Scalar Functions in SQL

We can perform different operations on the data stored in the SQL database. This can be done with the help of built-in functions provided by SQL. *Scalar functions are the built-in functions in SQL, and whatever be the input provided to the scalar functions, the output returned by these functions will always be a single value.*

In SQL, each record is operated independently by the scalar functions.

Some of the commonly used scalar functions in SQL includes:

1. **UCASE()**

2. **LCASE()**

3. **MID()**

4. **LENGTH()**

5. **ROUND()**

6. **NOW()**

7. **FORMAT()**

Let us see each of them in more detail with the help of examples. We will use the MySQL database for writing all the queries.

Consider we have customers table with the following records:

ID	NAME	AGE	ADDRESS	SALARY
1	Himani Gupta	21	Modinagar	22000
2	Shiva Tiwari	22	Bhopal	21000
3	Ajeet Bhargav	45	Meerut	65000
4	Ritesh Yadav	36	Azamgarh	26000
5	Balwant Singh	45	Varanasi	36000
6	Mahesh Sharma	26	Mathura	22000
7	Rohit Shrivastav	19	Ahemdabad	38000
8	Neeru Sharma	29	Pune	40000

9	Aakash Yadav	32	Mumbai	43500
10	Sahil Sheikh	35	Aurangabad	68800

1. UCASE()

- The UCASE() or upper case function will change the case of the string to upper case characters.

- Here, the string can also be passed as a parameter or the table's column name, which contains the string, can also be passed as a parameter to the UCASE () function.

Syntax to convert a string in uppercase:

```
SELECT UCASE (String);
```

OR

Syntax to convert a string in uppercase, which is stored in a table's column:

SELECT UCASE (ColumnName) FROM TableName;

Example 1:

```
mysql> SELECT UCASE ("Welcome to Javatpoint") AS UpperCase_String;
```

Here, the UCASE () function with the string 'Welcome to Javatpoint' as a parameter is used in a SELECT query to print the string in the upper case. 'Upper_Case_String' is an alias.

You will get the following output:

UpperCase_String
WELCOME TO JAVATPOINT

'WELCOME TO JAVATPOINT' is the upper case string for the parameter passed to the UCASE () function. Hence, it is displayed as an output.

Example 2:

```
mysql> SELECT ID, Name, Age, UCASE (Address) AS Address_UpperCase, Salary FROM customer
```

Here, the 'Address' column is passed as a parameter to the UCASE () function in a SELECT query to print the values of the address column in the upper case. 'Upper_Case_String' is an alias.

You will get the following output:

ID	Name	Age	Address_UpperCase	Salary
1	Himani Gupta	21	MODINAGAR	22000
2	Shiva Tiwari	22	BHOPAL	21000
3	Ajeet Bhargav	45	MEERUT	65000
4	Ritesh Yadav	36	AZAMGARH	26000
5	Balwant Singh	45	VARANASI	36000
6	Mahesh Sharma	26	MATHURA	22000
7	Rohit Shrivastav	19	AHEMDABAD	38000
8	Neeru Sharma	29	PUNE	40000
9	Aakash Yadav	32	MUMBAI	43500
10	Sahil Sheikh	35	AURANGABAD	68800

All the column values present in ID, Name, Age and salary are displayed as it is. The column values present in the address column are displayed in the upper case.
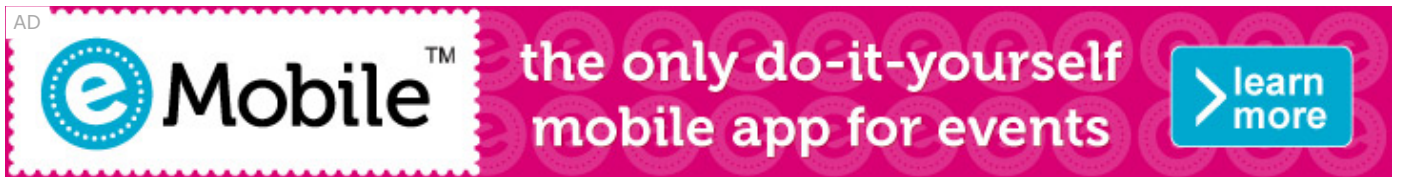
2. LCASE()

- The lower case function will change the case of the string to lower case characters.

