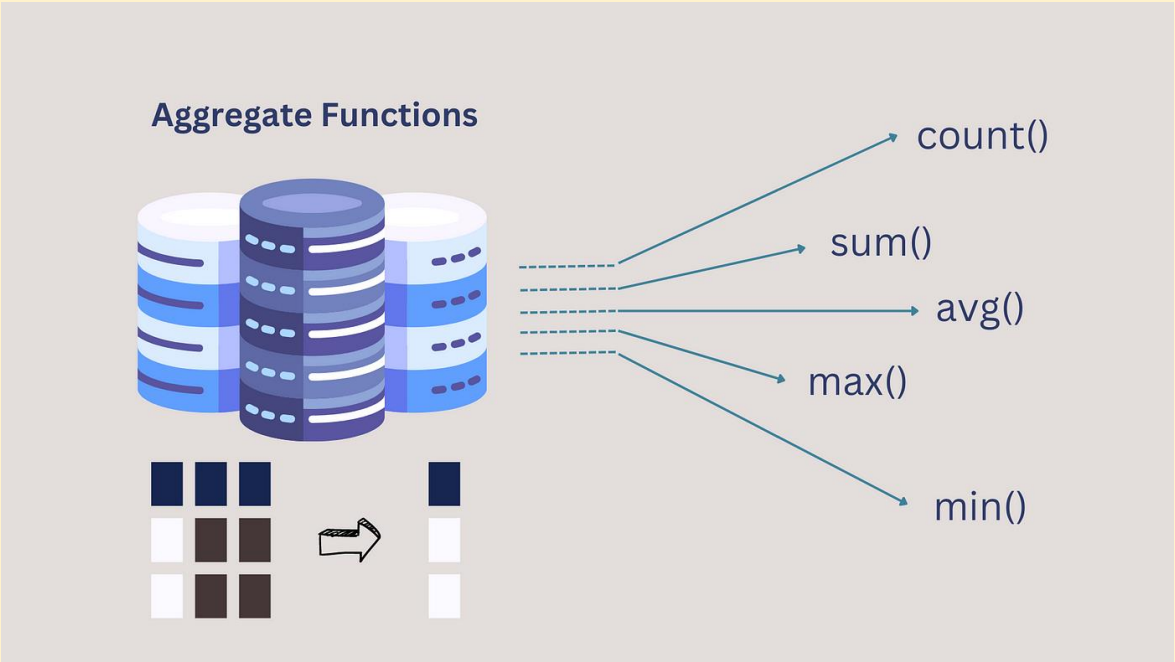
AGGREGATE FUNCTIONS IN SQL



Aggregate function: An aggregate function is a function that performs a calculation on a set of values, and returns a single value.

Aggregate functions are often used with the GROUP BY clause of the SELECT statement.

The most commonly used SQL aggregate functions are:

- **MIN()** - returns the smallest value within the selected column
- **MAX()** - returns the largest value within the selected column
- **COUNT()** - returns the number of rows in a set
- **SUM()** - returns the total sum of a numerical column
- **AVG()** - returns the average value of a numerical column

Example:

1. Count total number rows from data
   SELECT COUNT(*) AS TOTAL_ROWS FROM EMPLOYEES;
2. Find the sum of the salary
   SELECT SUM(SALARY) AS GRAND_TOTAL FROM EMPLOYEES;
3. Find the average salary of employee
   SELECT AVG(SALARY) AS AVG_SALARY FROM EMPLOYEES;
4. Find the minimum salary of the employees.
   SELECT MIN(SALARY) AS MIN_SALARY FROM EMPLOYEES;

5. Find the maximum salary of the employees
   SELECT MAX(SALARY) AS MAX_SALARY FROM EMPLOYEES;

ALIYAS(): SQL aliases are used to give a temporary name of a table, or a column in a table.
Aliases are often used to make column names more readable.
An alias only exists for the duration of that query.
An alias is created with the AS keyword.

Example: Find the maximum age of the employees

SELECT MAX(AGE) AS MAX_AGE FROM EMPLOYEES;

WHERE CLAUSE: It is used to filter the data according to condition.

Example: Extract all data from marksheets whose gender is Boy

SELECT * FROM MARKSHEET WHERE GENDER = “BOY”;

DISTINCT(): It is used to return only distinct (different) values.

Example: Count total number of unique department in data set.

SELECT COUNT(DISTINCT(DEPARTMENT)) FROM EMPLOYEES;

ORDER BY(): It is used to arrange the data in ascending order or descending order.

Example: Fetch all details and show the age of employees in descending order.

SELECT * FROM EMPLOYEES ORDER BY AGE DESC;

LIMIT CLAUSE(): It is used to extract the data from Dataset top, Bottom or Mid. The LIMIT clause is used to specify the number of records to return.

The LIMIT clause is useful on large tables with thousands of records. Returning a large number of records can impact performance.

Example: Show the data from employees to show 5 bottom employees based of age.

SELECT * FROM EMPLOYEES ORDER BY AGE LIMIT 5;

GROUP BY(): It is used to analyze the large amount of data in easy manner with the help of aggregate function. Group by are used to group the same category of data with aggregate functions such as sum, count, average, max, min.

Example: Find the maximum salary of each department.

SELECT MAX(SALARY) FROM EMPLOYEES GROUP BY DEPARTMENT;

CASE IN SQL: Case are used to divide the data according to condition, when the first condition is true, it will stop reading and return result. If No condition are true, it returns the value in the else clause.

Example: Write a SQL query to show the status if salary is greater than 50k then show GOOD else shoe BAD.

```
SELECT *,
CASE
WHEN SALARY > 50000 THEN "GOOD"
ELSE "BAD" END AS SATUS
FROM EMPLOYEES;
```

HAVING CLAUSE (): It is used to filter the data after using group by function, it is used to filter the data according to condition but its work with non existing data like Group by, joins.

The HAVING CLAUSE was added to SQL because the WHERE keyword cannot be used with aggregate functions.

Example: Show the minimum age of employees in each department whose age is less than 20.

```
SELECT DEPARTMENT, MIN(AGE) AS MIN_AGE FROM EMPLOYEES GROUP BY DEPARTMENT HAVING MIN_AGE<20;
```

UNION: It is used to combine the same category of query and its remove the duplicate value.

UNION ALL: It is used to combine the same category the same of query and its allow duplicate values.

Example:

```
SELECT * FROM EMPLOYEES WHERE AGE = 21
UNION
SELECT * FROM EMPLOYEES WHERE SALARY = 28353;
```