Unlock the Secret Weapon of SQL Pros

What Are They?

- Think of them as a super-powered GROUP BY
- They perform calculations across a set of table rows
- Unlike GROUP BY, they don't collapse your rows
- They let you see aggregate and detail data simultaneously
- They operate on a "window" of data related to the current row
- Essential for rankings, running totals, and moving averages

The OVER() Clause

- This clause defines your window
- It's the heart of every window function
- It tells SQL how to arrange and frame the data
- An empty OVER() applies the function to the entire result set
- You can use it with aggregates like SUM() or AVG()
- It's what makes a normal function a window function

Partitioning Data

- Use PARTITION BY inside OVER() to create groups
- It splits your data into smaller windows or partitions
- The function resets for each partition
- Example: Calculate total sales per department
- Similar to GROUP BY but rows remain independent
- You can partition by one or multiple columns

Ordering Within Windows

- Use ORDER BY inside OVER() to sort the window
- Crucial for rankings and cumulative calculations
- Defines the logical order of rows in the partition
- For running totals, order by a date column
- For rankings, order by the value you're ranking
- This order is independent of your main query order

Ranking Data

- ROW_NUMBER() assigns a unique sequential number
- RANK() leaves gaps for identical values (e.g., 1,2,2,4)
- DENSE_RANK() does not leave gaps (e.g., 1,2,2,3)
- Perfect for finding top N records per category
- Useful for creating leaderboards or performance tiers
- Often used with PARTITION BY and ORDER BY

Peeking at Neighbors

- LAG() looks at the previous row in the partition
- LEAD() looks at the next row in the partition
- Great for calculating period-over-period changes
- Example: Compare today's sales to yesterday's
- You can specify how many rows to look behind/ahead
- Helps analyze trends and sequential patterns

First & Last Value

- FIRST_VALUE() returns a value from the first row in the window
- LAST_VALUE() returns a value from the last row in the window
- Useful for comparing a row to the start of a period
- Example: Compare monthly sales to January's sales
- Often used with a framed window (ROWS UNBOUNDED PRECEDING)
- Helps in tracking progress from a starting point

Real-World Use Cases

- Calculating running totals and moving averages
- Identifying top performers within each department
- Assigning customer tiers based on purchase history
- Tracking employee sales rankings month-over-month
- Finding the time between consecutive customer orders
- Analyzing website session paths and user behavior

FOLLOW FOR MORE

Like this post? Repost!

Share your thoughts in the comments!