**Sql cheat cheat**

**RANKING FUNCTION**

**🡪Rank ()**

**It is used to rank the attributes values according to its attributes.**

**🡪over() can be used in rank().**

**Over contain order by attribute\_name ASC/DESC**

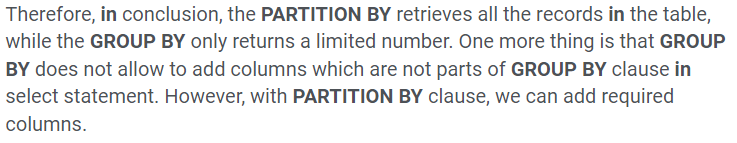
**🡪Rownumber()**

**It provides the specific row number of each row**

**🡪Partition by**

**Rank() over(partition by attribute order by attribute)**

**Difference between partition by and group by**

****

**Eg:->**

**Select name,marks,rank() over(partition by school\_position order by marks desc) as [‘Rank’] from xyz**

|  |  |  |
| --- | --- | --- |
| **name** | **marks** | **Rank** |
| **Maitri** | **97** | **2** |
| **Ketal** | **89** | **1** |

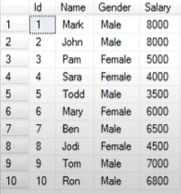
**🡪Dense rank()**

**Denserank() over(order by …)**

**Differnce between rank and denserank**

****

**Let try an example**

****

**Select \*,rank() over(order by Salary desc) as [‘Rank’], denserank() over(**

**Order by Salary desc) as [‘DenseRank’] from employee;**

****

**Window functions vs Aggregate functions : :->**

**The main difference between window functions and aggregate functions is that aggregate functions group multiple rows into a single result row; all the individual rows in the group are collapsed and their individual data is not shown. On the other hand, window functions produce a result for each individual row. This result is usually shown as a new column value in every row within the window.**

**Aggregate function collapses when data are same and produce only one result ,**

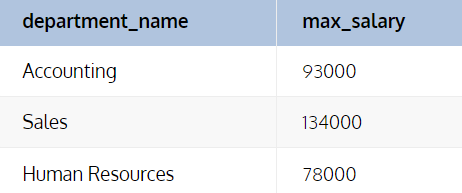
**While window function do not collapses and return all the values in multiple rows.**