- Here, the string can also be passed as a parameter or the table's column name, which contains the string, can also be passed as a parameter to the LCASE () function.

Syntax to convert a string in lowercase:

```
SELECT LCASE (String);
```

OR

Syntax to convert a string in lowercase which is stored in a table's column:

```
SELECT LCASE (ColumnName) FROM TableName;
```
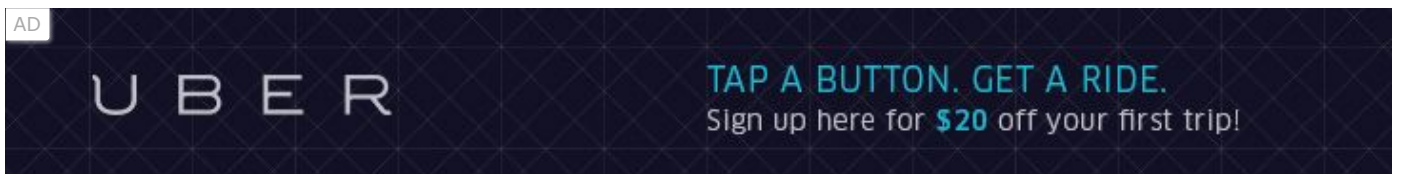
Example 1:

```
mysql> SELECT LCASE ("WELCOME TO JAVATPOINT") AS LowerCase_String;
```

Here, the LCASE () function with the string 'WELCOME TO JAVATPOINT' as a parameter is used in a SELECT query to print the string in lower case. 'LowerCase_String' is an alias.

You will get the following output:

LowerCase_String
welcome to javatpoint

'welcome to javatpoint' is the lower case string for the parameter passed to the LCASE () function. Hence, it is displayed as an output.

Example 2:

```
mysql> SELECT ID, LCASE (Name) AS NameLowerCase, Age, Address, Salary FROM customers;
```

Here, the 'Name' column is passed as a parameter to the LCASE () function in a SELECT query to print the values of the 'Name' column in lower case. 'NameLowerCase' is an alias.

You will get the following output:

ID	NameLowerCase	Age	Address	Salary
1	himani gupta	21	Modinagar	22000
2	shiva tiwari	21	Bhopal	21000
3	ajeet bhargav	45	Meerut	65000
4	ritesh yadav	36	Azamgarh	26000
5	balwant singh	45	Varanasi	36000
6	mahesh sharma	26	Mathura	22000
7	rohit shrivastav	19	Ahemdabad	38000
8	neeru sharma	29	Pune	40000
9	aakash yadav	32	Mumbai	43500
10	sahil sheikh	35	Aurangabad	68800

All the column values present in ID, Age, Address and salary are displayed as it is. The column values present in the 'Name' column are displayed in lower case.

3. MID()

- MID () function is used to extract substrings from the table's column, which contain values of string type.

Syntax to extract a substring:

```
SELECT MID (string, start, length);
```

where,

String: contains the entire string from which we want to extract a specific portion of the string.

Starting index: The integer value present in the starting index decides where the characters in the string are to be extracted from the beginning or end of the string.

Length: The integer value present in the length field tells the count of characters to be extracted from the string. If the length is not specified, then the entire string will be extracted.

> Note: Length is an optional parameter

OR

Syntax to extract a substring which is stored in a table's column:

SELECT MID (ColumnName, start, length) **FROM** TableName;

Example 1:

mysql> **SELECT** MID ("Welcome to the world of databases", 16, 18) **AS** Substring;

Here, the MID () function with the three parameters: 'Welcome to the world of databases', 16, 18 is passed as a parameter to the SELECT query to print the substring starting with the index 16 and contains 18 characters. 'Substring' is an alias.

