Statistical Functions

SQL statistical functions are essential tools for extracting meaningful insights from databases. These functions, such as **AVG()**, **SUM()**, **COUNT()**, **MIN()**, **MAX()**, **STDDEV()**, and **VAR()**, enable users to perform statistical calculations on numeric data. Whether determining averages, sums, counts, or measures of variability, these functions empower efficient data analysis within the SQL environment.

What is Statistics?

So, **Statistics** is a branch of mathematics that deals with data collection, analysis, interpretation, presentation, and organization. It involves the use of mathematical techniques to extract meaningful information from data. Statistics is widely used in various fields such as business, economics, social science, medicine, and engineering.

Where do We Use Stats?

let's define Statistical Function, A Statistical function is a mathematical function that helps us to process and analyze data to provide meaningful information about the dataset. For example mean, sum, min, max, standard deviation, etc.

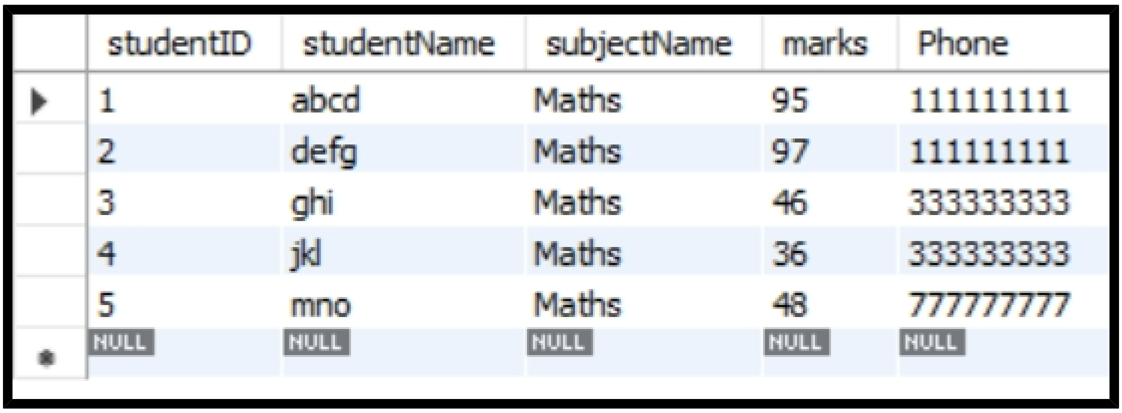
Statistical Functions in SQL

Here are Some Common Statistical Functions in SQL:

Function	Output
AVG()	Calculates the average value of a numeric column.
SUM()	Calculates the sum of values in a numeric column.
COUNT()	Counts the number of rows in a result set or the number of non-null values in a column.
MIN()	Returns the minimum value in a column.
MAX()	Returns the maximum value in a column.
VAR() / VARIANCE()	Calculates the population variance of a numeric column.
CORR()	Calculates the correlation coefficient between two numeric columns.
COVAR_POP()	Calculates the population covariance between two numeric columns.
PERCENTILE_CONT()	Calculates a specified percentile value for a numeric column.

Statistical Functions With Exmaple

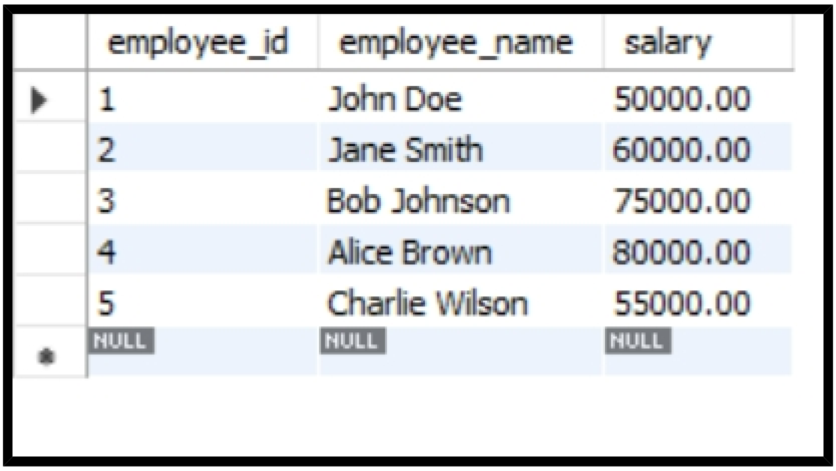
We have four tables in our database: 'studentDetails,' 'employees,' 'sales_data,' and 'financial_data.' (The pictures are displayed below.)



