SQL Aggregate Functions in SQL*Plus (Oracle) & MySQL

Aggregate functions perform calculations on multiple rows and return a **single** value. These functions are useful in **summarizing** data, such as totals, averages, counts, etc.

Below are detailed examples covering all aggregate functions in **SQL*Plus (Oracle)** and **MySQL**.

1. Aggregate Functions in SQL*Plus (Oracle)

1.1 SUM() – Total of a Column

Done

```
SELECT SUM(salary) AS total_salary FROM employees; -- Total salary of
all employees
SELECT department_id, SUM(salary) FROM employees GROUP BY
department_id; -- Total salary per department
```

1.2 AVG() – Average of a Column

Done

```
SELECT AVG(salary) AS avg_salary FROM employees; -- Average salary
SELECT department_id, AVG(salary) FROM employees GROUP BY
department_id; -- Average salary per department
```

1.3 COUNT() – Counting Rows

Done

```
SELECT COUNT(*) FROM employees; -- Total number of employees SELECT
COUNT(employee_id) FROM employees WHERE department_id = 10; --
Count employees in department 10
```

```
SELECT department_id, COUNT(*) FROM employees GROUP BY department_id;
-- Count per department
```

1.4 MAX() – Maximum Value

Done

```
SELECT MAX(salary) FROM employees; -- Highest salary in the company
SELECT department_id, MAX(salary) FROM employees GROUP BY
department_id; -- Highest salary per department
```

1.5 MIN() – Minimum Value

Done

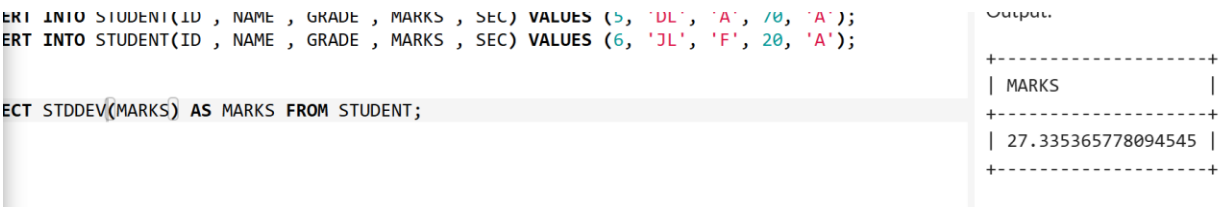
```
SELECT MIN(salary) FROM employees; -- Lowest salary in the company
SELECT department_id, MIN(salary) FROM employees GROUP BY
department_id; -- Lowest salary per department
```

1.6 MEDIAN() – Median Value (Oracle-Only)

NA

```
SELECT MEDIAN(salary) FROM employees; -- Median salary of all
employees
```

1.7 STDDEV() – Standard Deviation




```
SELECT STDDEV(salary) FROM employees; -- Standard deviation of
salaries
```

```
SELECT department_id, STDDEV(salary) FROM employees GROUP BY
department_id; -- Std deviation per department
```

1.8 VARIANCE() – Variance of a Column

```
SELECT VARIANCE(MARKS) AS MARKS FROM STUD
```

Output:

```
+-------------------+
| MARKS             |
+-------------------+
| 747.2222222222223 |
+-------------------+
```

```
SELECT VARIANCE(salary) FROM employees; -- Variance of salaries
```

1.9 GROUP BY with Aggregate Functions

Done

```
SELECT department_id, COUNT(*), AVG(salary), MAX(salary), MIN(salary)
FROM employees
GROUP BY department_id; -- Aggregate calculations per department
```

1.10 HAVING Clause (Filtering Groups)

Done

```
SELECT department_id, COUNT(*) AS employee_count
FROM employees
GROUP BY department_id
HAVING COUNT(*) > 5; -- Only departments with more than 5 employees
```

2. Aggregate Functions in MySQL

2.1 SUM() – Total of a Column

Done

```
SELECT SUM(salary) AS total_salary FROM employees; -- Total salary of
all employees
SELECT department_id, SUM(salary) FROM employees GROUP BY
department_id; -- Total salary per department
```

2.2 AVG() – Average of a Column

Done

```
SELECT AVG(salary) AS avg_salary FROM employees; -- Average salary
SELECT department_id, AVG(salary) FROM employees GROUP BY
department_id; -- Average salary per department
```

