SQL Functions

SQL has many built-in functions for performing processing on string or numeric data. Following is the list of all useful SQL built-in functions –

Consider a sales table, which is having the following records:

	sales_employee	fiscal_year	sale
1	Bob	2,016	100
2	Bob	2,017	150
3	Bob	2,018	200
4	Alice	2,016	150
5	Alice	2,017	100
6	Alice	2,018	200
7	John	2,016	200
8	John	2,017	150
9	John	2,018	250

SQL MAX Function

The SQL MAX aggregate function allows us to select the highest (maximum) value for a certain column.

Now suppose based on the above table we want to find maximum value of any rows, then we can do it as follows:

```
SELECT MAX (sale) from sales;
```

Results 1

SELECT MAX (sale) from sales

	max
1	250

```
SELECT MAX (fiscal_year) from sales;
```

max
2,018

SQL COUNT Function

The SQL COUNT aggregate function is used to count the number of rows in a database table.

Now suppose based on the above table we want to find count of any columns, then we can do it as follows:

```
SELECT COUNT (fiscal_year) from sales;
```

count
9

We can also apply count with Where condition as follows:

```
SELECT COUNT (sale) from sales
WHERE sale>100;
```

count
7

SQL MIN Function

The SQL MIN aggregate function allows us to select the lowest (minimum) value for a certain column.

```
SELECT MIN (sale) from sales;
```

Results 1

SELECT MIN (sale) from sales

	123 min
1	100

SQL AVG Function

The SQL AVG aggregate function selects the average value for certain table column.

e.g. Below query will display average sale of the fiscal year 2018.

```
SELECT  AVG(sale) from sales where fiscal_year = 2018;
```

Results 1

SELECT AVG(sale) from sales where fiscal_year = 2(

	123 avg
1	216.6666666667

SQL SUM Function

The SQL SUM aggregate function allows selecting the total for a numeric column.

```
SELECT  SUM(sale) from sales;
```

Results 1

SELECT SUM(sale) from sales

	123 sum
1	1,500

```
SELECT  fiscal_year,SUM(sale)
from sales
group by fiscal_year;
```

1

CT fiscal_year,SUM(sale) from sales group

123 fiscal_year	123 sum
2,017	400
2,016	450
2,018	650

Above query will give total sum of sales grouped by fiscal year.

SQL SQRT Functions

This is used to generate a square root of a given number.

```
SELECT  *, SQRT(sale) from sales;
```

sales 1

SELECT *, SQRT(sale) from sales | Enter a SQL expression to filter results (use Ctrl+

	ABC sales_employee	123 fiscal_year	123 sale	123 sqrt
1	Bob	2,016	100	10
2	Bob	2,017	150	12.2474487139
3	Bob	2,018	200	14.1421356237
4	Alice	2,016	150	12.2474487139
5	Alice	2,017	100	10
6	Alice	2,018	200	14.1421356237
7	John	2,016	200	14.1421356237
8	John	2,017	150	12.2474487139
9	John	2,018	250	15.8113883008

SQL CONCAT Function

This is used to concatenate any string inside any SQL command. e.g.

Query 1:

```
SELECT CONCAT(sales_employee,fiscal_year,sale)
from sales;
```

Results 1

SELECT CONCAT(sales_employee,fiscal_year,sale) f | Enter a SQL e

	concat
1	Bob2016100.00
2	Bob2017150.00
3	Bob2018200.00
4	Alice2016150.00
5	Alice2017100.00
6	Alice2018200.00
7	John2016200.00
8	John2017150.00
9	John2018250.00

Query 2:

```
SELECT CONCAT(sales_employee,'_',fiscal_year,'_',sale)
from sales;
```

Results 1

SELECT CONCAT(sales_employee,'_',fiscal_year,'_', | Enter a SQL expressior

	concat
1	Bob_2016_100.00
2	Bob_2017_150.00
3	Bob_2018_200.00
4	Alice_2016_150.0(
5	Alice_2017_100.0(
6	Alice_2018_200.0(
7	John_2016_200.0(
8	John_2017_150.0(
9	John_2018_250.0(

In the above queries we have concatenated all the columns of the sales table.

SQL Numeric Functions

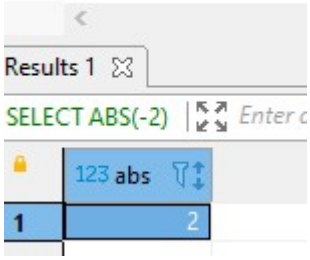
SQL numeric functions consists of complete list of SQL functions required to manipulate numbers in SQL.

