**SQL JOIN types**

## INNER JOIN

Returns only matching rows from both tables.

### Example: Show all customer orders with product details

SELECT o.order\_id, c.name AS customer\_name, p.name AS product\_name, od.quantity

FROM orders o

JOIN customers c ON o.customer\_id = c.customer\_id

JOIN order\_details od ON o.order\_id = od.order\_id

JOIN products p ON od.product\_id = p.product\_id;

🟢 **Returns:** Only those orders that exist in all 4 tables.

## 2. LEFT JOIN

Returns all records from the **left** table and matched records from the **right** table. Fills with NULL if no match.

### Example: List all customers and their orders (even if none)

SELECT c.name AS customer\_name, o.order\_id, o.order\_date

FROM customers c

LEFT JOIN orders o ON c.customer\_id = o.customer\_id;

🟢 **Returns:** All customers — Eve will show with NULL order.

## 3. RIGHT JOIN

Returns all records from the **right** table and matched records from the **left** table.

### Example: List all orders with customer info (even if customer missing — rare but for demo)

SELECT o.order\_id, c.name AS customer\_name, o.order\_date

FROM orders o

RIGHT JOIN customers c ON o.customer\_id = c.customer\_id;

🟢 Same as LEFT JOIN but direction reversed. Useful for demonstration.

## 4. FULL OUTER JOIN (MySQL workaround using UNION)

Returns all records when there is a match in **either** table.

### Example: All customers and all orders, even unmatched ones

SELECT c.name AS customer\_name, o.order\_id

FROM customers c

LEFT JOIN orders o ON c.customer\_id = o.customer\_id

UNION

SELECT c.name AS customer\_name, o.order\_id

FROM customers c

RIGHT JOIN orders o ON c.customer\_id = o.customer\_id;

🟢 Combines LEFT + RIGHT JOIN results.

## 5. CROSS JOIN

Returns the **Cartesian product** — all combinations.

### Example: Every customer with every product

SELECT c.name AS customer\_name, p.name AS product\_name

FROM customers c

CROSS JOIN products p;

🟢 Result = #customers × #products rows

## 6. SELF JOIN

Joins a table to itself.

### Example: Find customers in the same city

SELECT A.name AS customer1, B.name AS customer2, A.city

FROM customers A

JOIN customers B ON A.city = B.city AND A.customer\_id <> B.customer\_id;

🟢 Useful for grouping or finding pairs.

## BONUS: Aggregation with JOIN

### Total price of each order

SELECT o.order\_id, c.name AS customer\_name,

SUM(od.quantity \* p.price) AS total\_price

FROM orders o

JOIN customers c ON o.customer\_id = c.customer\_id

JOIN order\_details od ON o.order\_id = od.order\_id

JOIN products p ON od.product\_id = p.product\_id

GROUP BY o.order\_id, c.name;

## Summary Table

|  |  |  |
| --- | --- | --- |
| JOIN Type | Purpose | Keyword |
| INNER JOIN | Only matching rows | JOIN or INNER JOIN |
| LEFT JOIN | All from left, matched from right | LEFT JOIN |
| RIGHT JOIN | All from right, matched from left | RIGHT JOIN |
| FULL OUTER JOIN | All from both, even unmatched | LEFT JOIN + UNION + RIGHT JOIN |
| CROSS JOIN | All combinations | CROSS JOIN |
| SELF JOIN | Join table to itself | JOIN same table with alias |