

GROUP BY cannot give you the top 3.

D8A.ACADEMY/RESOURCES/TOP-N-PER-GROUP

THE QUESTION

top 3 products per category, by revenue

GROUP BY ← gives one row per group, not three

THE ANSWER

ROW_NUMBER() OVER (PARTITION BY category

ORDER BY revenue DESC) the best product gets 1

WHERE rn <= 3 ← in a CTE, not the same SELECT

REMEMBER

DENSE_RANK when ties should all stay

QUALIFY BigQuery, Snowflake: no CTE needed

LIMIT 3 works for one group only

WORTH MEMORIZING

01 The query

Number the products inside each category, best first. Then keep the numbers up to 3.

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

→ The WHERE cannot see rn in the same SELECT: window functions run after WHERE. Hence the CTE. BigQuery and Snowflake have QUALIFY rn <= 3 for exactly this.

02 Trap: two products tie for third

Three functions, three answers to the same tie.

FUNCTION	TIE FOR 3RD	ROWS KEPT
ROW_NUMBER 3, 4		exactly 3, one tied product dropped
RANK 3, 3, then 5		all tied, the next one skipped
DENSE_RANK 3, 3, then 4		all tied, nothing skipped

→ Interview phrasing: I use ROW_NUMBER for exactly N rows, DENSE_RANK when tied values must all be kept, and I say which one the question needs.