

SQL Joins Explanation



so, i have this sql joins tomorrow. teach me

Perfect — let's go step by step, very clear and practical, so you're fully ready for tomorrow's SQL joins.

First — What is a JOIN?

JOIN is used when you want to combine data from two or more tables based on a related column between them.

Assume 2 Tables (Example)

Table 1: Employees


emp_id	emp_name	dept_id
1	John	101
2	Alice	102
3	Bob	103
4	Sam	102

Table 2: Departments

dept_id	dept_name
101	HR
102	IT
104	Marketing

Now let's see each JOIN one by one:

1 INNER JOIN

 Returns only the rows where there is a match in both tables.

```
sql
```

```
SELECT Employees.emp_id, Employees.emp_name, Departments.dept_name
FROM Employees
INNER JOIN Departments
ON Employees.dept_id = Departments.dept_id;
```

Output:

emp_id	emp_name	dept_name
1	John	HR
2	Alice	IT
4	Sam	IT

❌ Bob (dept_id 103) and Marketing (dept_id 104) are not shown because they have no matching rows.

2 LEFT JOIN (LEFT OUTER JOIN)

👉 Returns all rows from the LEFT table, even if there is no match in the RIGHT table. Missing matches will show NULL.

```
sql

SELECT Employees.emp_id, Employees.emp_name, Departments.dept_name
FROM Employees
LEFT JOIN Departments
ON Employees.dept_id = Departments.dept_id;
```

Output:

emp_id	emp_name	dept_name
1	John	HR
2	Alice	IT
3	Bob	NULL
4	Sam	IT

Bob is included but no matching department, so NULL.

3 RIGHT JOIN (RIGHT OUTER JOIN)

👉 Opposite of LEFT JOIN. Returns all rows from the RIGHT table, even if there's no match in the LEFT.

```
sql

SELECT Employees.emp_id, Employees.emp_name, Departments.dept_name
FROM Employees
RIGHT JOIN Departments
ON Employees.dept_id = Departments.dept_id;
```

Output:

emp_id	emp_name	dept_name
1	John	HR
2	Alice	IT
4	Sam	IT
NULL	NULL	Marketing

Marketing is shown even though no employees are in that department.

4 FULL JOIN (FULL OUTER JOIN)

👉 Returns all rows from both tables. If no match, returns NULL.

```
sql

SELECT Employees.emp_id, Employees.emp_name, Departments.dept_name
FROM Employees
FULL OUTER JOIN Departments
ON Employees.dept_id = Departments.dept_id;
```

Output:

emp_id	emp_name	dept_name
1	John	HR
2	Alice	IT
3	Bob	NULL
4	Sam	IT
NULL	NULL	Marketing

5 CROSS JOIN

👉 Returns all possible combinations (Cartesian product).

```
sql

SELECT Employees.emp_id, Employees.emp_name, Departments.dept_name
FROM Employees
CROSS JOIN Departments;
```

Output:

Total rows = Employees rows × Departments rows (4 × 3 = 12 rows)

🎯 Quick Summary Table:

JOIN Type	What it returns
INNER JOIN	Only matching rows
LEFT JOIN	All from LEFT + matches from RIGHT
RIGHT JOIN	All from RIGHT + matches from LEFT
FULL JOIN	All from both + NULL where no match
CROSS JOIN	All combinations (cartesian)

🎯 Small Tip for Exam:

- If they give you two tables with missing data, just draw small table diagrams — easy to visualize joins.

Shall we now do 5-10 quick practice questions?

👉 If you say yes, I'll make you fully confident in less than 20 mins.



explain cross join

Excellent — let's focus fully on **CROSS JOIN**:

CROSS JOIN Simple Definition

CROSS JOIN returns **all possible combinations** of rows from both tables.

It **does not** need any condition (no ON clause).

Example

Let's say:

Table 1: Employees

emp_id	emp_name
1	John
2	Alice

Table 2: Departments

dept_id	dept_name
101	HR
102	IT

♦ Query:

```
sql

SELECT Employees.emp_name, Departments.dept_name
FROM Employees
CROSS JOIN Departments;
```

♦ Output:

emp_name	dept_name
John	HR
John	IT

emp_name	dept_name
Alice	HR
Alice	IT

Key Point

- 2 Employees × 2 Departments = 4 rows
- No condition → every row from Employees is paired with every row from Departments.

When CROSS JOIN is used?

- For generating combinations.
- Eg:
 - You want all product & color combinations.
 - You want all student & exam combinations.

Formula for total rows:

sql

Total Rows = Rows in Table 1 × Rows in Table 2

✓ **CROSS JOIN is very rare in real-world unless you're building combinations.**

Shall I give you 2-3 small real exam-type CROSS JOIN questions to practice?

👉 You'll master it quickly. Shall I?