**SQL**

**Groupby- GROUP** BY Clause is used to **group rows with same values**

select count(comments),Comments from trndebit group by comments

**Partition by**

PARTITION BY Divides the result set into partitions.

The window function is applied to each partition separately and computation restarts for each partition.

– count – normal count

* Row\_number- id no (like 1,2,3..)
* Rank like 1,1.1.4
* dense\_rank 1,1,1,2

select count(comments) over (partition by Comments) , \* from trndebit

select Row\_number() over (partition by Comments order by Comments desc) , \* from trndebit

select rank() over (partition by Comments order by Comments desc) , \* from trndebit

select dense\_rank() over (partition by Comments order by Comments desc) , \* from trndebit

**Aggregate functions:**

An aggregate function allows you to perform a calculation on a set of values to return a single scalar value. We often use aggregate functions with the [GROUP BY](http://www.zentut.com/sql-tutorial/sql-group-by/)and [HAVING](http://www.zentut.com/sql-tutorial/sql-having/) clauses of the [SELECT](http://www.zentut.com/sql-tutorial/sql-select/)statement.

**The following are the most commonly used SQL aggregate functions:**

* **AVG** – calculates the **average** of a set of **values**.
* **COUNT** – **counts rows** in a specified table or view.
* **MIN** – gets the minimum value in a set of **values**.
* MAX – gets the maximum value in a set of **values**.
* **SUM** – calculates the **sum** of **values**.

**Joins**

* **Joins** clause is used to combine records from two or more tables in a database.
* A JOIN is a means for combining fields from two tables by using values common to each.

There are different types of joins available in SQL −

* [INNER JOIN](https://www.tutorialspoint.com/sql/sql-inner-joins.htm) − returns rows when there is a match in both tables.
* [LEFT JOIN](https://www.tutorialspoint.com/sql/sql-left-joins.htm) − returns all rows from the left table, even if there are no matches in the right table.
* [RIGHT JOIN](https://www.tutorialspoint.com/sql/sql-right-joins.htm) − returns all rows from the right table, even if there are no matches in the left table.
* [FULL JOIN](https://www.tutorialspoint.com/sql/sql-full-joins.htm) − returns rows when there is a match in one of the tables.
* [SELF JOIN](https://www.tutorialspoint.com/sql/sql-self-joins.htm) − is used to join a table to itself as if the table were two tables, temporarily renaming at least one table in the SQL statement.
* [CARTESIAN JOIN](https://www.tutorialspoint.com/sql/sql-cartesian-joins.htm) − returns the Cartesian product of the sets of records from the two or more joined tables.