

SQL Functions

SQL Server, MySQL server, and other commercial database systems offer a host of built-in functions that include mathematical, date, string, and conversion functions, which can be used for both processing and manipulating data within the server environments. These inherent functions provide valuable enhancements/extensions to the SQL language, and are useful programmatic utilities that are available to an SQL developer.

In this article we'll discuss SQL functions, what they are, and how they work.

SQL Mathematical Functions

Most SQL dialects offer a plethora of mathematical functions as added enhancements to the SQL language. In addition to many trigonometric functions, some of the more widely used mathematical functions include ABS, COUNT, MIN/MAX, ROUND, and SUM, which are all outlined below.

ABS:

The ABS() function returns the absolute value of a constant or expression, and returns NULL if the constant or expression is NULL.

As examples,

```
SELECT ABS(2);      /* Returns a value of "2" */
```

```
SELECT ABS(-2);    /* Returns a value of "2" */
```

COUNT:

The COUNT() function returns the number of rows that matches a specified criterion in a Select statement.

The format of the COUNT() function is:

```
SELECT COUNT(ColumnName)  
FROM   Table  
WHERE  Criteria
```

where "ColumnName" is the name of a Column in table "Table", and Criteria is a condition.

For example,

```
SELECT COUNT(EmployeeName)
FROM Employees
WHERE Salary > 100000; /* Returns the count of Employees with Salary > $100k */
```

MIN and MAX:

The MIN() function returns the minimum value in a set of values, whereas the MAX() function returns the maximum value in a set of values.

For example,

```
SELECT MAX(Salary)
FROM Employees; /* Returns the highest salary of all employees in table
Employees */
```

```
SELECT MIN(Salary)
FROM Employees; /* Returns the lowest salary of all employees in table
Employees */
```

ROUND:

The ROUND() function rounds a number to a specified number of decimal places.

The format of the ROUND() function is:

```
ROUND( Number, DecimalPlaces); /* Round Number to # of DecimalPlace */
```

As examples,

```
SELECT ROUND(346.315, 2); /* Returns a value of 346.32 */
```

```
SELECT ROUND (346.314, 2); /* Returns a value of 346.31 */
```

```
SELECT ROUND (346.27863, 3); /* Returns a value of 346.279 */
```

SUM:

The SUM() function calculates the sum of a set of values.

As an example,

```
SELECT SUM(Quantity)
FROM OrderDetails; /* Returns the Sum of Column Quantity from the
OrderDetails Table */
```

SQL Date Functions

Most SQL dialects offer a number of date functions which are added enhancements to the SQL language. Although the names of date functions differ across the different database server implementations, their functionality is identical. In SQL Server, some of the more popular date functions include GETDATE, DATEPART, DATEADD, DATEDIFF, and CONVERT, which are all outlined below.

GETDATE:

The GETDATE() function is a SQL Server function that returns the current date and time. Its MySQL equivalent is the NOW() function.

```
SELECT GETDATE();  /* Returns this system date. As an example,  
                    '2022-03-19 11:22.04' */
```

In the following example, any time a record is inserted into Table Orders, the current date/time will be inserted into the OrderDate column, given that its default is defined by the function GETDATE().

```
CREATE TABLE Orders (  
    OrderId      INT                NOT NULL,  
    ProductName VARCHAR(25)         NOT NULL,  
    OrderDate    DATETIME           NOT NULL DEFAULT GETDATE();
```

DATEPART:

The DATEPART() function returns a specific portion of a date/time string, based on a unit specification. The unit specification can be any portion of the date/time string including Year, Month, Day, Hour, Minute, Seconds, etc. The MySQL equivalent is the EXTRACT() function.

The format of the DATEPART() function is:

```
DATEPART (unit, datestring)
```

For example, using the Table Orders specified above, we can retrieve specific units (year, month, and day) of the OrderDate column for OrderId 7, as follows:

```
SELECT      DATEPART(yyyy, OrderDate),  
            DATEPART(mm, OrderDate),  
            DATEPART(dd, OrderDate)  
FROM Orders  
WHERE      OrderId=7 ;
```

DATEADD:

The DATEADD() function adds (or subtracts using negative units) a specified time interval (based on a specified unit) from a date, where the unit specification can be in Years, Months, Days, Hours, Minutes, Seconds, etc. The MySQL equivalent functions are DATE_ADD() and DATE_SUB().