You will get the following output:

Substring
world of databases

'World of databases' is the substring for the parameter passed to the MID () function. Hence, it is displayed as an output.

Example 2:

mysql> **SELECT** ID, MID (**Name**, 3, 5) **AS** NameSubstring, Age, Address, Salary **FROM** customers;

Here, the MID () function with the three parameters: Name, 3, 5 is passed as a parameter to the SELECT query to print the substring starting with the index 3 and contains 5 characters. ' NameSubstring ' is an alias.

You will get the following output:

ID	NameMid	Age	Address	Salary

1	mani	21	Modinagar	22000
2	iva T	21	Bhopal	21000
3	eet B	45	Meerut	65000
4	tesh	36	Azamgarh	26000
5	lwant	45	Varanasi	36000
6	hesh	26	Mathura	22000
7	hit S	19	Ahemdabad	38000
8	eru S	29	Pune	40000
9	kash	32	Mumbai	43500
10	hil S	35	Aurangabad	68800

All the column values present in ID, Age, Address and salary are displayed as it is. Only the substrings starting with the index 3 having 5 characters are displayed in the' Name' column.
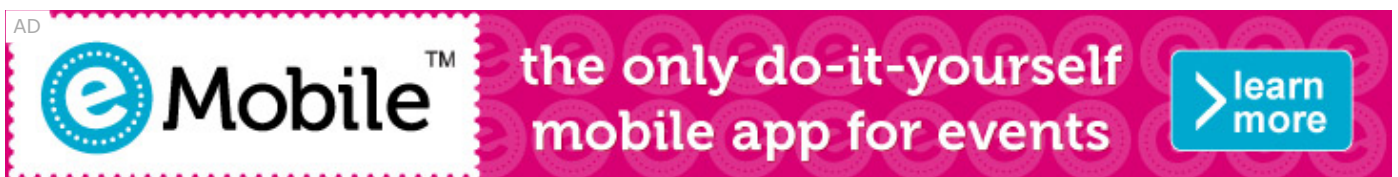
4. LENGTH()

- LENGTH () function returns the length of the string in the column.

Syntax to find the length of a string:

```
SELECT LENGTH (String);
```

OR

Syntax to find the length of a string from the table's column:

```
SELECT LENGTH (ColumnName) FROM TableName;
```

Example 1:

mysql> SELECT LENGTH ("Welcome to the world of databases") AS LengthOfString;

Here, the LENGTH () function with the string 'Welcome to the world of databases' as a parameter is used in a SELECT query to print the length of the string. 'LengthOfString' is an alias.

You will get the following output:

LengthOfString
33

'33' is the length of the string for the parameter passed to the LENGTH () function. Hence, it is displayed as an output.

Example 2:

mysql> **SELECT** ID, LENGTH (**Name**) **AS** LengthOfName, Age, Address, Salary **FROM** customers;

Here, the 'Name' column is passed as a parameter to the LENGTH () function in a SELECT query to print the length of the values of the 'Name' column. 'LengthOfName' is an alias.

You will get the following output:

ID	LengthOfName	Age	Address	Salary
1	12	21	Modinagar	22000
2	12	21	Bhopal	21000
3	13	45	Meerut	65000
4	12	36	Azamgarh	26000
5	13	45	Varanasi	36000
6	13	26	Mathura	22000
7	16	19	Ahemdabad	38000
8	12	29	Pune	40000
9	12	32	Mumbai	43500

| 10 | 12 | 35 | Aurangabad | 68800 |

All the column values present in ID, Age, Address and salary are displayed as it is. In the 'Name' column, only the length of the string is displayed.

5. ROUND()

The ROUND () function is used to round a numeric column to the number of decimals specified.

Syntax to round the numeric values:

```
SELECT ROUND (NumericValue, Decimals);
```

where,

Decimal represents the number of decimals to be fetched.

