



SQL

Common SQL Windows function with examples



Window functions perform calculations across a set of rows related to the current row, providing values for each row in the result set.

‘employees’ table



employee_id	name	department	salary
1	Rajesh Kumar	HR	50000
2	Priya Sharma	IT	60000
3	Amit Singh	IT	55000
4	Anjali Mehta	HR	55000
5	Rahul Desai	Sales	62000



ROW_NUMBER

Assigns a unique sequential number to each row within a partition

Example:

```
SELECT employee_id, name, salary,  
       ROW_NUMBER() OVER (ORDER BY salary DESC) AS rank  
FROM employees;
```

Result:

employee_id	name	salary	rank
5	Rahul Desai	62000	1
2	Priya Sharma	60000	2
3	Amit Singh	55000	3
4	Anjali Mehta	55000	4
1	Rajesh Kumar	50000	5



RANK

Assigns a rank to each row based on a specified order, with ties sharing the same rank

Example:

```
SELECT employee_id, name, salary,  
       RANK() OVER (ORDER BY salary DESC) AS rank  
FROM employees;
```

Result:

employee_id	name	salary	rank
5	Rahul Desai	62000	1
2	Priya Sharma	60000	2
3	Amit Singh	55000	3
4	Anjali Mehta	55000	3
1	Rajesh Kumar	50000	5



DENSE_RANK

Similar to RANK, but doesn't skip ranks for ties.

Example:

```
SELECT employee_id, name, salary,  
       DENSE_RANK() OVER (ORDER BY salary DESC) AS rank  
FROM employees;
```

Result:

employee_id	name	salary	rank
5	Rahul Desai	62000	1
2	Priya Sharma	60000	2
3	Amit Singh	55000	3
4	Anjali Mehta	55000	3
1	Rajesh Kumar	50000	4



NTILE

Divides rows into a specified number of groups (tiles)

Example:

```
SELECT employee_id, name, salary,  
       NTILE(2) OVER (ORDER BY salary DESC) AS quartile  
FROM employees;
```

Result:

employee_id	name	salary	quartile
5	Rahul Desai	62000	1
2	Priya Sharma	60000	1
3	Amit Singh	55000	2
4	Anjali Mehta	55000	2
1	Rajesh Kumar	50000	2



LAG

Accesses data from the previous row.

Example:

```
SELECT employee_id, name, salary,  
       LAG(salary, 1) OVER (ORDER BY salary) AS previous_salary  
FROM employees;
```

Result:

employee_id	name	salary	previous_salary
1	Rajesh Kumar	50000	NULL
4	Anjali Mehta	55000	50000
3	Amit Singh	55000	55000
2	Priya Sharma	60000	55000
5	Rahul Desai	62000	60000



LEAD

Accesses data from the next row.

Example:

```
SELECT employee_id, name, salary,  
       LEAD(salary, 1) OVER (ORDER BY salary) AS next_salary  
FROM employees;
```

Result:

employee_id	name	salary	next_salary
1	Rajesh Kumar	50000	55000
4	Anjali Mehta	55000	55000
3	Amit Singh	55000	60000
2	Priya Sharma	60000	62000
5	Rahul Desai	62000	NULL



Found it interesting?
Give it a thumbs up!



FOLLOW

@ankit-reflects