Date Functions

Function	Type	Example	Result
NOW() * Returns current local date and time.	date/time	NOW()	ex. '2020-02-24 09:31:31'
DATE(date)	date/time	DATE('2020-01-01 11:31:31')	'2020-02-24'
CURRENT_DATE() * Returns current local date	date	CURRENT_DATE	'2020-02-24'
CURRENT_TIME() * Returns current local time.	time	CURRENT_TIME	'11:52:10'
UTC_DATE() * Returns current UTC date.	date	UTC_DATE	'2020-02-24'
UTC_TIME() * Returns current UTC date.	time	UTC_TIME	'18:52:10'

- There are a number of functions that give the current date and time. The DATE() function is a date formatting function, but I include it in the list because it is often confused with the NOW() function.
- CURRENT_DATE, CURRENT_TIME, UTC_DATE, UTC_TIME can be used with the parentheses "()" or not. They accept no parameters

Example

```
SELECT NOW() AS 'NOW()',
   DATE('2020-01-01') AS 'DATE(), date only',
   CURRENT_DATE AS 'CURRENT_DATE',
   CURRENT_TIME AS 'CURRENT_TIME',
   UTC_DATE AS 'UTC_DATE',
   UTC_TIME AS 'UTC_TIME';
```

Results

```
SELECT NOW() AS 'NOW()',
    DATE('2020-01-01') AS 'DATE(), date only',
    CURRENT_DATE AS 'CURRENT_DATE',
    CURRENT_TIME AS 'CURRENT_TIME',
    UTC_DATE AS 'UTC_DATE',
    UTC_TIME AS 'UTC_TIME';
```

Result Grid

NOW()	DATE(), date only	CURRENT_DATE	CURRENT_TIME	UTC_DATE	UTC_TIME
2020-02-24 12:34:21	2020-01-01	2020-02-24	12:34:21	2020-02-24	19:34:21

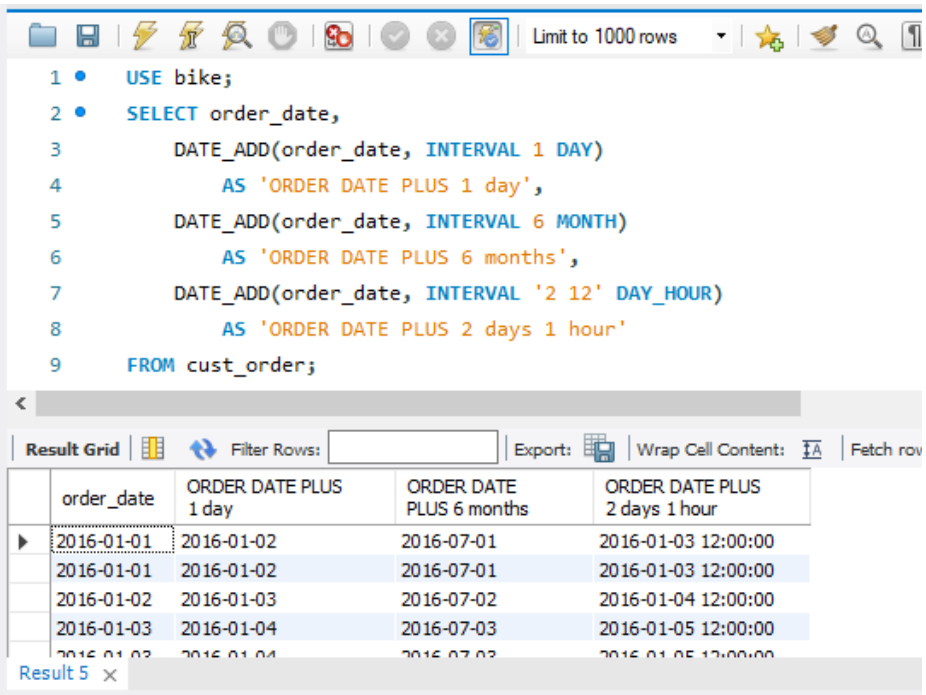
Function	Type	Example	Result
DATE_ADD(date, interval expression unit)	DATE, DATETIME	DATE_ADD('2020-01-01', INTERVAL 1 DAY)	'202-01-02'

- Returns a date with a DATE or DATETIME value equal to the original value plus the specified interval.

