

1 What are window functions?

A **window function** in SQL performs a calculation **across a set of rows related to the current row**, without collapsing the rows like GROUP BY does.

- **GROUP BY** aggregates rows → you lose detail.
- **Window functions** calculate over a “window” of rows → you **keep all rows**.

Think of it like looking through a sliding window over your table: you can see other rows and calculate things like sums, averages, ranks, etc., for each row **without removing it**.

2 Basic Syntax

```
SELECT
    column1,
    column2,
    AGG_FUNCTION() OVER (
        PARTITION BY columnX
        ORDER BY columnY
        ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
    ) AS window_result
FROM table_name;
```

- AGG_FUNCTION() → SUM, AVG, COUNT, RANK, ROW_NUMBER, etc.
 - OVER() → defines the “window” of rows.
 - PARTITION BY → like grouping (optional). It defines subsets of rows to calculate over.
 - ORDER BY → order the rows within each partition (important for ranking functions).
 - ROWS BETWEEN ... → defines the range of rows in the window (optional; default depends on function).
-

3 Simple Example

Imagine this table of sales:

Employee	Month	Sales
Alice	Jan	100
Alice	Feb	150
Bob	Jan	200
Bob	Feb	100

a) Total sales per employee, but keep all rows:

```
SELECT
    Employee,
    Month,
    Sales,
    SUM(Sales) OVER (PARTITION BY Employee) AS Total_Sales
FROM Sales;
```

Result:

Employee	Month	Sales	Total_Sales
Alice	Jan	100	250
Alice	Feb	150	250
Bob	Jan	200	300
Bob	Feb	100	300

- Notice: Each row **still exists**, but we can see total sales per employee.
-

b) Ranking employees by sales per month:

```
SELECT
    Employee,
    Month,
    Sales,
    RANK() OVER (ORDER BY Sales DESC) AS Sales_Rank
FROM Sales;
```

Result:

Employee	Month	Sales	Sales_Rank
Bob	Jan	200	1
Alice	Feb	150	2
Alice	Jan	100	3
Bob	Feb	100	3

- RANK() gives each row a rank **without collapsing rows**.
-

4 Key points for beginners

1. **Window functions do not reduce rows.**
 2. They allow **aggregates and rankings per row.**
 3. Use PARTITION BY to group within the window.
 4. Use ORDER BY inside OVER() for ranking or cumulative calculations.
-

Analogy:

- GROUP BY → “Summarize and throw away details.”
- WINDOW FUNCTION → “Look at your row, and also see the context of nearby or grouped rows.”