**SQL Aggregation Functions**

# Index (Table of Contents)

# Introduction to SQL Aggregation Functions

# Common SQL Aggregation Functions

# COUNT

# SUM

# AVG

# MIN

# MAX

# Group By Clause

# Having Clause

# Distinct

# Queries Solving

**Aggregation Functions in SQL**

Aggregation functions let you calculate summary values across multiple rows, like totals, averages, counts, or min/max values.

They work best with the **GROUP BY** clause, which groups rows by specific attributes and applies the function to each group separately. This helps turn large datasets into easy-to-read summaries.

**Key Points**

* Work on a set of values and return a single scaler value.
* Common uses: totals (SUM), averages (AVG), counts (COUNT), smallest/largest values (MIN, MAX).
* They don’t change the original data [Non-Destructive]—only produce derived results.
* Often paired with **SELECT** to show both raw columns and aggregated values.

|  |
| --- |
| SELECT COUNT(\*) AS row\_count,  COUNT(employee\_id) AS employee\_ids\_present,  SUM(salary) AS total\_salary,  AVG(salary) AS avg\_salary,  MIN(salary) AS min\_salary,  MAX(salary) AS max\_salary FROM employees; |

## Use Cases of Aggregation Functions:

1. **Summarizing data:** total sales, average grades, employee counts by department.
2. **Statistical analysis:** mean, min, max, and distribution across categories.
3. **Data reporting:** summary KPIs for dashboards and decision-making.

# Common SQL Aggregation Functions

## COUNT: Returns the number of rows that match a specified condition. Variants differ in NULL handling:

|  |
| --- |
| **## Count all rows** SELECT COUNT(\*) FROM employees; |
| **## Count non-NULL values in a column** SELECT COUNT(employee\_id) AS employee\_count FROM employees; |
| **Example:** **Count the number of employees in a company?**  SELECT COUNT(employee\_id) AS employee\_count FROM employees; |

* **SUM:** Returns the total sum of a numeric column. NULL values are ignored.

**Example:** **Calculate the total salary paid to all employees?**

|  |
| --- |
| SELECT SUM(salary) AS total\_salary  FROM employees; |

## AVG: Returns the average value of a numeric column. NULL values are ignored.

**Example:** **Calculate the total salary paid to all employees?**

|  |
| --- |
| SELECT AVG(salary) AS average\_salary FROM employees; |

## MIN: Returns the smallest value in a column.

**Example:** **Get the minimum salary of the employees?**

|  |
| --- |
| SELECT MIN(salary) AS minimum\_salary FROM employees; |

## MAX: Returns the largest value in a column.

**Example:** **Get the maximum salary of the employees?**

|  |
| --- |
| SELECT MAX(salary) AS maximum\_salary FROM employees; |

# Group By Clause

It is used to group rows that have the same values into summary rows & is often used with aggregation functions (like COUNT, SUM, AVG, etc.) to perform calculations for each group of data.

**Syntax:**

|  |
| --- |
| SELECT column1, aggregate\_function(column2) FROM table\_name WHERE condition GROUP BY column1; |

**Example:** **Group the data by department and calculate average salary?**

|  |
| --- |
| SELECT department, AVG(salary) AS average\_salary FROM employees GROUP BY department; |

## GROUP BY Multiple Columns

Group data by multiple columns to create more specific groupings. Each unique combination creates a separate group.

**Example:** **Group employees by department and hire\_date to calculate average salary for each group?**

|  |
| --- |
| SELECT department, hire\_date, AVG(salary) AS average\_salary FROM employees GROUP BY department, hire\_date; |

# Having Clause

It is used to filter records after they have been grouped. It is similar to the WHERE clause but is used with aggregated data, while WHERE works on individual rows.

**Syntax:**

|  |
| --- |
| SELECT column1, aggregate\_function(column2) FROM table\_name GROUP BY column1 HAVING aggregate\_function(column2) condition; |

**Example:** **Find departments where the average salary is greater than $75,000?**

|  |
| --- |
| SELECT department, AVG(salary) AS average\_salary FROM employees GROUP BY department HAVING AVG(salary) > 75000; |

# Distinct

DISTINCT removes duplicate rows from the result set. It can be applied to one or more columns. When multiple columns are specified, uniqueness is determined by the combination of all columns.

**Performance note:** DISTINCT can be expensive on large datasets due to deduplication processing.

**Syntax:**

|  |
| --- |
| SELECT DISTINCT column1, column2, ... FROM table\_name; |

# Queries Solving

## Query 1: Count Employees Hired After 2020-01-01 in Each Department?

|  |
| --- |
| SELECT department, COUNT(employee\_id) AS employee\_count FROM employees WHERE hire\_date > '2020-01-01' GROUP BY department; |

## Query 2: Total Salary of Employees with a Salary Above $80,000 by Department?

|  |
| --- |
| SELECT department, SUM(salary) AS total\_salary FROM employees WHERE salary > 80000 GROUP BY department; |

## Query 3: Average Salary of Employees in Departments with More Than 5 Employees?

|  |
| --- |
| SELECT department, AVG(salary) AS average\_salary FROM employees GROUP BY department HAVING COUNT(employee\_id) > 5; |

## Query 4: Minimum Salary of Employees Hired After 2018 by Department?

|  |
| --- |
| SELECT department, MIN(salary) AS min\_salary FROM employees WHERE hire\_date > '2018-01-01' GROUP BY department; |

## Query 5: Maximum Salary in Each Department for Employees with 'Manager' in their First Name?

|  |
| --- |
| SELECT department, MAX(salary) AS max\_salary FROM employees WHERE first\_name LIKE '%Manager%' GROUP BY department; |

## Query 6: Total Salary and Count of Employees Grouped by Department and Hire Date?

|  |
| --- |
| SELECT department, hire\_date, SUM(salary) AS total\_salary, COUNT(employee\_id) AS emp\_count FROM employees GROUP BY department, hire\_date; |

## Query 7: Departments with Total Salary Greater than $500,000?

|  |
| --- |
| SELECT department, SUM(salary) AS total\_salary FROM employees GROUP BY department HAVING SUM(salary) > 500000; |

## Query 8: Average Salary of Employees Hired Before 2016 Grouped by Department?

|  |
| --- |
| SELECT department, AVG(salary) AS average\_salary FROM employees WHERE hire\_date < '2016-01-01' GROUP BY department; |

## Query 9: Count of Employees Grouped by Hire Year?

|  |
| --- |
| SELECT EXTRACT(YEAR FROM hire\_date) AS hire\_year, COUNT(employee\_id) AS emp\_count FROM employees GROUP BY EXTRACT(YEAR FROM hire\_date); |

## Query 10: Departments Where the Maximum Salary is Above $100,000?

|  |
| --- |
| SELECT department, MAX(salary) AS max\_salary FROM employees GROUP BY department HAVING MAX(salary) > 100000; |