



# POSTGRESQL

## CLASS 5

SCALAR AND AGGREGATE FUNCTIONS

# COUNT

- ▶ There are many aggregate functions present in the PostgreSQL database. One of the aggregate function that is used to find the row count is COUNT() aggregate function. This function counts the total number of rows according to the query statement and clauses. When it is used on a particular column then only non-NULL values are considered.
- ▶ `SELECT COUNT (* | [DISTINCT] ALL | columnName)  
FROM tableName  
[WHERE conditionalStatements];`
- ▶ The parameter that can be passed to count function can be either \* that means all the rows to be counted in the resultset or column name with distinct or all keyword prepended to it. By default, it is ALL keyword when mentioned in a particular columnName. When the distinct keyword is used unique valued columns are considered in the resultset. The table name is the name of the table from which the result is to retrieved and row count to be found. ConditionalStatements are the conditions that you wish to apply in the where clause and are optional.

# MIN

- ▶ The PostgreSQL MIN function is used for finding the minimum value for the specific column within the table and it takes a name of the column as a parameter.
- ▶ The column we can pass to the PostgreSQL MIN function can have the data types such as string, number or any other data type which can be compared.
- ▶ We can use the PostgreSQL MIN function with various clauses as well as other aggregate functions as well.
- ▶ `MIN ([DISTINCT] ALL | column_name)`

# MAX

- ▶ PostgreSQL max function is an aggregate function basically used to return the maximum value from the set of values in PostgreSQL. Max function is used in many applications like finding the maximum salary of the employee or highest mark of the student
- ▶ Max (column\_name (Name of column which was used with max function to get the maximum value from the set of values.))

# NOW

- ▶ We can get the current date and time by using the PostgreSQL NOW() function. The PostgreSQL NOW() function's return type is the timestamp with the time zone. Depending upon the current database server's time zone setting, the PostgreSQL NOW() function returns us the current date and time. We can adjust the result returned by the PostgreSQL NOW() function to other timezones as well. Also, we can get the current date and time without a timezone. This function can be used for giving the default value to the column of the table.
- ▶ `SELECT NOW()`

# RANDOM

- ▶ PostgreSQL random function is mostly useful to return a random value between 0 and 1; the default result of a random result is different at every time of the query execution. We can also return the random number between the specified range and values. The random function is essential and useful in PostgreSQL to select any random number between a series of values
- ▶ `SELECT RANDOM();`

# ROUND

- ▶ Whenever we deal with numeric values in the PostgreSQL database, the precision and format in which we retrieve those values are of immense importance. The accuracy of the numbers carries a lot of importance in real-life use cases like, for example, the precision of the measurements of certain aircraft or machine equipment or any other instrument, numeric values related to currency and transactions, etc
- ▶ `SELECT ROUND(45.145);`
- ▶ `SELECT ROUND(985.566,2);`

# MOD

- ▶ The PostgreSQL MOD() function is a mathematical function. The keyword MOD stands for the 'modulo', which is used to perform modulo operation(divide the two number and return remainder). It is used to calculate and return the remainder of a number. It takes two numbers as an input parameter. It uses a formula to divide the first number by the second number, and it returns the remainder. If the second parameter value is zero, then the MOD() function throws an error or exception, which is the divide by zero error message.DISTINCT clause in the SELECT statement
- ▶ MOD(p,q)
- ▶ **p:** This defines the number whose remainder to be calculated.
- ▶ **q:** This defines the number which is used to divide into value n.



# AGE

- ▶ PostgreSQL age() function is used to calculate the age between two dates, it will return the number of years, days, and months between the two different dates. Age function in PostgreSQL will accept the two arguments as date timestamp and return the calculated difference between two different dates. The age function in PostgreSQL is to perform the two different calculations depending on which parameter we have used with age function. Age function is very important and useful in PostgreSQL to subtract the arguments and produce the symbolic result which was used in months and years.

# AGE

- ▶ The following illustrates the syntax of the AGE() function:
- ▶ AGE(timestamp,timestamp);
- ▶ SELECT AGE('2017-01-01','2011-06-24');
- ▶ This will output: “5 years 6 mons 7 days”
- ▶ To calculate age based on the current timestamp
- ▶ SELECT current\_date,  
    AGE(timestamp '2000-01-01');
- ▶ This will output: “22 years 6 months 29 days”

# CALCULATING MODE AND MEDIAN

- ▶ `SELECT mode() WITHIN GROUP(ORDER BY unit_price) AS "Mode"`  
`FROM products;`
- ▶ `SELECT PERCENTILE_DISC(0.5) WITHIN GROUP(ORDER BY unit_price)`  
`AS "Median" FROM products;`