SQL Functions: An In-Depth Guide

SQL functions are essential tools that allow you to perform operations on data, transform it, and retrieve meaningful insights. They can be categorized into different types, each serving specific purposes. Below is a detailed guide to the various SQL functions:

1. Aggregate Functions

Aggregate functions perform a calculation on a set of values and return a single value. Common aggregate functions include:

COUNT()

Returns the number of rows that match a specified condition.

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

SUM()

Adds up all the values in a numeric column.

```
SELECT SUM(Salary) FROM Employees;
```

AVG()

Calculates the average value of a numeric column.

```
SELECT AVG(Salary) FROM Employees;
```

MIN()

Retrieves the minimum value from a column.

```
SELECT MIN(Salary) FROM Employees;
```

MAX()

Retrieves the maximum value from a column.

```
SELECT MAX(Salary) FROM Employees;
```

2. String Functions

String functions are used to manipulate and query string values. Key string functions include:

LENGTH()

Returns the length of a string.

```
SELECT LENGTH('SQL Functions');
```

UPPER()

Converts a string to uppercase.

```
SELECT UPPER('sql functions');
```

LOWER()

Converts a string to lowercase.

```
SELECT LOWER('SQL FUNCTIONS');
```

SUBSTRING()

Extracts a part of a string.

```
SELECT SUBSTRING('SQL Functions', 5, 9);
```

CONCAT()

Concatenates two or more strings.

```
SELECT CONCAT('SQL', ' ', 'Functions');
```

TRIM()

Removes leading and trailing spaces from a string.

```
SELECT TRIM('   SQL Functions   ');
```

3. Date and Time Functions

Date and time functions are used to manipulate and retrieve date and time values. Examples include:

NOW()

Returns the current date and time.

```
SELECT NOW();
```

CURDATE()

Returns the current date.

```
SELECT CURDATE();
```

DATEDIFF()

Calculates the difference in days between two dates.

```
SELECT DATEDIFF('2024-12-31', '2024-01-01');
```

DATE_ADD()

Adds a time/date interval to a date and returns the new date.

```
SELECT DATE_ADD('2024-01-01', INTERVAL 10 DAY);
```

DATE_FORMAT()

Formats a date according to a specified format.

```
SELECT DATE_FORMAT('2024-07-11', '%W, %M %d, %Y');
```

4. Numeric Functions

Numeric functions are used to perform operations on numeric data. Key functions include:

ROUND()

Rounds a number to a specified number of decimal places.

```
SELECT ROUND(123.4567, 2);
```

FLOOR()

Returns the largest integer less than or equal to a number.

```
SELECT FLOOR(123.4567);
```

CEIL()

Returns the smallest integer greater than or equal to a number.

```
SELECT CEIL(123.4567);
```

ABS()

Returns the absolute value of a number.

```
SELECT ABS(-123.4567);
```

MOD()

Returns the remainder of a division operation.

```
SELECT MOD(10, 3);
```

5. Conditional Functions

Conditional functions allow you to implement conditional logic in your queries. Examples include:

CASE

Performs conditional logic and returns different results based on conditions.

```
SELECT
  CASE
    WHEN Salary > 50000 THEN 'High'
    WHEN Salary BETWEEN 30000 AND 50000 THEN 'Medium'
    ELSE 'Low'
  END AS SalaryCategory
FROM Employees;
```

COALESCE()

Returns the first non-null value in a list.

```
SELECT COALESCE(NULL, NULL, 'SQL Functions', 'Backup');
```

NULLIF()

Returns NULL if two expressions are equal; otherwise, it returns the first expression.

```
SELECT NULLIF(Salary, 0) FROM Employees;
```

Usage Examples

Example 1: Aggregating Data

```
SELECT Department, COUNT(*) AS NumberOfEmployees, AVG(Salary) AS AverageSalary
FROM Employees
GROUP BY Department;
```

Example 2: String Manipulation

```
SELECT
  CONCAT(UPPER(FirstName), ' ', UPPER(LastName)) AS FullName,
  LENGTH(FirstName) + LENGTH(LastName) AS TotalLength
FROM Employees;
```

Example 3: Date Calculation

```
SELECT
  EmployeeName,
  DATE_ADD(HireDate, INTERVAL 5 YEAR) AS FiveYearAnniversary
FROM Employees;
```

Example 4: Conditional Logic

```
SELECT
  EmployeeName,
  CASE
    WHEN PerformanceRating >= 4 THEN 'Excellent'
    WHEN PerformanceRating = 3 THEN 'Good'
    ELSE 'Needs Improvement'
  END AS PerformanceCategory
FROM Employees;
```