	studentID	studentName	subjectName	marks	Phone
▶	1	abcd	Maths	95	111111111
	2	defg	Maths	97	111111111
	3	ghi	Maths	46	333333333
	4	jkl	Maths	36	333333333
	5	mno	Maths	48	777777777
*	NULL	NULL	NULL	NULL	NULL

Table : StudentDetails

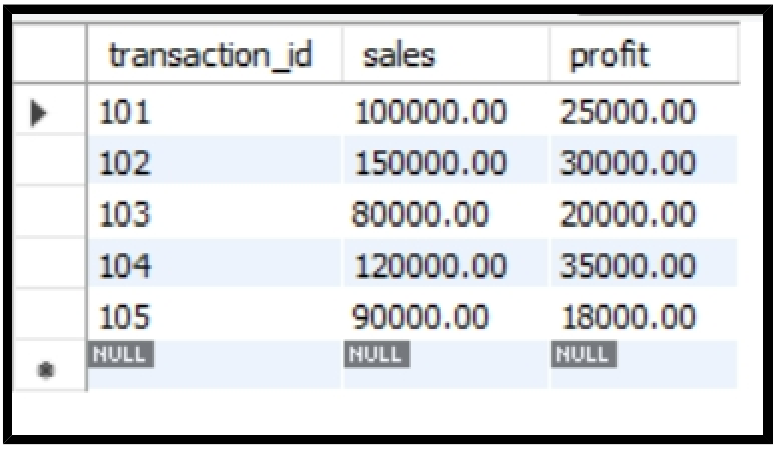
employees Table



	employee_id	employee_name	salary
▶	1	John Doe	50000.00
	2	Jane Smith	60000.00
	3	Bob Johnson	75000.00
	4	Alice Brown	80000.00
	5	Charlie Wilson	55000.00
*	NULL	NULL	NULL

Table:Employees

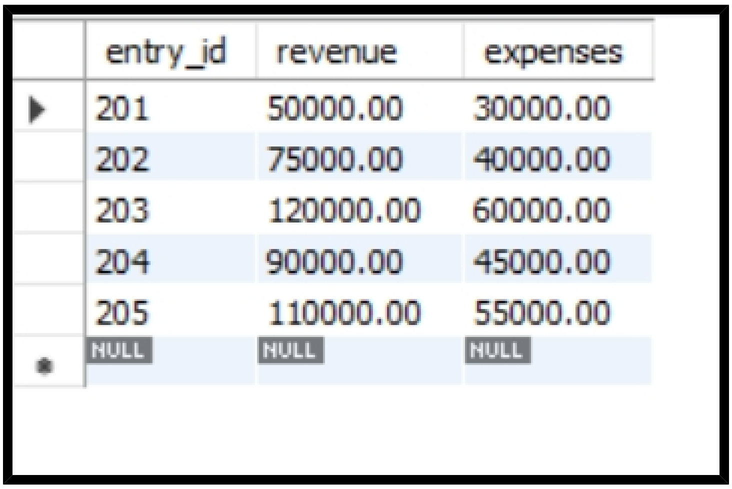
sales_data



	transaction_id	sales	profit
▶	101	100000.00	25000.00
	102	150000.00	30000.00
	103	80000.00	20000.00
	104	120000.00	35000.00
	105	90000.00	18000.00
*	NULL	NULL	NULL

Table:Sales_data

financial_data



	entry_id	revenue	expenses
▶	201	50000.00	30000.00
	202	75000.00	40000.00
	203	120000.00	60000.00
	204	90000.00	45000.00
	205	110000.00	55000.00
*	NULL	NULL	NULL

Table: financial_data

1. AVG() Function

Calculate the average or arithmetic mean for a group of numbers or a numeric column.