Example

```
USE bike;
SELECT order_date,
   DATE_ADD(order_date, INTERVAL 1 DAY) AS 'ORDER DATE PLUS 1 day',
   DATE_ADD(order_date, INTERVAL 6 MONTH) AS 'ORDER DATE PLUS 6 months',
   DATE_ADD(order_date, INTERVAL '2 12' DAY_HOUR)
   AS 'ORDER DATE PLUS 2 days 1 hour'
FROM cust_order;
```

Results



order_date	ORDER DATE PLUS 1 day	ORDER DATE PLUS 6 months	ORDER DATE PLUS 2 days 1 hour
2016-01-01	2016-01-02	2016-07-01	2016-01-03 12:00:00
2016-01-01	2016-01-02	2016-07-01	2016-01-03 12:00:00
2016-01-02	2016-01-03	2016-07-02	2016-01-04 12:00:00
2016-01-03	2016-01-04	2016-07-03	2016-01-05 12:00:00

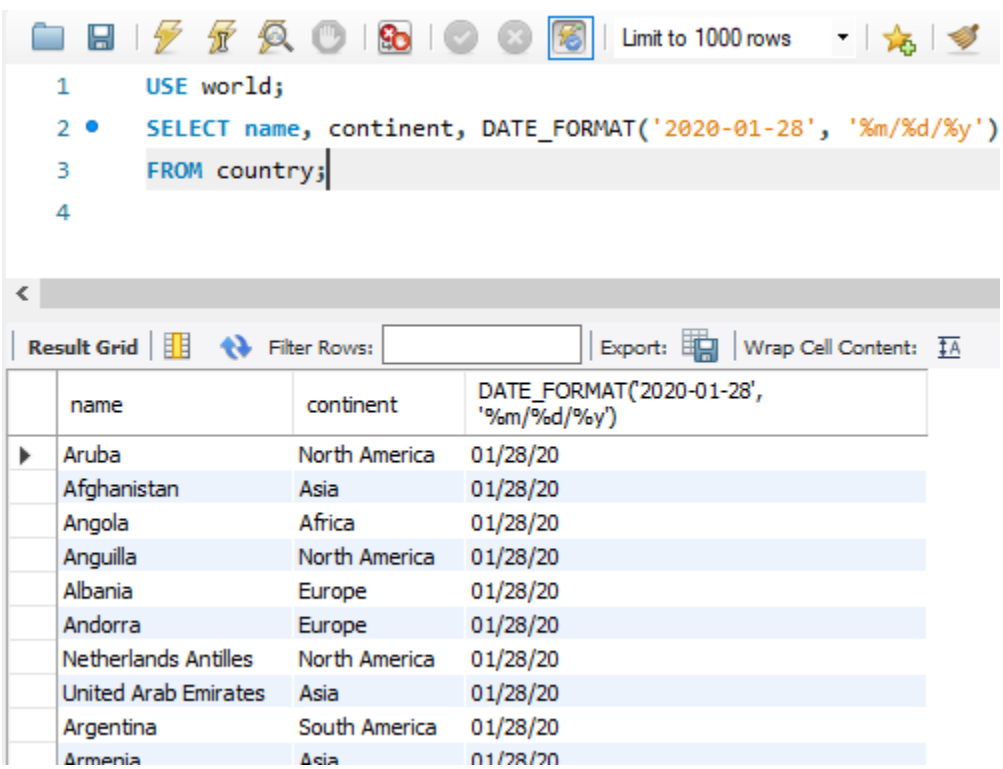
Function	Type	Example	Result
DATE_FORMAT	DATE	DATE_FORMAT('2020-09-03', '%m/%d/%y')	09/03/14

- Dates must be enclosed in quotes
- You can pass a DATE or DATETIME datatype to DATE_FORMAT

Example

```
USE world;
SELECT name, continent, DATE_FORMAT('2020-01-28', '%m/%d/%y')
FROM country;
```

