

Day27_SQL_Functions

June 24, 2025

SQL Useful Functions

Today, we are learning about **SQL built-in functions** that simplify data processing and manipulation in MySQL.

Functions are essential for working with data — from performing calculations to string formatting.

1. **Numeric & Math Functions** Perform operations like rounding, roots, and calculations.
 2. **Bitwise & Conversion Functions** Work with binary, octal, and base conversions.
 3. **Trigonometric Functions** Use built-in math for angles and radians.
 4. **String Functions** Manipulate and extract information from text.
-

1 Start Logging Output

```
mysql> tee C:/Users/Public/sql_useful_functions.txt;
```

2 Create Database and Table

```
mysql> CREATE DATABASE SQLFunctionsDemo;  
Query OK, 1 row affected (0.38 sec)
```

```
mysql> USE SQLFunctionsDemo;  
Database changed
```

```
mysql> CREATE TABLE Dummy (  
-> id INT AUTO_INCREMENT PRIMARY KEY,  
-> name VARCHAR(50),  
-> marks INT,  
-> price DECIMAL(10,2),  
-> comment TEXT  
-> );  
Query OK, 0 rows affected (1.77 sec)
```

3 Insert Sample Records

```
mysql> INSERT INTO Dummy (name, marks, price, comment) VALUES
-> ('Akshay', 85, 199.99, 'Nice work'),
-> ('Swara', 92, 149.49, 'Top scorer'),
-> ('Ravi', 76, 175.00, 'Average performance'),
-> ('Neha', 89, 160.55, 'Excellent'),
-> ('Amit', 66, 120.00, 'Needs improvement');
Query OK, 5 rows affected (0.80 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

4 Numeric & Math Functions

```
mysql> SELECT ABS(-45); -- Absolute
+-----+
| ABS(-45) |
+-----+
|         45 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT CEIL(4.3); -- Round up
+-----+
| CEIL(4.3) |
+-----+
|          5 |
+-----+
1 row in set (0.02 sec)
```

```
mysql> SELECT FLOOR(4.7); -- Round down
+-----+
| FLOOR(4.7) |
+-----+
|          4 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT ROUND(123.456, 2); -- Round to 2 decimals
+-----+
| ROUND(123.456, 2) |
+-----+
|          123.46 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT MOD(10, 3); -- Remainder
+-----+
```

```

| MOD(10, 3) |
+-----+
|          1 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT POWER(2, 3);  -- 2^3
+-----+
| POWER(2, 3) |
+-----+
|          8 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SQRT(49);  -- Square root
+-----+
| SQRT(49) |
+-----+
|          7 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT PI();  -- Pi constant
+-----+
| PI()      |
+-----+
| 3.141593 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT EXP(2);  -- e^2
+-----+
| EXP(2)      |
+-----+
| 7.38905609893065 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LOG10(1000);  -- Base-10 log
+-----+
| LOG10(1000) |
+-----+
|          3 |
+-----+
1 row in set (0.00 sec)

```

5 Bitwise & Conversion Functions

```
mysql> SELECT BIT_COUNT(7); -- Binary of 7 is 111 → 3 ones
```

```
+-----+
| BIT_COUNT(7) |
+-----+
|           3 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT BIN(10); -- Decimal to binary
```

```
+-----+
| BIN(10) |
+-----+
| 1010    |
+-----+
1 row in set (0.02 sec)
```

```
mysql> SELECT OCT(10); -- Decimal to octal
```

```
+-----+
| OCT(10) |
+-----+
| 12      |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT CONV(8, 10, 2); -- Convert 8 from base 10 to base 2
```

```
+-----+
| CONV(8, 10, 2) |
+-----+
| 1000           |
+-----+
1 row in set (0.00 sec)
```

6 Trigonometric Functions

```
mysql> SELECT SIN(PI()/2);
```

```
+-----+
| SIN(PI()/2) |
+-----+
|           1 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT COS(0);
```

```
+-----+
| COS(0) |
```

```
+-----+
|      1 |
+-----+
1 row in set (0.01 sec)
```

```
mysql> SELECT TAN(PI()/4);
+-----+
| TAN(PI()/4) |
+-----+
| 0.9999999999999999 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT RADIANS(180);
+-----+
| RADIANS(180) |
+-----+
| 3.141592653589793 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT DEGREES(PI());
+-----+
| DEGREES(PI()) |
+-----+
|      180 |
+-----+
1 row in set (0.00 sec)
```

7 String Functions