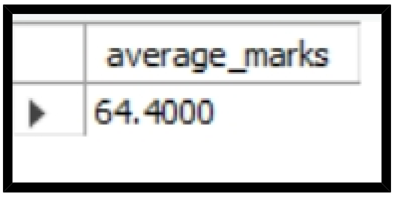
Syntax

```
SELECT AVG(column_name) FROM table_name;
```

Example Query

```
SELECT AVG(marks) AS average_marks FROM studentDetails;
```

Output



	average_marks
▶	64.4000

AVG_MARKS

2. SUM() Function

The total of all numeric values in a group i.e. Calculates the total sum of values in a numeric column.

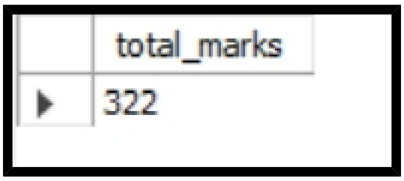
Syntax

```
SELECT SUM(column_name) FROM table_name;
```

Example Query

```
SELECT SUM(marks) AS total_marks FROM studentDetails;
```

Output



```
total_marks
322
```

Sum of marks

3. Count() Function

The number of cell locations in a range that contain a numeric character i.e Counts the number of rows in a result set or the number of non-null values in a column.

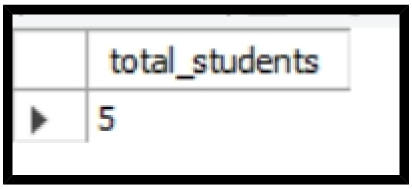
Syntax

```
SELECT COUNT(*) FROM table_name;
SELECT COUNT(column_name) FROM table_name;
```

Example Query

```
SELECT COUNT(studentID) AS total_students FROM studentDetails;
```

Output



```
total_students
5
```

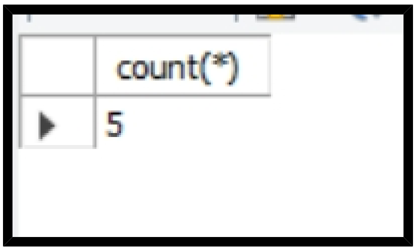
Count of Student

Example Query

```
SELECT COUNT(*) FROM studentdetails;
```

Output

Return the count of rows that meet a specified condition .



```
count(*)
5
```

count all rows

4. Max() Function

Returns the highest numeric value in a group of numbers.

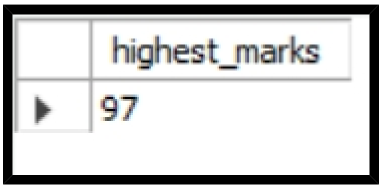
Syntax

```
SELECT MAX(column_name) FROM table_name;
```

Example Query

```
SELECT MAX(marks) AS highest_marks FROM studentDetails;
```

Output



	highest_marks
▶	97

Maximum marks

5. MIN() Function

Returns the lowest numeric value in a group of numbers.

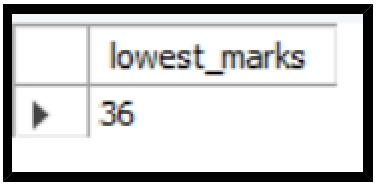
Syntax

```
SELECT MIN(column_name) FROM table_name;
```

Example Query

```
SELECT MIN(marks) AS lowest_marks FROM studentDetails;
```

Output



	lowest_marks
▶	36

Minimum marks

6. VAR() / VARIANCE() Function

Calculates the population variance of a numeric column.

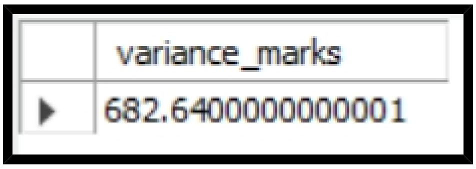
Syntax

```
SELECT VAR(column_name) FROM table_name;
```

Example Query

```
SELECT VARIANCE(marks) AS variance_marks FROM studentDetails;
```

Output



	variance_marks
▶	682.6400000000001

Variance marks

7. STDDEV() / STDDEV_POP() Function

The standard deviation for a group of numbers based on a sample.