OR

Syntax to round the numeric values from the table's column:

```
SELECT ROUND (ColumnName, Decimals) FROM TableName;
```

Example 1:

```
mysql> SELECT ROUND (18000.44444, 2) AS RoundedValue;
```

Here, the ROUND () function with two different parameters containing numeric values 18000.44444, 2 is used in a SELECT query to print the rounded value for the given number. Since 2 is passed as a second parameter, the number passed in the first parameter will be rounded to two decimal points. ' RoundedValue ' is an alias.

You will get the following output:

RoundedValue
18000.44

'18000.44' is the rounded value for the parameter passed to the ROUND () function. Hence, it is displayed as an output.

Example 2:

```
mysql> SELECT ROUND (18000.44444, 0) AS RoundedValue;
```

Here, the ROUND () function with two different parameters containing numeric values 18000.44444, 0 is used in a SELECT query to print the rounded value for the given number. Since 0 is passed as a second parameter, the number passed in the first parameter will be rounded without any decimal points. ' RoundedValue ' is an alias.

You will get the following output:

RoundedValue
18000

'18000' is the rounded value for the parameter passed to the ROUND () function. Hence, it is displayed as an output.

Example 3:

mysql> SELECT ROUND (18000.44444, 3) AS RoundedValue;

Here, the ROUND () function with two different parameters containing numeric values 18000.44444, 3 is used in a SELECT query to print the rounded value for the given number. Since 3 is passed as a second parameter, the number passed in the first parameter will be rounded to three decimal points. ' RoundedValue ' is an alias.

You will get the following output:

RoundedValue
18000.444

'18000.444' is the rounded value for the parameter passed to the ROUND () function. Hence, it is displayed as an output.

6. NOW()

- ○ NOW () function returns the current system' date and time.

Syntax to find the current date and time:

SELECT NOW ();

OR

Syntax to find the current date and time from a table's column:

SELECT NOW () **FROM** TableName;

Example 1:

mysql> **SELECT** NOW () **AS** CurrentDatenTime;

The NOW () function is used in a SELECT query to print the current system date and time. 'CurrentDatenTime' is an alias.

You will get the following output:

CurrentDatenTime
2021-09-13 10:38:24

'2021-09-13 10:38:24' is the current date and time of the system. Hence, it is displayed as an output.

Example 2:

mysql> **SELECT** ID, **Name**, Age, Address, Salary, NOW () **AS** 'DatenTime' **FROM** customers;

Here, the NOW() function is used in a SELECT query to print the current system's date and time in a table's column. 'DatenTime' is an alias.

You will get the following output:

ID	Name	Age	Address	Salary	DatenTime
1	Himani Gupta	21	Modinagar	22000	2021-09-13 20:45:56
2	Shiva Tiwari	22	Bhopal	21000	2021-09-13 20:45:56
3	Ajeet Bhargav	45	Meerut	65000	2021-09-13 20:45:56
4	Ritesh Yadav	36	Azamgarh	26000	2021-09-13 20:45:56
5	Balwant Singh	45	Varanasi	36000	2021-09-13 20:45:56
6	Mahesh Sharma	26	Mathura	22000	2021-09-13 20:45:56
7	Rohit Shrivastav	19	Ahemdabad	38000	2021-09-13 20:45:56
8	Neeru Sharma	29	Pune	40000	2021-09-13 20:45:56
9	Aakash Yadav	32	Mumbai	43500	2021-09-13 20:45:56
10	Sahil Sheikh	35	Aurangabad	68800	2021-09-13 20:45:56

All the column values present in ID, Name, Age, Address and salary are displayed as it is. In the 'DatenTime' column, the current system date and time is displayed.

7. FORMAT()

- The FORMAT () function is used to format how a column is to be displayed.

