**WINDOW FUNCTIONS AND RANKING:**

***Objective :***

To leverage window functions to perform calculations across a set of rows.

**ROW\_NUMBER()** assigns a unique sequential number to each row, with no ties and it doesn't account for duplicate values.

**RANK()** assigns a rank to each row, where rows with the same value receive the same rank, but the subsequent rank is skipped (e.g., 1, 2, 2, 4).

**DENSE\_RANK()** also assigns ranks to rows with ties but does not skip any ranks, so tied rows receive the same rank, and the next rank is consecutive (e.g., 1, 2, 2, 3).

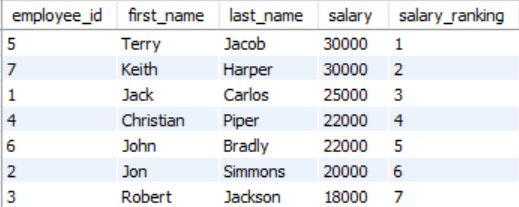
**LEAD()** is a window function in SQL that allows you to access data from a subsequent row or rows in the result set.

**LAG()** is a window function in SQL that allows you to access data from a previous row or rows in the result set.

***Queries :***

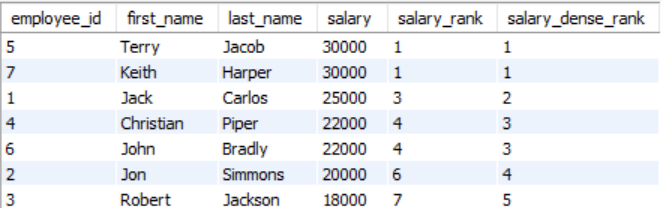
**select employee\_id, first\_name, last\_name, salary, row\_number() over(order by salary desc) as salary\_ranking from Employees;**

// Selects employee details and assigns a unique rank based on salary in descending order.



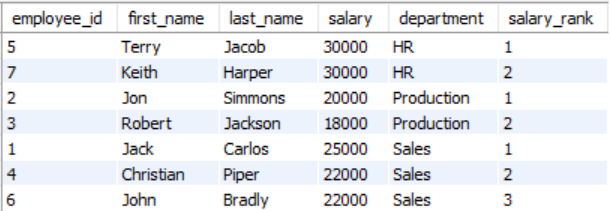
**select employee\_id, first\_name, last\_name, salary, rank() over(order by salary desc) as salary\_rank, dense\_rank() over(order by salary desc) as salary\_dense\_rank from Employees;**

// Selects employee details and calculates both a standard rank and a dense rank based on descending salary.



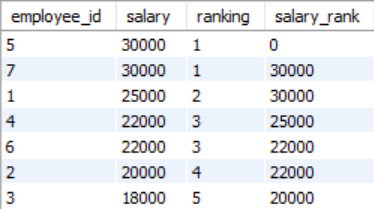
**select employee\_id, first\_name, last\_name, salary, department, row\_number() over(partition by department order by salary desc) as salary\_rank from Employees;**

// Selects employee details and ranks employees within each department by salary in descending order.



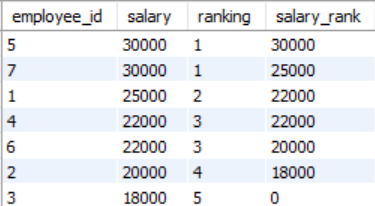
**select employee\_id, salary, dense\_rank() over(order by salary desc) as ranking, lag(salary,1,0) over(order by salary desc) as salary\_rank from Employees;**

// Selects employee salary and calculates the dense rank and the previous employee's salary based on descending salary order.



**select employee\_id, salary, dense\_rank() over(order by salary desc) as ranking, lead(salary,1,0) over(order by salary desc) as salary\_rank from Employees;**

// Selects employee salary and calculates the dense rank and the next employee's salary based on descending salary order.



**Note :** In the **over** clause, the **partition by** parameter is optional while **order by** is mandatory.