We have multiple SQL Numeric functions to tackle and make our work easier. Let's have a look at them one by one.

ABS()

Returns the absolute value of numeric expression.

```
SELECT ABS(-2);
```



ACOS()

Returns the arccosine of numeric expression. Returns NULL if the value is not in the range -1 to 1.

```
SELECT ACOS(0.55);
```

ASIN()

Returns the arcsine of numeric expression. Returns NULL if value is not in the range -1 to 1

```
SELECT ASIN(1);
```

Results 1

SELECT ASIN(1) | Enter a SQ

	123 asin
1	1.5707963268

ATAN()

Returns the arctangent of numeric expression.

```
SELECT ATAN(5);
```

Results 1

SELECT ATAN(5) | Enter a

	123 atan
1	1.3734007669

ATAN2()

Returns the arctangent of the two variables passed to it.

```
SELECT ATAN2(6,1);
```

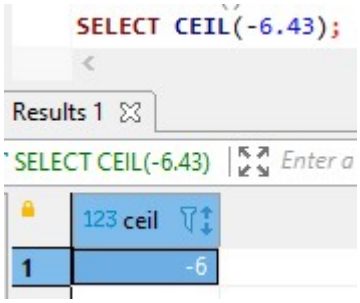
Results 1

SELECT ATAN2(6,1) | Enter

	123 atan2
1	1.4056476494

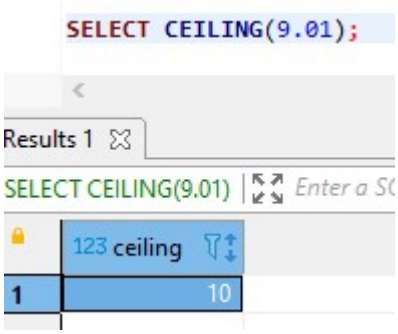
CEIL()

Returns the smallest integer value that is not less than passed numeric expression.



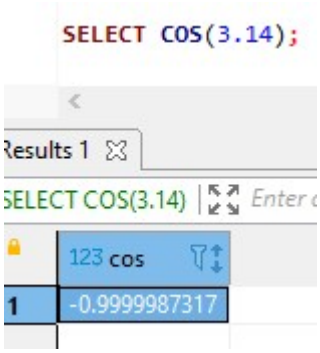
CEILING()

Returns the smallest integer value that is not less than passed numeric expression.



COS()

Returns the cosine of passed numeric expression. The numeric expression should be expressed in radians.



COT()

Returns the cotangent of passed numeric expression.

```
SELECT COT(1);
```

Results 1

SELECT COT(1) | Enter a

	123 cot
1	0.6420926159

DEGREES()

Returns numeric expression converted from radians to degrees.

```
SELECT DEGREES (3.14);
```

Results 1

SELECT DEGREES (3.14) | Enter a

	123 degrees
1	179.9087476711

EXP()

Returns the base of the natural logarithm (e) raised to the power of passed numeric expression.

```
SELECT EXP(0);
```

Results 1

SELECT EXP(0) | Enter a SC

	123 exp
1	1

FLOOR()

Returns the largest integer value that is not greater than passed numeric expression.

```
SELECT FLOOR(-5.42);
```

Results 1

SELECT FLOOR(-5.42)

	123 floor
1	-6

GREATEST()

Returns the largest value of the input expressions.

```
SELECT GREATEST(43,35,21,18,133,919,314,515,647,54);
```

Results 1

SELECT GREATEST(43,35,21,18,133,919,314,515,647,

	123 greatest
1	919

LOG()

Returns the natural logarithm of the passed numeric expression.

```
SELECT LOG(2,16);
```

Results 1

SELECT LOG(2,16)

	123 log
1	4

LOG10()

Returns the base-10 logarithm of the passed numeric expression.

```
SELECT LOG10(100);
```

Results 1

SELECT LOG10(100)

	123 log10
1	2

MOD()

Returns the remainder of one expression by diving by another expression.

```
SELECT MOD(39,4);
```

Results 1

SELECT MOD(39,4)

	123 mod
1	3

PI()

Returns the value of pi.

```
SELECT PI();
```

Results 1

SELECT PI()

	123 pi
1	3.1415926536

POW()

Returns the value of one expression raised to the power of another expression.