```
mysql> SELECT LENGTH('Akshay'); -- Bytes
+-----+
| LENGTH('Akshay') |
+-----+
|      6 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT CHAR_LENGTH('Akshay'); -- Characters
+-----+
| CHAR_LENGTH('Akshay') |
+-----+
|      6 |
+-----+
1 row in set (0.02 sec)
```

```

mysql> SELECT CONCAT('SQL', ' Functions'); -- Combine
+-----+
| CONCAT('SQL', ' Functions') |
+-----+
| SQL Functions                |
+-----+
1 row in set (0.00 sec)

mysql> SELECT UPPER('akshay'); -- Uppercase
+-----+
| UPPER('akshay') |
+-----+
| AKSHAY          |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LOWER('SQL'); -- Lowercase
+-----+
| LOWER('SQL') |
+-----+
| sql          |
+-----+
1 row in set (0.00 sec)

mysql> SELECT TRIM(' Akshay '); -- Remove spaces
+-----+
| TRIM(' Akshay ') |
+-----+
| Akshay           |
+-----+
1 row in set (0.00 sec)

mysql> SELECT REPLACE('Hi Akshay', 'Akshay', 'Swara'); -- Replace
+-----+
| REPLACE('Hi Akshay', 'Akshay', 'Swara') |
+-----+
| Hi Swara                                |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SUBSTRING('Akshay', 2, 3); -- 'ksh'
+-----+
| SUBSTRING('Akshay', 2, 3) |
+-----+
| ksh                        |
+-----+
1 row in set (0.00 sec)

```

```

mysql> SELECT REVERSE('SQL'); -- 'LQS'
+-----+
| REVERSE('SQL') |
+-----+
| LQS           |
+-----+
1 row in set (0.00 sec)

mysql> SELECT INSTR('Database', 'base'); -- Position
+-----+
| INSTR('Database', 'base') |
+-----+
|                          5 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LEFT('Akshay', 3); -- 'Aks'
+-----+
| LEFT('Akshay', 3) |
+-----+
| Aks              |
+-----+
1 row in set (0.00 sec)

mysql> SELECT RIGHT('Akshay', 3); -- 'hay'
+-----+
| RIGHT('Akshay', 3) |
+-----+
| hay                |
+-----+
1 row in set (0.00 sec)

mysql> SELECT REPEAT('SQL', 3); -- 'SQLSQLSQL'
+-----+
| REPEAT('SQL', 3) |
+-----+
| SQLSQLSQL        |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SPACE(5); -- 5 spaces
+-----+
| SPACE(5) |
+-----+
|          |
+-----+
1 row in set (0.00 sec)

```

8 Using Functions with Table Columns

-- 1. Absolute difference from 90 marks

```
mysql> SELECT name, marks, ABS(marks - 90) AS MarksDifference FROM Dummy;
```

name	marks	MarksDifference
Akshay	85	5
Swara	92	2
Ravi	76	14
Neha	89	1
Amit	66	24

-- 2. Round off price to nearest integer

```
mysql> SELECT name, price, ROUND(price, 0) AS RoundedPrice FROM Dummy;
```

name	price	RoundedPrice
Akshay	199.99	200
Swara	149.49	149
Ravi	175.00	175
Neha	160.55	161
Amit	120.00	120

-- 3. Convert names to uppercase

```
mysql> SELECT name, UPPER(name) AS NameUpper FROM Dummy;
```

name	NameUpper
Akshay	AKSHAY
Swara	SWARA
Ravi	RAVI
Neha	NEHA
Amit	AMIT

-- 4. Extract first 10 characters from comment

```
mysql> SELECT name, comment, SUBSTRING(comment, 1, 10) AS ShortComment FROM Dummy;
```

name	comment	ShortComment
Akshay	Nice work	Nice work
Swara	Top scorer	Top scorer
Ravi	Average performance	Average pe
Neha	Excellent	Excellent
Amit	Needs improvement	Needs impr

+-----+-----+-----+

9 End Logging

```
mysql> notee;
```