SQL Aggregate Functions

An **aggregate function** performs a calculation on a set of values and returns a **single summarized value** as the result.

They are often used with the `GROUP BY` clause to group data and perform calculations on each group separately.

Key Points about Aggregate Functions:

- They **ignore `NULL` values**, except for `COUNT(*)`.
- They **return a single value** per group (or for the entire table if `GROUP BY` is not used).

Common SQL Aggregate Functions

1. `MIN()` – Minimum Value

Returns the **smallest (minimum)** value in a column.
Can be used with `GROUP BY` to find the minimum value within each group.

2. `MAX()` – Maximum Value

Returns the **largest (maximum)** value in a column.
Can be used with `GROUP BY` to find the maximum value within each group.

3. `COUNT()` – Count Rows

Counts the number of **rows in a table or a specific column**.

- `COUNT(*)` counts **all rows, including `NULL`s**.
- `COUNT(column_name)` counts **only non-NULL values** in the column.
  Can be used with `GROUP BY` to count rows in each group.

4. `SUM()` – Total Sum

Returns the **total sum** of a numerical column.
Ignores `NULL` values.
Can be used with `GROUP BY` to get the sum for each group.

5. `AVG()` – Average Value

Returns the **average (mean)** value of a numerical column.
Ignores `NULL` values.
Can be used with `GROUP BY` to get the average for each group.

Handling NULL Values in Aggregate Functions

- Aggregate functions **ignore `NULL` values**, except `COUNT(*)`.
- `SUM()`, `AVG()`, `MIN()`, and `MAX()` do not consider `NULL` values in calculations.

- `COUNT(*)` includes `NULL` rows, but `COUNT(column_name)` does not.

Using Aggregate Functions Without `GROUP BY`

- If no `GROUP BY` is used, the aggregate function is applied **to all rows** as a single group.

Using Multiple Aggregate Functions Together

Multiple aggregate functions can be used in a single query to get summarized data. They can also be combined with `GROUP BY` to apply them to different groups.

Interview Questions on SQL Aggregate Functions

1. What is the difference between `COUNT(*)` and `COUNT(column_name)`?

- `COUNT(*)` **counts all rows**, including `NULL` values.
- `COUNT(column_name)` **counts only non-NULL values** in that column.

2. How do aggregate functions handle `NULL` values?

- `SUM()`, `AVG()`, `MIN()`, and `MAX()` **ignore** `NULL` values.
- `COUNT(*)` counts `NULL` rows, but `COUNT(column_name)` ignores `NULL`s.

3. What happens if you use an aggregate function without `GROUP BY`?

- The aggregate function applies to **all rows** as a single group.

4. How can you use aggregate functions with filtering (`HAVING` clause)?

- The `HAVING` clause filters groups based on aggregate function results.

Summary Table

Function	Description	NULL Handling
`MIN()`	Returns smallest value	Ignores `NULL`s
`MAX()`	Returns largest value	Ignores `NULL`s
`COUNT(*)`	Counts all rows	Includes `NULL`s
`COUNT(column)`	Counts non-null values	Ignores `NULL`s
`SUM()`	Returns total sum	Ignores `NULL`s
`AVG()`	Returns average value	Ignores `NULL`s

SQL `MIN()` and `MAX()` Functions

`MIN()` Function – Find the Smallest Value

The `MIN()` function returns the **smallest** value in a specified column. It is commonly used to find the **minimum numerical value** or **earliest date** in a table.

`MAX()` Function – Find the Largest Value

The `MAX()` function returns the **largest** value in a specified column. It is used to find the **maximum numerical value** or **latest date** in a table.

Key Points:

- Works on **numeric**, **date/time**, and **string** values.
- Ignores `NULL` values.
- Can be used with or without `GROUP BY`.