**Eg🡪**

**SQL queries that return the department name and the max salary of each department.**

**Using aggregate function,**

**Select department\_name,max(salary) from xyz group by department**

****

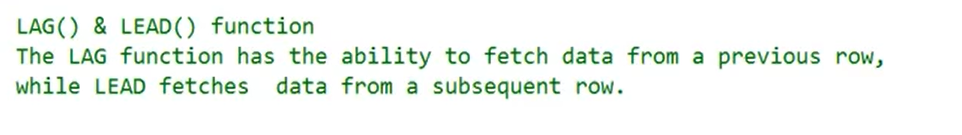
**Using Window functions,**

**Select department\_name,max(salary),over(partition**

**ANALYTIC FUNCTION**

**🡪Lead()**

**🡪Lag()**

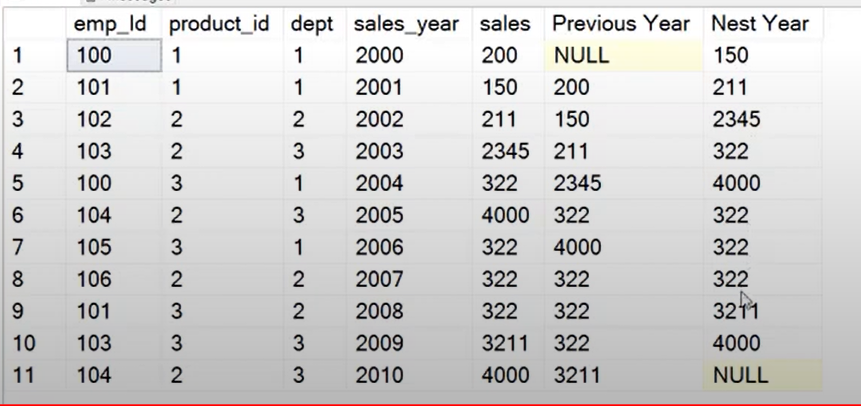
****

**Example🡪**

**Select emp\_id,prod\_id,dept,sales\_year,sales,**

**Lag(sales) over(order by sales asc) as Previous\_year,**

**Lead(sales) over(order by sales asc) as Next\_year from xyz;**

****

**🡪First\_value()**

**Returns first value of the column in all the rows.**

**🡪Last\_value()**

**Returns last\_value of the column in all the rows.**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Eg🡪**

**Select \*,first\_value() over (order by sales asc) as first value from xyz;**

****

**🡪Nth\_value()**

**Returns the nth value of the attribute which is being partitioned.**

**Eg🡪**

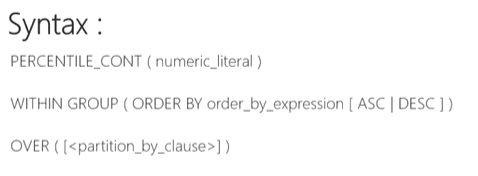
**Write a sql query to display the second most highest salary of each department along with department name.**

**Select department\_name,nth\_value(salary,2) over(partition by department\_name order by salary desc ) as SecondHighestSalary from xyz;**

