

# Window Functions in SQL

## An Easy Guide for Beginners

### ■ Why Do We Need Window Functions?

Imagine you have a list of salespeople and their sales amounts. You want to see each person's sales **AND their ranking** in the same result.

Salesperson	Region	Amount	What We Want
Alice	East	\$100	Show rank in East
Bob	East	\$150	Show rank in East
Carol	West	\$200	Show rank in West

### The Problem with Regular SQL:

If you use **GROUP BY**, you lose the individual rows (you only get summaries). If you **don't** use **GROUP BY**, you can't calculate rankings or running totals.

### The Solution:

**Window functions** let you keep all your rows **AND** add calculations like rankings, running totals, or comparisons to other rows.

### ■ Why Do We Need ORDER BY?

Think of it like ranking students in a race:

- **Without ORDER BY:** SQL doesn't know who came 1st, 2nd, or 3rd. The order is random!
- **With ORDER BY:** You tell SQL *"rank them by finish time"* — now it makes sense.

### Window functions need ORDER BY to know:

- What order to rank things in
- What order to calculate running totals
- Which row comes "before" or "after" another

## ■ A Simple Example

Our Data:

Salesperson	Region	Amount
Alice	East	100
Bob	East	150
Eve	East	130
Carol	West	200
Dave	West	120

**Goal:** Rank each salesperson within their region (highest sales = rank 1)

## ■ The SQL Query

```
SELECT
  salesperson,
  region,
  amount,
  ROW_NUMBER() OVER (PARTITION BY region ORDER BY amount DESC) AS rank
FROM sales;
```

### Let's Break It Down:

- 1■■ **PARTITION BY region** — Create separate "groups" for each region (East and West)
- 2■■ **ORDER BY amount DESC** — Within each group, sort by amount (highest first)
- 3■■ **ROW\_NUMBER()** — Assign a number (1, 2, 3...) to each row in the group

### The Result:

Salesperson	Region	Amount	Rank
Bob	East	150	1
Eve	East	130	2
Alice	East	100	3
Carol	West	200	1
Dave	West	120	2

■ Notice: Bob is rank 1 in East, Carol is rank 1 in West — each region has its own ranking!

## ■ Key Takeaways

- **OVER** — Tells SQL you're using a window function
- **PARTITION BY** — Splits data into groups (like "regions" or "departments")
- **ORDER BY** — Sorts rows within each group so rankings/calculations make sense
- Window functions **don't remove rows** — they just add new columns with calculations!