----------LECTURE- 8----------

🌟 What is a JOIN (simple definition)

A JOIN is used to combine rows from two tables based on a related column.
Example:

- **A table has customer details**
- **Another table has customer orders**
  JOIN connects them using customer_id.

🚀 1. CROSS JOIN (also called Cartesian Join)

✓ Meaning
Every row from table A joins with every row from table B.

✓ Formula
If A has 5 rows and B has 4 rows → result = 20 rows

✓ Use Case
Rare in real life.
Used for combinations or testing.

✓ Example

```
SELECT *
FROM A
CROSS JOIN B;
```

✓ Key Point
❗ No ON condition needed
Because every row pairs with every row.

🚀 2. INNER JOIN (most common)

✓ Meaning
Only the matching rows from both tables are shown.

If A.customer_id = B.customer_id

Only those rows appear.

✓ Use Case
When you want "only valid connections"

Example: show customers who have orders.

✓ Example

```
SELECT *
FROM customers c
INNER JOIN orders o
ON c.customer_id = o.customer_id;
```

✓ Key Point

❗ Rows that don't match → are removed.

We can also show as many Alias columns with +, -, *, divide conditions within inner joins

We can use INNER JOIN With 3 tables also as below

```
SELECT *
FROM customers c
INNER JOIN orders o
ON c.customer_id = o.customer_id;
INNER JOIN Product p
ON c.pid = p.pid
```

🚀 3. LEFT JOIN (LEFT OUTER JOIN)

✓ Meaning

Get all rows from LEFT table

- matching rows from RIGHT
- non-matching RIGHT rows become NULL

✓ Use Case

Show all customers, even those with no orders.

✓ Example

```
SELECT *
FROM customers c
LEFT JOIN orders o
ON c.customer_id = o.customer_id;
```

✓ Key Point

Left table = complete

Right table = matching + NULL where no match

🚀 4. RIGHT JOIN (RIGHT OUTER JOIN)

✓ Meaning

Get all rows from RIGHT table

- matching rows from LEFT
- non-matching LEFT rows become NULL
-

✓ Use Case

Show all orders, even if the customer record is missing.

✓ Example

```
SELECT *
FROM customers c
RIGHT JOIN orders o
ON c.customer_id = o.customer_id;
```

✓ Key Point

Right table = complete
Left table = matching + NULL where no match

🚀 5. FULL JOIN (FULL OUTER JOIN)

✓ Meaning

Show all rows from both tables:

- All matched rows
- All unmatched rows from LEFT
- All unmatched rows from RIGHT

✓ Use Case

Show everything from both tables.

✓ Example (not in MySQL directly)

```
SELECT *
FROM A
FULL OUTER JOIN B
ON A.id = B.id;
```

✓ MySQL Trick (since FULL JOIN doesn't exist)

Use LEFT JOIN + RIGHT JOIN + UNION:

```
SELECT * FROM A LEFT JOIN B ON A.id = B.id
UNION
SELECT * FROM A RIGHT JOIN B ON A.id = B.id;
```

🚀 6. SELF JOIN

✓ Meaning

A table joins with itself.

✓ Use Case

Used for hierarchical data:

- Employee ↔ Manager
- Category ↔ Subcategory

✓ Example

```
SELECT e.name AS employee,
       m.name AS manager
FROM employees e
LEFT JOIN employees m
ON e.manager_id = m.emp_id;
```

✓ Key Point

Use table alias (E and M) to avoid confusion.

☆ QUICK SUMMARY (Interview Ready)

JOIN TYPE	What You Get
CROSS JOIN	All combinations (A × B)
INNER JOIN	Only matching rows
LEFT JOIN	All rows from LEFT + matching from RIGHT
RIGHT JOIN	All rows from RIGHT + matching from LEFT
FULL JOIN	All rows from both sides (matched + unmatched)
SELF JOIN	Table joined with itself

☆ EASY MEMORY TRICK

INNER → Only matches

LEFT → Left is king

RIGHT → Right is king

FULL → Everyone invited

CROSS → Everyone marries everyone

SELF → Table looks into mirror