

CHEATSHEET

SQL Joins Cheatsheet

Every join type, side by side, with the query that produces each result

6

Join types

1

Base example

2

Pages

Setup used in every example

```
customers(id, name)      orders(id, customer_id, total)
1  Ada                  101  1      40
2  Grace                102  1      15
3  Alan                 103  4      99  <- customer 4 doesn't exist in customers
```

INNER JOIN — only matching rows

```
SELECT c.name, o.total
FROM customers c
INNER JOIN orders o ON o.customer_id = c.id;

-- Result: Ada|40, Ada|15  (Grace and Alan drop out, order 103 drops out)
```

LEFT JOIN — all of the left table

```
SELECT c.name, o.total
FROM customers c
LEFT JOIN orders o ON o.customer_id = c.id;

-- Result: Ada|40, Ada|15, Grace|NULL, Alan|NULL
```

- The most common join for reporting: "show every customer, with orders if they have any."
- Filter the right side inside the ON clause (not WHERE) to keep unmatched left rows in the result.

RIGHT JOIN — all of the right table

```
SELECT c.name, o.total
FROM customers c
RIGHT JOIN orders o ON o.customer_id = c.id;

-- Result: Ada|40, Ada|15, NULL|99  (order 103's customer doesn't exist)
```

FULL OUTER JOIN — everything, matched or not

```
SELECT c.name, o.total
FROM customers c
FULL OUTER JOIN orders o ON o.customer_id = c.id;

-- Result: Ada|40, Ada|15, Grace|NULL, Alan|NULL, NULL|99
```

DIALECT NOTE

MySQL has no FULL OUTER JOIN keyword. Emulate it with LEFT JOIN UNION RIGHT JOIN, or LEFT JOIN UNION ALL a RIGHT JOIN that excludes the overlap.

CROSS JOIN — every combination

```
SELECT c.name, o.id
FROM customers c
CROSS JOIN orders o;

-- Result: 3 customers x 3 orders = 9 rows, no ON clause
```

SELF JOIN — a table joined to itself

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

Quick decision table

Need	Use
Only rows that match both sides	INNER JOIN
All of table A, matches from B if any	LEFT JOIN
All of table B, matches from A if any	RIGHT JOIN
Everything from both sides	FULL OUTER JOIN
Every combination of rows (rare)	CROSS JOIN
Compare rows within the same table	SELF JOIN