POWER()

Returns the value of one expression raised to the power of another expression.

```
SELECT POWER(2,3);
```

Results 1

SELECT POWER(2,3)

	123 power
1	8

RADIANS()

Returns the value of passed expression converted from degrees to radians.

```
SELECT RADIANS(180);
```

Results 1

SELECT RADIANS(180)

	123 radians
1	3.1415926536

ROUND()

Returns numeric expression rounded to an integer. Can be used to round an expression to a number of decimal points.

```
SELECT ROUND(4.49999);
```

Results 1

SELECT ROUND(4.49999)

	123 round
1	4

SIN()

Returns the sine of numeric expression given in radians.

```
SELECT SIN(0);
```

Results 1

SELECT SIN(0)

	123 sin
1	0

SQRT()

Returns the non-negative square root of numeric expression.

```
SELECT SQRT(81);
```

Results 1

SELECT SQRT(81)

	123 sqrt
1	9

SQL String Functions

Complete list of SQL functions required to manipulate strings in SQL.

We have multiple SQL String functions to tackle and make our work easier. Let's have a look at them one by one.

LEFT()

Returns the leftmost number of characters as specified.

```
SELECT LEFT('postgresqltutorial', 5);
```

Results 1

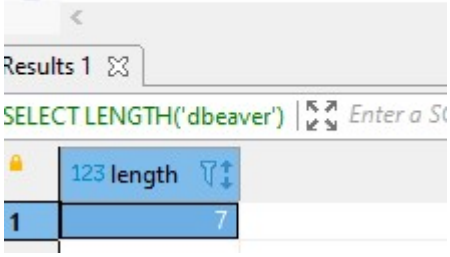
SELECT LEFT('postgresqltutorial', 5)

	ABC left
1	postg

LENGTH()

Returns the length of a string in bytes

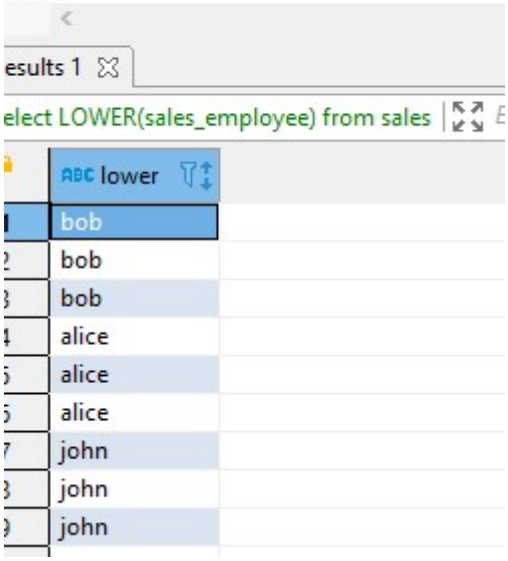
```
SELECT LENGTH('dbeaver');
```



LOWER()

Returns the argument in lowercase

```
select LOWER(sales_employee)
from sales;
```



LTRIM()

Removes leading spaces

```
SELECT LTRIM('  mysql');
```

Results 1

SELECT LTRIM(' mysql') | Enter a SQL

	ltrim
1	mysql

REPEAT()

Repeat a string the specified number of times

```
SELECT REPEAT('MySQL', 3);
```

Results 1

SELECT REPEAT('MySQL', 3) | Enter a SQ

	repeat
1	MySQLMySQLMySQL

REPLACE()

Replaces occurrences of a specified string

```
SELECT REPLACE('postgresql', 'p', 'PP');
```

Results 1

SELECT REPLACE('postgresql', 'p', 'PP') | Enter a SQL expr

	replace
1	PPostgresql

REVERSE()

Reverses the characters in a string

```
select REVERSE(sales_employee)
from sales;
```

Results 1

select REVERSE(sales_employee) from sales

	reverse
1	boB
2	boB
3	boB
4	ecilA
5	ecilA
6	ecilA
7	nhoJ
8	nhoJ
9	nhoJ

RIGHT()

Returns the specified rightmost number of characters

```
SELECT RIGHT('postgresql', 3);
```

Results 1

SELECT RIGHT('postgresql', 3) Enter a SQL

	right
1	sql

RTRIM()

Removes trailing spaces

```
SELECT RTRIM('mysql     ');
```

Results 1

SELECT RTRIM('mysql ')

	rtrim
1	mysql

TRIM()

Removes leading and trailing spaces

```
SELECT TRIM('   sql   ');
```

Results 1

SELECT TRIM(' sql ')

	btrim
1	sql

UPPER()

Converts to uppercase

```
-- UPPER() Converts to uppercase
select upper(sales_employee)
from sales;
```

Results 1

select upper(sales_employee) from sales

	upper
1	BOB
2	BOB
3	BOB
4	ALICE
5	ALICE
6	ALICE
7	JOHN
8	JOHN
9	JOHN

This is not the end. We have many more SQL functions that we can explore and use.

Happy Learning..!!