**🡪Range function**

**DISTRIBUTION FUNCTION**

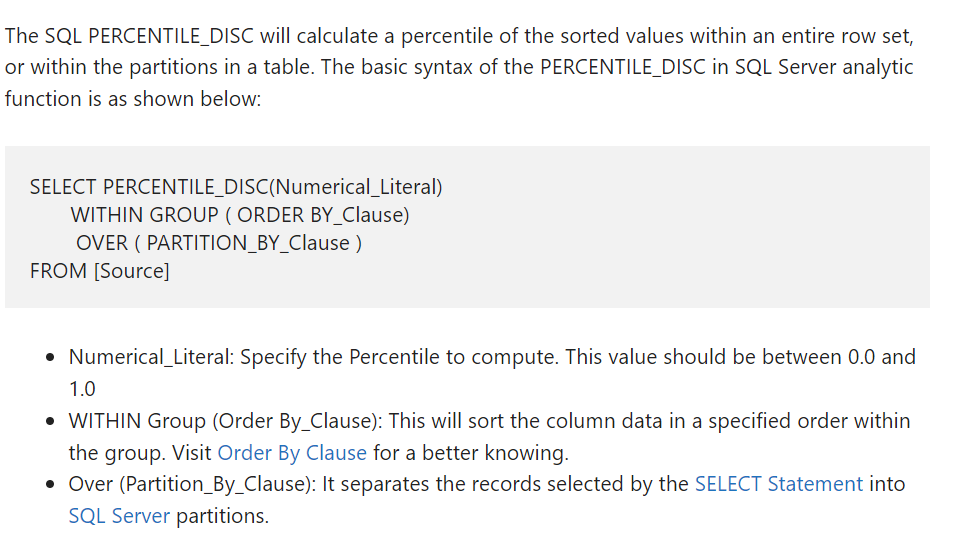
**🡪percent\_cont()**

****

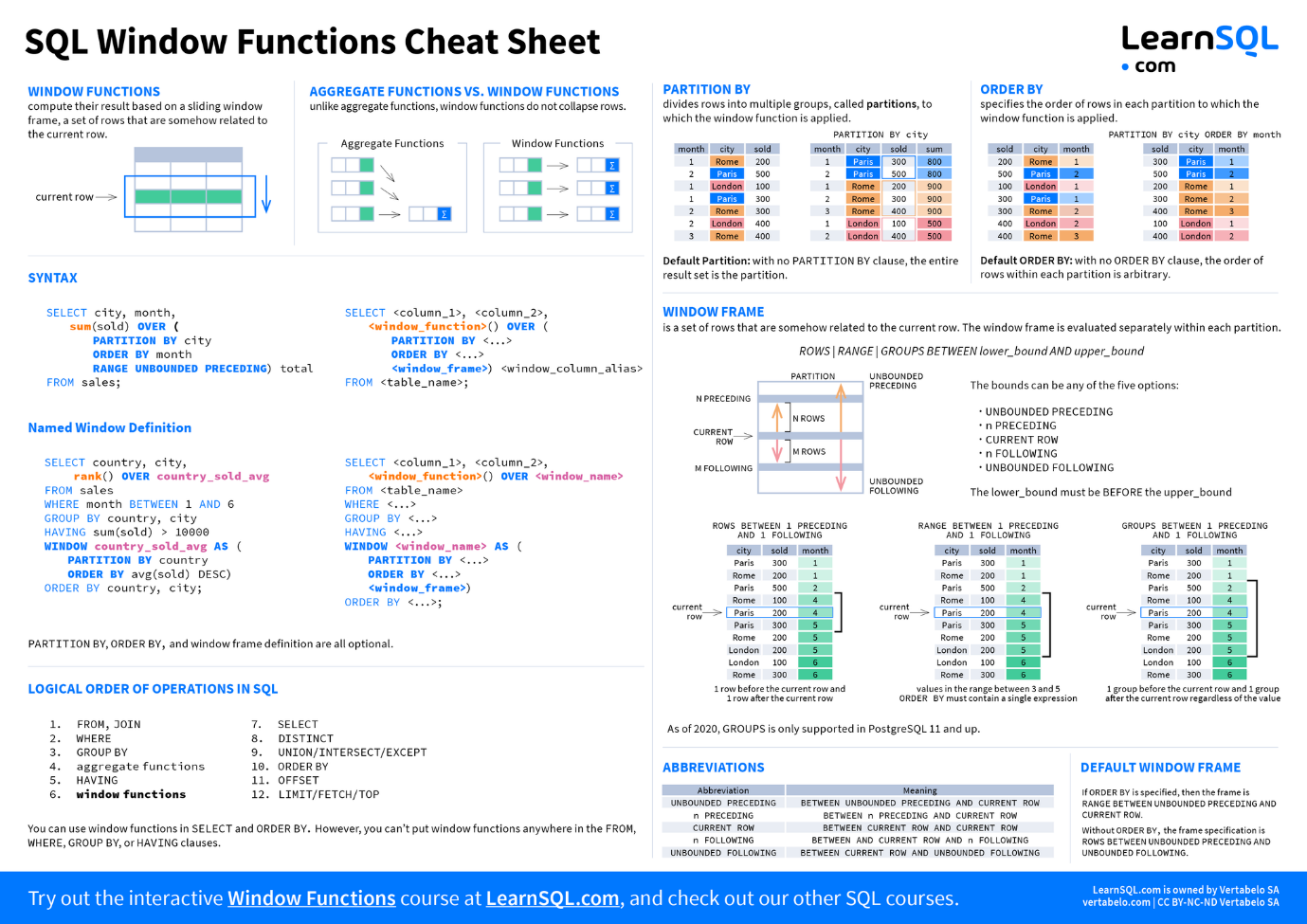
**🡪percent\_rank()**

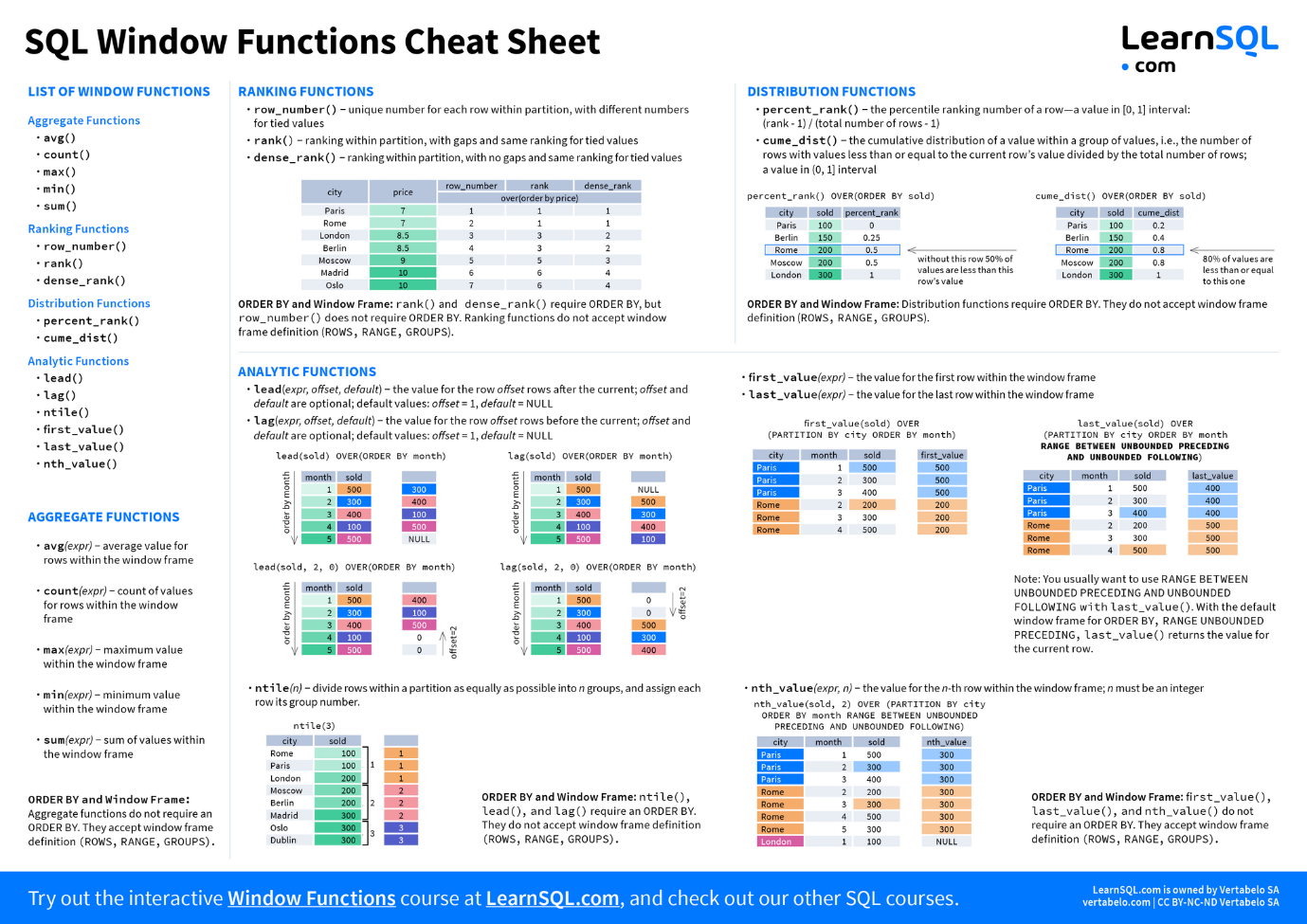
**percent\_rank() within group (order by attribute ) over(partition by attribute)**

**🡪percent\_disc()**

****

**🡪cume\_dist()**





**Source code CheatSheet Link:** [**https://learnsql.com/blog/sql-window-functions-cheat-sheet/**](https://learnsql.com/blog/sql-window-functions-cheat-sheet/)

**Global churn rate= inactive users/total users**