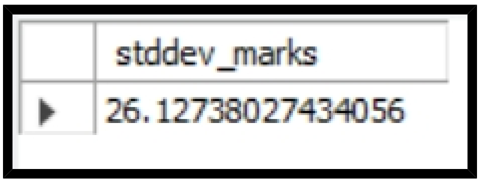
Syntax

```
SELECT STDDEV(column_name) FROM table_name;
```

Example Query

```
SELECT STDDEV(marks) AS stddev_marks FROM studentDetails;
```

Output



	stddev_marks
▶	26.12738027434056

Standrad deviation for marks

8. PERCENTILE_CONT() Function

Calculates a specified percentile value for a numeric column.

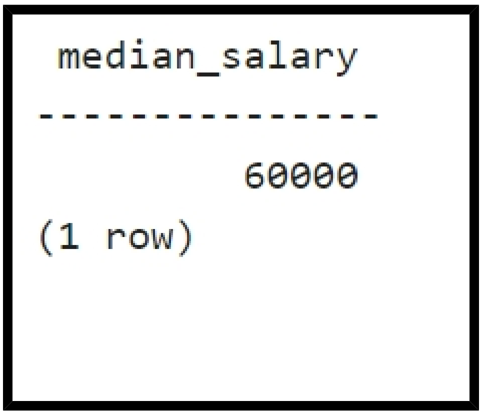
Syntax

```
SELECT PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY column_name) FROM table_name;
```

Example Query

```
SELECT PERCENTILE_CONT(0.5) WITHIN GROUP (ORDER BY salary) AS median_salary FROM employees;
```

Output



```
 median_salary
---------------
         60000
(1 row)
```

Median salary of employee's

9. CORR() Function

Calculates the correlation coefficient between two numeric columns.

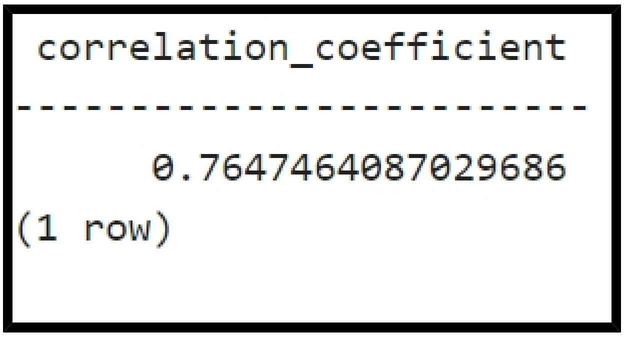
Syntax

```
SELECT CORR(column1, column2) FROM table_name;
```

Example Query

```
SELECT CORR(sales, profit) AS correlation_coefficient FROM sales_data;
```

Output



```
 correlation_coefficient
-------------------------
      0.7647464087029686
(1 row)
```

correlation coefficient between 'sales' and 'profit'

10 .COVAR_POP() Function

Calculates the population covariance between two numeric columns.

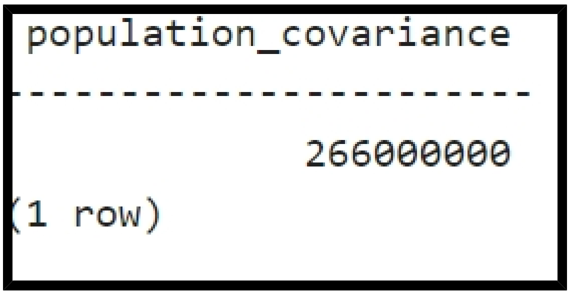
Syntax

```
SELECT COVAR_POP(column1, column2) FROM table_name;
```

Example Query

```
SELECT COVAR_POP(revenue, expenses) AS population_covariance FROM
financial_data;
```

Output



```
 population_covariance
-----------------------
            266000000
(1 row)
```

Population Covariance between revenue and expenses

Conclusion

In SQL, **statistical functions** help to analyze and summarise data in the database. These functions assist in extracting meaningful information from the given datasets. For determining the number of occurrences , calculating totals , finding averages or calculating the variance in the dataset statistical functions plays a vital role .Overall, the integration of Statistical Functions elevates SQL’s capabilities, making it an invaluable asset for businesses and analysts seeking actionable intelligence from their relational databases.