The format of the DATEADD() function is:

DATEADD(unit, number, date)

For example, here is the SQL code to calculate 30 days beyond the OrderDate in the Orders Table, for OrderId 7:

```
SELECT      DATEADD(day, 30, OrderDate)  
FROM Orders  
WHERE      OrderId = 7;
```

DATEDIFF:

The DATEDIFF() function, similarly named in MySQL, returns the time between two dates. The unit specification for the difference can again be expressed in Years, Months, Days, Hours, Minutes, Seconds, etc.

The format of the DATEDIFF() function is:

DATEDIFF(unit, startdate, enddate)

For example, using the Table Orders example above, to calculate the number of days that has elapsed between the Today and the OrderDate column value, for OrderId 7 would be:

```
SELECT      DATEDIFF(day, GETDATE(), OrderDate)  
FROM Orders  
WHERE      OrderId=7;
```

CONVERT:

The CONVERT() function can be used to display date/time data in different formats. The MySQL equivalent function is the DATE_FORMAT() function.

The format of the CONVERT() function is:

CONVERT(data_type(length), expression, style)

where “style” can be a myriad of numeric output formats (see SQL Server documentation for an exhaustive list).

For example:

```
SELECT GETDATE();  /* Returns this system date. As an example,  
2022-03-19 11:22.04 */
```

```
SELECT CONVERT(VARCHAR(10), GETDATE(), 10);  
/* Returns '03-19-22', since style "10" equates to "mm-dd-yy" */
```

SQL String Functions

Most SQL dialects offer a number of string manipulations functions which are added enhancements to the SQL language. Some of the more prominent SQL Server string functions include CHARINDEX, CONCAT, FORMAT, LEFT, LEN, RIGHT, TRIM, and SUBSTRING. If the string function names (described below) in MySQL differ, their counterpart name is specified.

CHARINDEX:

The CHARINDEX() function searches for a substring in a string, and returns the starting numeric position of the specified substring. If the substring is not found within the string, this function returns a value of “0”. The MySQL equivalent function is named INSTR.

The format of the CHARINDEX() function is:

```
CHARINDEX(substring, string, start)
```

As examples,

```
SELECT CHARINDEX('b', 'Zebra');  /* Returns a value of '3' */
```

```
SELECT CHARINDEX('y', 'Zebra');  /* Returns a value of '0' */
```

CONCAT:

The CONCAT() function adds two or more strings together.

The format of the CONCAT() function is:

```
CONCAT(string1, string2, ..., stringN)
```

As an example,

SELECT CONCAT('Today is ', 'Sunday'); /* Returns the string 'Today is Sunday' */

FORMAT:

The FORMAT() function formats a value with the specified format. It is typically used to format date/time values and numeric values.

The format of the FORMAT() function is:

FORMAT(value, format)

As an example,

SELECT FORMAT(111223333, '##-##-#####'); /* Returns the string '111-22-3333' */

LEFT:

The LEFT() function returns a substring with a specified number of characters from a string (starting from the left; that is position 1).

The format of the LEFT() functions is:

LEFT(string, NumberOfChars)

As an example,

SELECT LEFT("Johanson", 5); /* Returns a substring of 'Johan' */

LEN:

The function LEN() returns the length of a string: The Mysql equivalent function is named CHAR_LENGTH.

The format of the LEN() function is:

LEN(string)

As an example,

SELECT LEN('My dog is named Lily'); /* Returns a value of 20 */

RIGHT:

The RIGHT() function returns a substring with a specified number of characters from a string (starting from the right).

The format of the RIGHT() functions is:

RIGHT(string, NumberOfChars)

As an example,

SELECT RIGHT('Johanson', 5); /* Returns a substring of 'anson' */

TRIM:

The TRIM() function removes both leading and trailing spaces from a string.

The format of the TRIM() function is:

SELECT TRIM(string)

As an example,

SELECT TRIM(' This is a string '); /* Returns the string 'This is a string' */

SUBSTRING:

The SUBSTRING() function returns a substring of characters from a string, given a starting position and a specified length of the required substring.

The format of the SUBSTRING() function is:

SUBSTRING(string, start, length)

As an example,

SELECT SUBSTRING('Mr. Jones', 5, 5); /* Returns a substring of 'Jones' */

SQL Conversion Functions

In addition to assisting date/time columns/values from being converted to different display formats (noted in the SQL Date functions section above), the CONVERT() function can also be used to convert a value of any type into a different specified datatype.

In this context, the format of the CONVERT() function is:

SELECT CONVERT(datatype, value);

As an example,

SELECT CONVERT(VARCHAR(20), 12345.34); /* Returns a string of '12345.34' */

Conclusion

The use of mathematical, date/time, string, and conversion functions, offer many built-in facilities that help augment the SQL language. These powerful functions can be used for both processing and manipulating data in a database server environment.