Results



name	continent	DATE_FORMAT('2020-01-28', '%m/%d/%y')
Aruba	North America	01/28/20
Afghanistan	Asia	01/28/20
Angola	Africa	01/28/20
Anguilla	North America	01/28/20
Albania	Europe	01/28/20
Andorra	Europe	01/28/20
Netherlands Antilles	North America	01/28/20
United Arab Emirates	Asia	01/28/20
Argentina	South America	01/28/20
Armenia	Asia	01/28/20

Format List

Specifier	Description
%a	Abbreviated weekday name (Sun..Sat)
%b	Abbreviated month name (Jan..Dec)
%c	Month, numeric (0..12)
%D	Day of the month with English suffix (0th, 1st, 2nd, 3rd, …)
%d	Day of the month, numeric (00..31)
%e	Day of the month, numeric (0..31)
%f	Microseconds (000000..999999)
%H	Hour (00..23)
%h	Hour (01..12)
%I	Hour (01..12)
%i	Minutes, numeric (00..59)
%j	Day of year (001..366)
%k	Hour (0..23)
%l	Hour (1..12)
%M	Month name (January..December)
%m	Month, numeric (00..12)
%p	AM or PM
%r	Time, 12-hour (***hh:mm:ss*** followed by AM or PM)
%S	Seconds (00..59)
%s	Seconds (00..59)
%T	Time, 24-hour (***hh:mm:ss***)
%U	Week (00..53), where Sunday is the first day of the week; WEEK() mode 0
%u	Week (00..53), where Monday is the first day of the week; WEEK() mode 1
%V	Week (01..53), where Sunday is the first day of the week; WEEK() mode 2; used with %X
%v	Week (01..53), where Monday is the first day of the week; WEEK() mode 3; used with %x
%W	Weekday name (Sunday..Saturday)
%w	Day of the week (0=Sunday..6=Saturday)
%X	Year for the week where Sunday is the first day of the week, numeric, four digits; used with %V
%x	Year for the week, where Monday is the first day of the week, numeric, four digits; used with %v
%Y	Year, numeric, four digits
%y	Year, numeric (two digits)
%%	A literal % character
%***x***	***x***, for any "***x***" not listed above

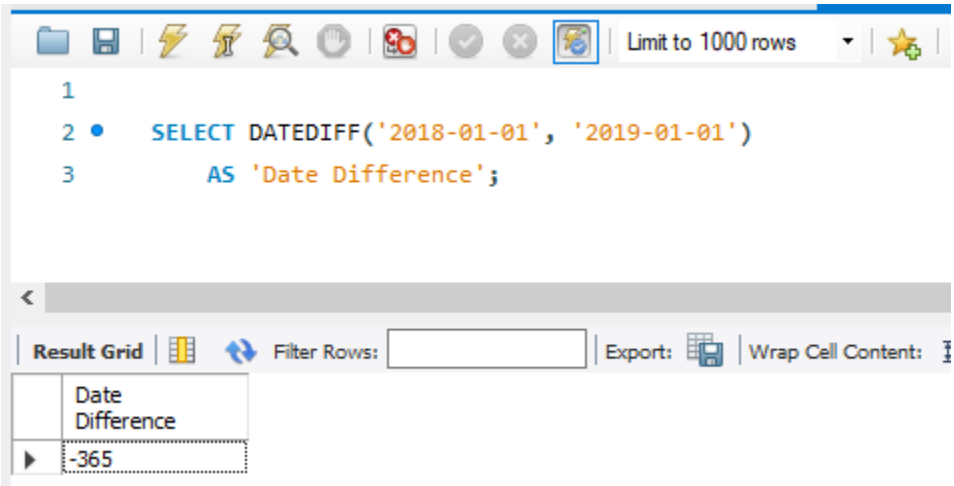
Function	Type	Example	Result
DATEDIFF(date1, date2]	DATE	DATEDIFF('2018-01-01', '2019-01-01')	-356

- The DATEDIFF function has two parameters. Both are dates.
- The value returned by the function is an integer and is the number of days between the two dates.
- If you provide the latest date, first the results will be positive. Otherwise, it will be negative.