Examples of `MIN()` and `MAX()`

1. Finding the Minimum and Maximum Salary

sql Copy Edit

```sql
SELECT MIN(Salary) AS LowestSalary, MAX(Salary) AS HighestSalary
FROM Employees;
```

2. Finding the Earliest and Latest Hire Date

sql Copy Edit

```sql
SELECT MIN(HireDate) AS FirstEmployee, MAX(HireDate) AS LastEmployee
FROM Employees;
```

3. Using `GROUP BY` to Find Minimum and Maximum Salary per Department

sql Copy Edit

```sql
SELECT Department,
       MIN(Salary) AS LowestSalary,
       MAX(Salary) AS HighestSalary
FROM Employees
GROUP BY Department;
```

4. Finding the Employee with the Lowest and Highest Salary

```sql
SELECT * FROM Employees
WHERE Salary = (SELECT MIN(Salary) FROM Employees);
```

```sql
SELECT * FROM Employees
WHERE Salary = (SELECT MAX(Salary) FROM Employees);
```

SQL `COUNT()` Function

The `COUNT()` function is used to **count the number of rows** in a result set. It is commonly used to determine the total number of records in a table or the count of specific values in a column.

Types of `COUNT()` Usage

1. `COUNT(*)` – Counts All Rows (Including NULLs)

Counts the total number of rows in a table, **including rows with `NULL` values**.

2. `COUNT(column_name)` – Counts Non-NULL Values

Counts only the **non-NULL** values in a specified column. It ignores `NULL` values.

3. `COUNT(DISTINCT column_name)` – Counts Unique Non-NULL Values

Counts the number of **distinct (unique) non-NULL values** in a column.

1. Count Total Employees (Including NULL Values)

```sql
SELECT COUNT(*) AS TotalEmployees
FROM Employees;
```

2. Count Employees with Non-NULL Email Addresses

```sql
SELECT COUNT(Email) AS EmployeesWithEmail
FROM Employees;
```

(Only counts rows where `Email` is NOT `NULL`.)

3. Count Unique Job Titles

```sql
SELECT COUNT(DISTINCT JobTitle) AS UniqueJobTitles
FROM Employees;
```

(Counts only distinct, non-NULL job titles.)

4. Count Employees Per Department

```sql
SELECT Department, COUNT(*) AS EmployeeCount
FROM Employees
GROUP BY Department;
```

(Groups employees by department and counts the number of employees in each.)

5. Count Employees with Salary Greater Than 50,000

```sql
SELECT COUNT(*) AS HighSalaryEmployees
FROM Employees
WHERE Salary > 50000;
```

Key Points About `COUNT()`

- `COUNT(*)` includes all rows, even those with `NULL` values.
- `COUNT(column_name)` ignores `NULL` values.
- `COUNT(DISTINCT column_name)` counts only unique, non-NULL values.
- Can be used with `GROUP BY` to get counts for specific groups.
- Can be combined with `HAVING` to filter results based on counts.

SQL `SUM()` Function

The `SUM()` function in SQL is used to **calculate the total sum of a numeric column**. It is commonly used for financial and analytical calculations, such as summing salaries, sales, or inventory values.

Key Points About `SUM()`

- Works **only on numeric columns** (e.g., `INT`, `DECIMAL`, `FLOAT`).
- **Ignores `NULL` values** when computing the total.
- Can be used with `GROUP BY` to get **totals for different groups**.
- Can be combined with `HAVING` to filter results based on sums.

1. Calculate the Total Salary of All Employees

sql Copy Edit

```sql
SELECT SUM(Salary) AS TotalSalary
FROM Employees;
```

2. Calculate Total Sales for a Specific Product

sql Copy Edit

```sql
SELECT SUM(SalesAmount) AS TotalSales
FROM Sales
WHERE ProductID = 101;
```

3. Calculate Total Salary Per Department

sql Copy Edit

```sql
SELECT Department, SUM(Salary) AS TotalSalary
FROM Employees
GROUP BY Department;
```

(Groups employees by department and calculates total salary per department.)

SQL `AVG()` Function

The `AVG()` function in SQL is used to **calculate the average (mean) of a numeric column**. It is commonly used for financial and statistical analysis, such as calculating the average salary, product price, or exam scores.

Key Points About `AVG()`

- Works **only on numeric columns** (e.g., `INT`, `DECIMAL`, `FLOAT`).
- **Ignores `NULL` values** in calculations.
- Can be used with `GROUP BY` to calculate **averages for different groups**.
- Can be combined with `HAVING` to filter results based on averages.

Vikas_8460716214

1. Calculate the Average Salary of All Employees

sql Copy Edit

```sql
SELECT AVG(Salary) AS AverageSalary
FROM Employees;
```

2. Calculate the Average Price of Products in a Category

sql Copy Edit

```sql
SELECT AVG(Price) AS AveragePrice
FROM Products
WHERE Category = 'Electronics';
```