# what is SQL FUNCTIONS

*SQL  functions are built in operations  provided by the databases to perform calculations,*

*manipulate data,format data and return specific values based on the input..*

*They are often used in:*

*SELECT queries*

*WHERE conditions.*

*GROUP BY and HAVING clauses.*

*ORDER BY clauses*

## 1. STRING FUNCTIONS

These functions are used to perform operations on character/string data types. They help in modifying, analyzing, or formatting string values.

### 1.1 UPPER()

**Definition:**

The UPPER() function is used to convert all characters in a string to uppercase letters. It is useful when we want to compare or display text in a uniform format.

Syntax:

SELECT UPPER(column_name or string);

SELECT UPPER('arjun');

-- Output: 'ARJUN'

### 1.2 LOWER()

**Definition:**

The LOWER() function converts all characters in a string to lowercase letters. It is used when you want case-insensitive comparisons or formatting.

Syntax:

SELECT LOWER(column_name or string);

SELECT LOWER('HELLO');

-- Output: 'hello'

### 1.3 LENGTH()

**Definition:**

The LENGTH() function returns the total number of characters in a string, including spaces and special characters. It's useful for validations and data analysis.

Syntax:

SELECT LENGTH(string);

Example:

SELECT LENGTH('Python Rocks');

-- Output: 12

### 1.4 SUBSTRING() or SUBSTR()

**Definition:**

The SUBSTRING() function extracts a specific portion of a string, starting from a defined position and length. It is helpful when you need part of a string like a first name, ID, etc.

Syntax:

SELECT SUBSTRING(string, start_position, length);

Example:

SELECT SUBSTRING('Database', 1, 4);

-- Output: 'Data'

## 1.5 CONCAT()

**Definition:**

The CONCAT() function is used to join two or more strings together into one string. It's useful in combining first name and last name, or adding labels.

Syntax:

SELECT CONCAT(string1, string2);

Example:

SELECT CONCAT('Hello', 'World');

-- Output: 'HelloWorld'

## 1.6 TRIM()

**Definition:**

The TRIM() function removes leading and trailing spaces from a string. It's commonly used in cleaning user input or preparing data for comparison.

Syntax:

```
SELECT TRIM(string);
```

Example:

```
SELECT TRIM('   SQL   ');
-- Output: 'SQL'
```

## 1.7 REPLACE()

**Definition:**

The REPLACE() function replaces all occurrences of a substring within a string with a new substring. It's helpful when correcting or updating text in a database.

Syntax:

```
SELECT REPLACE(original_string, search_string, replace_with);
```

Example:

```
SELECT REPLACE('I like Java', 'Java', 'Python');
-- Output: 'I like Python'
```

## 2. NUMERIC FUNCTIONS

These functions are used to perform mathematical calculations and are mostly used on numeric columns.

### 2.1 ROUND()

**Definition:**

The ROUND() function rounds a numeric value to the nearest whole number or to the specified decimal places. It's used in financial calculations and reports.

Syntax:

```
SELECT ROUND(number, decimal_places);
```

Example:

```
SELECT ROUND(123.4567, 2);
-- Output: 123.46
```

### 2.2 FLOOR()

**Definition:**

The FLOOR() function returns the largest integer less than or equal to the given number. It always rounds down.

Syntax:

SELECT FLOOR(number);

Example:

SELECT FLOOR(5.9);

-- Output: 5

## 2.3 CEIL() or CEILING()

**Definition:**

The CEIL() function returns the smallest integer greater than or equal to the given number. It always rounds up.

Syntax:

SELECT CEIL(number);

Example:

SELECT CEIL(5.1);

-- Output: 6

## 2.4 MOD()

**Definition:**

The MOD() function returns the remainder of division between two numbers. It's helpful in even/odd checks, grouping, and logic conditions.

Syntax:

```
SELECT MOD(dividend, divisor);
```

Example:

```
SELECT MOD(10, 3);
-- Output: 1
```

## 3. DATE FUNCTIONS

These functions are used to process and manipulate date and time values.

### 3.1 CURRENT_DATE / CURDATE()

**Definition:**

This function returns the current date according to the system clock.

*Syntax:*

```
SELECT CURRENT_DATE;
```

Example:

```
SELECT CURRENT_DATE;
-- Output: '2025-04-21'
```

### *3.2 NOW() / CURRENT_TIMESTAMP*

**Definition:**

This function returns the current date and time.

*Syntax:*

```
SELECT NOW();
```

Example:

```
SELECT NOW();
-- Output: '2025-04-21 11:15:00'
```

### 3.3 YEAR(), MONTH(), DAY()

**Definition:**

These functions extract the year, month, or day from a date.

Syntax:

```
SELECT YEAR('2025-04-21');
SELECT MONTH('2025-04-21');
SELECT DAY('2025-04-21');
```

Output:

2025, 4, 21

### 3.4 DATEDIFF()

***Definition:***

Returns the difference in days between two dates.

Syntax:

```
SELECT DATEDIFF(date1, date2);
```

Example:

```
SELECT DATEDIFF('2025-05-01', '2025-04-21');
-- Output: 10
```

## 4. CONVERSION FUNCTIONS

Used to convert one data type to another.

### 4.1 CAST()

Definition:

The CAST() function is used to explicitly convert data from one type to another (e.g., string to int, float to int, etc.).

Syntax:

SELECT CAST(value AS data_type);

Example:

SELECT CAST('123' AS INT);

-- Output: 123

## 5. AGGREGATE FUNCTIONS (GROUP FUNCTIONS)

These functions perform calculations on a group of rows and return a single value. Commonly used with GROUP BY.

### 5.1 COUNT()

Definition:

Returns the total number of rows in a table or matching a condition.

Syntax:

SELECT COUNT(*) FROM table_name;

Example:

SELECT COUNT(*) FROM employees;

-- Output: total number of employees

## 5.2 SUM()

Definition:

Returns the total sum of values in a numeric column.

Syntax:

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

Example:

```
SELECT SUM(salary) FROM employees;
-- Output: total salary
```

## 5.3 AVG()

Definition:

Returns the average value of a numeric column.

Syntax:

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

Example:

```
SELECT AVG(marks) FROM students;
-- Output: average marks
```

## 5.4 MIN()

Definition:

Returns the minimum value in a column.

Syntax:

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

Example:

```
SELECT MIN(age) FROM users;
-- Output: youngest age
```

## 5.5 MAX()

**Definition:**

Returns the maximum value in a column.

Syntax:

```
SELECT MAX(price) FROM products;
-- Output: highest price
```