Windows Function

Window Function	Explanation	Example	Code
ROW_NUMBER()	Works like a S.No for each row within a partition. It’s commonly used when you want to uniquely number rows within specific categories without worrying about handling ties in the ranking.	Ranking each row uniquely within a category (e.g., assigning row numbers within each department).	SELECT department, employee_name, ROW_NUMBER() OVER(PARTITION BY department ORDER BY salary DESC) AS num FROM employees
RANK()	Assigns a rank to each row within a partition, with gaps if there are ties.	Ranking employees by salary within each department; ties receive the same rank, and the next rank skips	SELECT department, employee_name, RANK() OVER(PARTITION BY department ORDER BY salary DESC) AS num FROM employees
DENSE_RANK()	Assigns a rank to each row within a partition, without gaps if there are ties.	Ranking employees by sales, with ties receiving the same rank, but no skips in ranking.	SELECT department, employee_name, DENSE_RANK() OVER(PARTITION BY department ORDER BY salary DESC) AS num FROM employees
NTILE(n)	Divides the result set into n buckets and assigns a bucket number to each row within the partition.	Dividing employees' salaries into 4 parts within each department to see relative position.	SELECT department, employee_name, NTILE(4) OVER(PARTITION BY department ORDER BY salary DESC) AS part FROM employees

Window Function	Explanation	Example	Code
LAG()	Accesses data from the previous row within the same partition.	Getting the previous month's sales for comparison.	SELECT month, sales, LAG(sales, 1) OVER(ORDER BY month) AS previous_month_sales FROM sales_data
LEAD()	Accesses data from the next row within the same partition.	Getting the next month's forecasted sales based on historical data.	SELECT month, sales, LEAD(sales, 1) OVER(ORDER BY month) AS next_month_sales FROM sales_data
FIRST_VALUE()	Returns the first value in an ordered partition.	Finding the first sale amount in each category to compare subsequent sales.	SELECT category, sale_amount, FIRST_VALUE(sale_amount) OVER (PARTITION BY category ORDER BY sale_date) AS first_sale FROM sales
LAST_VALUE()	Returns the last value in an ordered partition.	Getting the last transaction date for each customer within a year.	SELECT customer_id, transaction_date, LAST_VALUE(transaction_date) OVER (PARTITION BY customer_id ORDER BY transaction_date) AS last_sale FROM sales