2.3 COUNT() – Counting Rows

Done

```
SELECT COUNT(*) FROM employees; -- Total number of employees SELECT
COUNT(employee_id) FROM employees WHERE department_id = 10; --
Count employees in department 10
SELECT department_id, COUNT(*) FROM employees GROUP BY department_id;
-- Count per department
```

2.4 MAX() – Maximum Value

Done

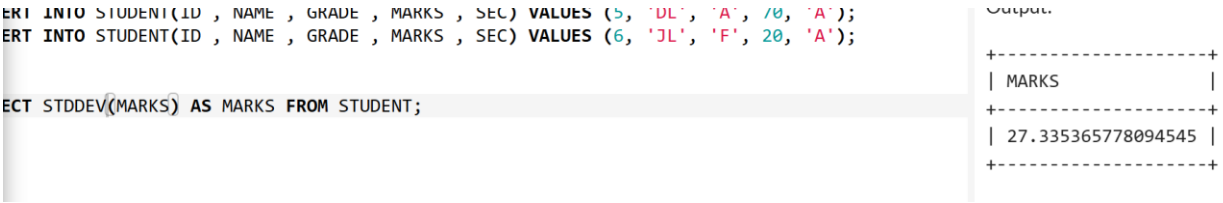
```
SELECT MAX(salary) FROM employees; -- Highest salary in the company
SELECT department_id, MAX(salary) FROM employees GROUP BY
department_id; -- Highest salary per department
```

2.5 MIN() – Minimum Value

Done

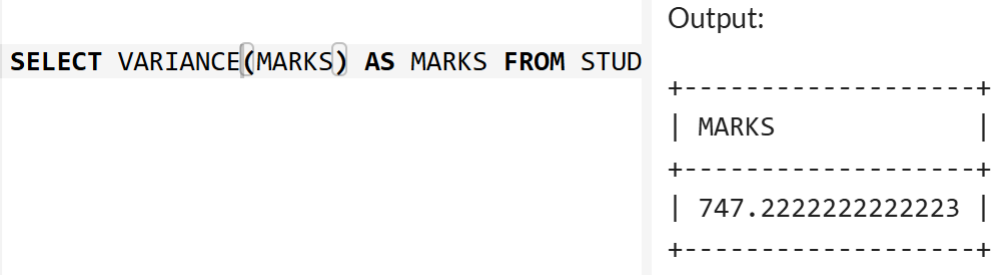
```
SELECT MIN(salary) FROM employees; -- Lowest salary in the company
SELECT department_id, MIN(salary) FROM employees GROUP BY
department_id; -- Lowest salary per department
```

2.6 STDDEV() – Standard Deviation




```
SELECT STDDEV(salary) FROM employees; -- Standard deviation of
salaries
SELECT department_id, STDDEV(salary) FROM employees GROUP BY
department_id; -- Std deviation per department
```

2.7 VARIANCE() – Variance of a Column




```
SELECT VARIANCE(salary) FROM employees; -- Variance of salaries
```

2.8 GROUP BY with Aggregate Functions

Done

```
SELECT department_id, COUNT(*), AVG(salary), MAX(salary), MIN(salary)
FROM employees
GROUP BY department_id; -- Aggregate calculations per department
```

2.9 HAVING Clause (Filtering Groups)

Done

```
SELECT department_id, COUNT(*) AS employee_count
FROM employees
GROUP BY department_id
HAVING COUNT(*) > 5; -- Only departments with more than 5 employees
```

---

3. Key Differences Between SQL*Plus (Oracle) and MySQL Aggregate Functions

Feature	Oracle (SQL*Plus)	MySQL
Basic Aggregate Functions	`SUM()`, `AVG()`, `COUNT()`, `MAX()`, `MIN()`	`SUM()`, `AVG()`, `COUNT()`, `MAX()`, `MIN()`
Median Calculation	`MEDIAN()`	Not available (requires workaround)
Standard Deviation	`STDDEV()`	`STDDEV()`
Variance Calculation	`VARIANCE()`	`VARIANCE()`
Handling NULL values	Ignores NULL values in aggregate functions	Ignores NULL values in aggregate functions
HAVING Clause	Used after `GROUP BY`	Used after `GROUP BY`
GROUP BY	Used to group and aggregate	Used to group and aggregate

---

4. Special Notes

- **Oracle** provides the `MEDIAN()` function, but **MySQL does not**. In MySQL, median must be calculated using a workaround.
- **Both** ignore NULL values when computing aggregates unless explicitly handled.
- **HAVING** is used **after GROUP BY** to filter aggregated results in both Oracle and MySQL.