

SQL window functions, one page.

D8A.ACADEMY/RESOURCES/WINDOW-FUNCTIONS

THE ANATOMY

```
SUM(amount) OVER (
  PARTITION BY city -- restart per group
  ORDER BY order_date -- running order
) -- one value PER ROW, no row collapses
```

RANKING ON TIES

AMOUNT	ROW_NUM	RANK	DENSE
300	1	1	1
200	2	2	2
200	3	2	2
100	4	4	3

LOOK AROUND

LAG(x)	value from the previous row
LEAD(x)	value from the next row
use for	deltas, retention, sessions

RUNNING TOTAL

```
SUM(amount) OVER (
  ORDER BY order_date
) AS running_total
```

VS GROUP BY

GROUP BY	collapses to one row per group
window	keeps every row, adds context

→ a window = a value per row, computed over a group, nothing collapses

WORTH MEMORIZING

01 Top-N per group

The interview question, solved once and for all.

```
WITH ranked AS (
  SELECT *,
    ROW_NUMBER() OVER (
      PARTITION BY city
      ORDER BY amount DESC
    ) AS rn
  FROM sales
)
SELECT * FROM ranked WHERE rn <= 3;
```

→ Top 3 sales per city, churned-user latest events, best product per category: all this exact shape.

02 Ranking a tie

Same data, three different rankings.

AMOUNT	ROW_NUMBER	RANK	DENSE_RANK
300	1	1	1
200	2	2	2
200	3	2	2
100	4	4	3

→ RANK skips after ties (1,2,2,4). DENSE_RANK does not (1,2,2,3). ROW_NUMBER never ties.

03 Trap: a window in WHERE

Window functions run after WHERE. You cannot filter on them directly.

```
-- wrong
WHERE ROW_NUMBER() OVER (...) <= 3

-- right: name it in a CTE,
-- filter in the outer query
WITH ranked AS (...)
SELECT * FROM ranked WHERE rn <= 3
```

→ Same execution-order story as sheet 005. BigQuery, Snowflake and DuckDB offer QUALIFY as the shortcut.