Syntax to format a value:

```
SELECT FORMAT (Value, Decimal);
```

OR

Syntax to format a value present in table's column:

SELECT FORMAT (ColumnName, Decimal) FROM TableName;

Example 1:

```
mysql> SELECT FORMAT (6789.6789, 2) AS Formatted_Number;
```

The FORMAT () function with two different parameters containing numeric values 6789.6789, 2 is used in a SELECT query to print the formatted value after rounding off the given number. Since 2 is passed as a second parameter, the number passed in the first parameter will be rounded to two decimal points. 'Formatted_Number' is an alias.

You will get the following output:

Formatted_Number
6,789.68

'6,789.68' is the formatted value for the parameter passed to the FORMAT() function. Hence, it is displayed as an output.

Example 2:

```
mysql> SELECT ID, Name, Age, Address, FORMAT (Salary / 0.5, 2) AS FormattedSalary FROM cus
```

The ROUND () function with two different parameters containing the column 'Salary' / 0.5, 2 is used in a SELECT query to print the formatted value. This formatted value will be obtained after the value contained in the 'Salary' column is divided by 2. Further, since 2 is passed as a second parameter, the value obtained after dividing the salary by 0.5 will be rounded to two decimal points. 'FormattedSalary' is an alias.

You will get the following output:

ID	Name	Age	Address	FormattedSalary
1	Himani Gupta	21	Modinagar	44,000.00
2	Shiva Tiwari	22	Bhopal	42,000.00
3	Ajeet Bhargav	45	Meerut	130,000.00
4	Ritesh Yadav	36	Azamgarh	52,000.00
5	Balwant Singh	45	Varanasi	72,000.00
6	Mahesh Sharma	26	Mathura	44,000.00
7	Rohit Shrivastav	19	Ahemdabad	76,000.00
8	Neeru Sharma	29	Pune	80,000.00

| 9 | Aakash Yadav | 32 | Mumbai | 87,000.00 |
| 10 | Sahil Sheikh | 35 | Aurangabad | 137,600.00 |

All the column values present in ID, Name, Age and Address are displayed as it is. In the 'FormattedSalary' column, the formatted salary values are displayed.

← Prev

Next →

For Videos Join Our Youtube Channel: Join Now

Feedback

- Send your Feedback to feedback@javatpoint.com

Help Others, Please Share

f ᵗ ᵖ

Learn Latest Tutorials

Splunk	SPSS	Swagger	Transact-SQL
Tumblr	ReactJS	Regex	Reinforcement Learning
R Programming	RxJS	React Native	Python Design Patterns
Python Pillow	Python Turtle	Keras	

Preparation

| Aptitude | Reasoning | Verbal Ability | Interview Questions |
| Company Questions | | | |

Trending Technologies

| Artificial Intelligence | AWS Tutorial | Selenium tutorial | Cloud Computing |
| Artificial Intelligence | AWS | Selenium | Cloud Computing |

Hadoop tutorial

Hadoop

ReactJS Tutorial

ReactJS

Data Science Tutorial

Data Science

Angular 7 Tutorial

Angular 7

Blockchain Tutorial

Blockchain

Git Tutorial

Git

Machine Learning Tutorial

Machine Learning

DevOps Tutorial

DevOps

B.Tech / MCA

DBMS tutorial

DBMS

Data Structures tutorial

Data Structures

DAA tutorial

DAA

Operating System

Operating System

Computer Network tutorial

Computer Network

Compiler Design tutorial

Compiler Design

Computer Organization and Architecture

Computer Organization

Discrete Mathematics Tutorial

Discrete Mathematics

Ethical Hacking

Ethical Hacking

Computer Graphics Tutorial

Computer Graphics

Software Engineering

Software Engineering

html tutorial

Web Technology

Cyber Security tutorial

Cyber Security

Automata Tutorial

Automata

C Language tutorial

C Programming

C++ tutorial

C++

Java tutorial

Java

.Net Framework tutorial

.Net

Python tutorial

Python

List of Programs

Programs

Control Systems tutorial

Data Mining Tutorial

Control System   Data Mining   Data
                                                 Warehouse
                                                 Tutorial

                                                 Data Warehouse