# 7. Window Functions and Ranking

## Objective

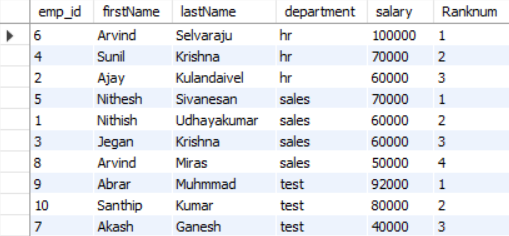
To observe the concepts of window functions.

## Understanding

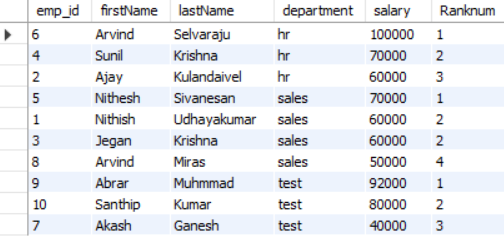
* Window functions are used to do small calculation across the rows.
* Eg: If I want to sum up the salary of three employees, We can use window function for that. It won’t take the full data.
* There three window functions such as,
  + Aggregate (count, sum, avg, max, min)
  + Ranking (dense\_rank, rank, row\_number)
  + Value (first\_value, last\_value, lag, lead)
* **Over()** - Do this for each row, looking through a window or group.
* **Partition by()** - Group it like teams in a tournament.

## Queries

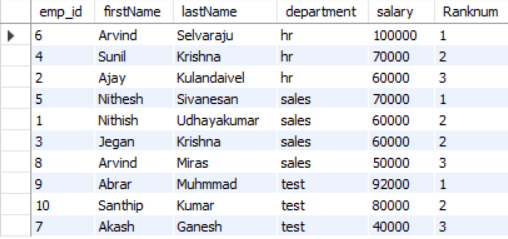
1. ***select \*, ROW\_NUMBER() OVER(PARTITION BY department ORDER BY salary DESC) AS Ranknum***
2. ***FROM employee;***



1. ***select \*, RANK() OVER(PARTITION BY department ORDER BY salary DESC) AS Ranknum***
2. ***FROM employee;***

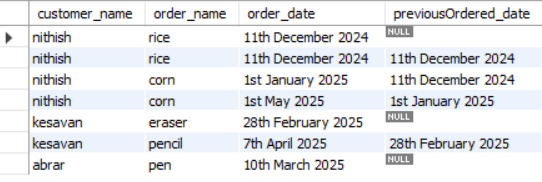


1. ***select \*, DENSE\_RANK() OVER(PARTITION BY department ORDER BY salary DESC) AS Ranknum***
2. ***FROM employee;***



* Row\_number() – Give row number to the each row.
* Dense\_rank() – Give rank to the rows, No skipping on tie
* Rank() – Give rank to the rows, Skip the rank on ties
* These result are shown in the screenshots.

1. ***select c.customer\_name,o.order\_name,***
2. ***date\_format(o.order\_date,"%D %M %Y") AS order\_date,***
3. ***date\_format(lag(o.order\_date) over(partition by o.customer\_id order by o.order\_date), "%D %M %Y") AS previousOrdered\_date***
4. ***from orders o***
5. ***left join customers c***
6. ***on o.customer\_id=c.customer\_id;***

******

* Here,I used **lag()** function to show the previous order date of the user.
* And used date\_format and left join to return formatted date and to show the customer name who ordered.