D8A CHEATSHEET · #088 · LEFT JOIN + WHERE

12,400 customers. The report shows 8,310.

D8A.ACADEMY/RESOURCES/LEFT-JOIN-WHERE-TRAP

THE QUERY

`LEFT JOIN orders o`	keep every customer, orders or not
`WHERE o.status = 'paid'`	← NULL = 'paid' is not true. row gone

THE FIX

`AND o.status = 'paid'`	in the ON clause, with the join
`ON, not WHERE`	← the 4,090 customers come back
`WHERE o.id IS NULL`	customers with no paid order

REMEMBER

`NULL in WHERE`	never true, never false: dropped
`WHERE c.country`	left table: WHERE is the right place
`count twice`	COUNT(*) with and without the WHERE

WORTH MEMORIZING

01 Same join, two row counts

The condition filters orders in ON. In WHERE it filters the joined rows, and the customers with no orders are joined rows with NULLs.

WHERE: 8,310 ROWS

```sql
SELECT c.id, o.id AS order_id
FROM customers c
LEFT JOIN orders o
  ON o.customer_id = c.id
WHERE o.status = 'paid';
-- no orders: no row
```

ON: 12,400 ROWS

```sql
SELECT c.id, o.id AS order_id
FROM customers c
LEFT JOIN orders o
  ON o.customer_id = c.id
 AND o.status = 'paid';
-- no orders: order_id NULL
```

02 Trap: three conditions, three results

Where the condition sits decides which rows survive. Three cases from the same review.

`WHERE o.status = 'paid'`	For a customer with no orders, o.status is NULL. NULL = 'paid' is neither true nor false, and WHERE keeps only true. The customer is gone.
`WHERE o.status <> 'paid'`	Same thing. NULL <> 'paid' is not true either. The customer with no orders disappears from both reports.
`ON ... AND c.country = 'FR'`	A left-table filter in ON does not filter. Every non-French customer still comes back, with NULL orders. Left-table conditions go in WHERE.

→ Quick test before you send the number: run the query with and without the WHERE and compare COUNT(*). If the LEFT JOIN count equals the INNER JOIN count, the WHERE ate the NULLs.