PostgreSQL JOINS – Complete Structured Notes

1. What is a JOIN?

A **JOIN** is used to combine rows from **two or more tables** based on a related column between them.

2. Sample Tables Used in Examples

students table

```
CREATE TABLE students (
  c_id INT
);

INSERT INTO students VALUES
(1),(2),(3),(4);
```

courses table

```
CREATE TABLE courses (
  course_id INT
);

INSERT INTO courses VALUES
(1),(2),(3),(5);
```

Data in tables

students

c_id
1
2
3
4

courses

course_id
1
2
3
5

---

3. JOIN Syntax (General Form)

```
SELECT <columns from left and/or right table>
FROM <left_table>
<JOIN TYPE> JOIN <right_table>
ON <join_condition>;
```

---

4. INNER JOIN

Meaning

Returns **only matching rows** from both tables.

Diagram

```
Students      Courses
    1  ─────▶    1
    2  ─────▶    2
    3  ─────▶    3
```

Query

```
SELECT s.c_id, c.course_id
FROM students s
INNER JOIN courses c
ON s.c_id = c.course_id;
```

Output

c_id	course_id
1	1
2	2

c_id	course_id
3	3

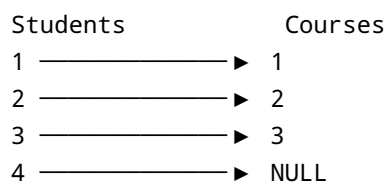## 5. LEFT JOIN (LEFT OUTER JOIN)

Meaning

Returns **all rows from left table** and matching rows from right table. If no match → NULL on right side.

Diagram

```
Students         Courses
1 ───────────► 1
2 ───────────► 2
3 ───────────► 3
4 ───────────► NULL
```

Query

```sql
SELECT s.c_id, c.course_id
FROM students s
LEFT JOIN courses c
ON s.c_id = c.course_id;
```

Output

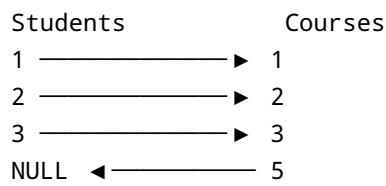| c_id	course_id
1	1
2	2
3	3
4	NULL

---

6. RIGHT JOIN (RIGHT OUTER JOIN)

Meaning

Returns **all rows from right table** and matching rows from left table. If no match → NULL on left side.

Diagram

```
Students         Courses
1 ───────────►  1
2 ───────────►  2
3 ───────────►  3
NULL  ◄──────── 5
```

Query

```sql
SELECT s.c_id, c.course_id
FROM students s
RIGHT JOIN courses c
ON s.c_id = c.course_id;
```

Output

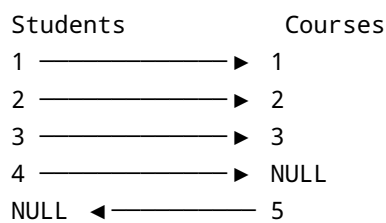| c_id	course_id
1	1
2	2
3	3
NULL	5

---

7. FULL JOIN (FULL OUTER JOIN)

Meaning

Returns **all rows from both tables**. Non-matching rows contain NULL values.

Diagram

```
Students         Courses
1 ───────────►  1
2 ───────────►  2
3 ───────────►  3
4 ───────────►  NULL
NULL  ◄──────── 5
```

Query

```sql
SELECT s.c_id, c.course_id
FROM students s
```

```
FULL JOIN courses c
ON s.c_id = c.course_id;
```

Output

c_id	course_id
1	1
2	2
3	3
4	NULL
NULL	5

---

8. CROSS JOIN

Meaning

Returns **Cartesian product** (every row from left table with every row from right table).

Formula

```
Total rows = rows_in_students × rows_in_courses
           = 4 × 4 = 16
```

Query

```
SELECT s.c_id, c.course_id
FROM students s
CROSS JOIN courses c;
```

Sample Output

c_id	course_id
1	1
1	2
1	3
…	…
4	5

---

9. Quick Comparison Table

JOIN Type	Matches Only	Unmatched Left	Unmatched Right
INNER	Yes	No	No
LEFT	Yes	Yes	No
RIGHT	Yes	No	Yes
FULL	Yes	Yes	Yes
CROSS	No condition	All	All

10. Exam One-Line Definitions

- **INNER JOIN**: Returns only matching rows from both tables.
- **LEFT JOIN**: Returns all rows from left table and matching rows from right.
- **RIGHT JOIN**: Returns all rows from right table and matching rows from left.
- **FULL JOIN**: Returns all rows from both tables.
- **CROSS JOIN**: Returns Cartesian product of two tables.

✅These notes are **Canva-ready**, **exam-oriented**, and **interview-friendly**.

If you want: - ER-style Venn diagrams - Marathi explanation - Interview questions on JOINs

Tell me ♀