2. LEFT JOIN (or LEFT OUTER JOIN)

Returns all rows from the left table, and matched rows from the right table. If there's no match, NULLs are returned.

**Example:**

SELECT employees.name, departments.dept\_name FROM employees LEFT JOIN departments ON employees.dept\_id = departments.id;

**Result:** All employees are shown, even if they don't belong to a department.

🪞 3. RIGHT JOIN (or RIGHT OUTER JOIN)

Returns all rows from the right table, and matched rows from the left table. If there's no match, NULLs are returned.

**Example:**

SELECT employees.name, departments.dept\_name FROM employees RIGHT JOIN departments ON employees.dept\_id = departments.id;

**Result:** All departments are shown, even if no employees are assigned to them.

🌐 4. FULL JOIN (or FULL OUTER JOIN)

Returns all rows when there is a match in either left or right table. If there's no match, NULLs are returned on the side without a match.

**Example:**

SELECT employees.name, departments.dept\_name FROM employees FULL OUTER JOIN departments ON employees.dept\_id = departments.id;

**Result:** Combines LEFT and RIGHT JOIN results. Shows all employees and all departments, matched or not.

🔍 5. CROSS JOIN

Returns the Cartesian product of both tables — every combination of rows.

**Example:**

SELECT employees.name, departments.dept\_name FROM employees CROSS JOIN departments;

**Result:** Every employee is paired with every department.

🧩 6. SELF JOIN

A table joins with itself to compare rows within the same table.

**Example:**

SELECT A.name AS Employee, B.name AS Manager FROM employees A JOIN employees B ON A.manager\_id = B.id;

**Result:** Shows each employee along with their manager.

Would you like to try writing a query with one of these joins? Or maybe see a visual diagram to make it clearer?

SELECT

s.user\_id,

ROUND(

COALESCE(SUM(CASE WHEN c.action = 'confirmed' THEN 1 ELSE 0 END), 0) \* 1.0 /

COALESCE(COUNT(c.action), 0),

2

) AS confirmation\_rate

FROM Signups s

LEFT JOIN Confirmations c ON s.user\_id = c.user\_id

GROUP BY s.user\_id;