Example

```
SELECT DATEDIFF('2018-01-01', '2019-01-01')
AS 'Date Difference';
```

Results



Numeric Functions

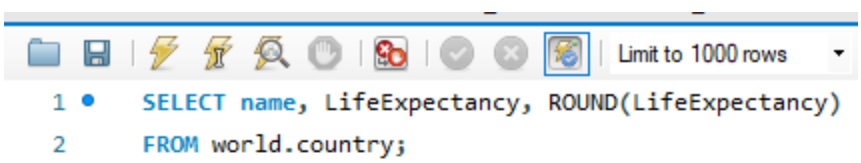
Function	Type	Example	Result
ROUND(number[, length])	Number	ROUND(13.37, 1)	13.4

- The ROUND function has two parameters. The first is a number, usually a DECIMAL or a FLOAT. The second defines the number of decimals to which the number will be rounded.
- If no length is provided, the number is rounded to a whole number.

Example

```
USE world;
SELECT name, LifeExpectancy, ROUND(LifeExpectancy)
FROM world.country;
```

Results



Result Grid | Filter Rows: | Export: | Wrap Cell Co

name	LifeExpectancy	ROUND(LifeExpectancy)
Aruba	78.4	78
Afghanistan	45.9	46
Angola	38.3	38
Anguilla	76.1	76
Albania	71.6	72
Andorra	83.5	84
Netherlands Antilles	74.7	75
United Arab Emirates	74.1	74

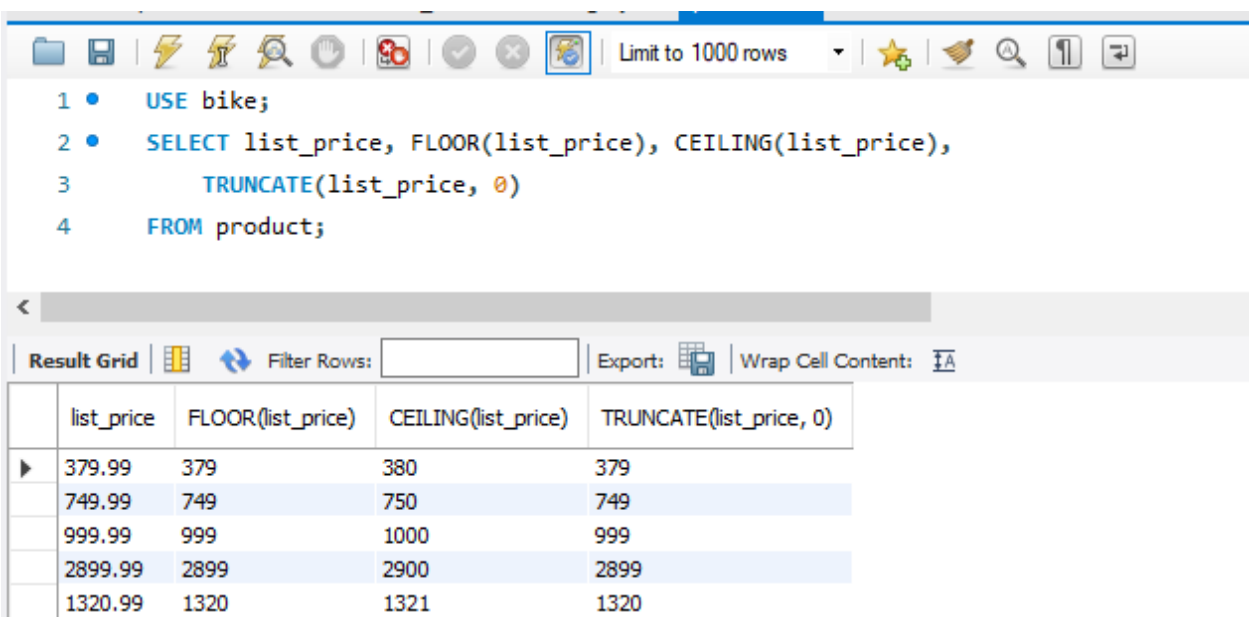
Function	Type	Example	Result
FLOOR(number)	number	FLOOR(7.7)	7
CEILING(number)	number	CEILING(6.2)	7
TRUNCATE(NUMBER, length)	number	TRUNCATE(7.9)	7

- FLOOR() will return the next lowest whole number no matter what the decimal point.
- CEILING() will return the next highest whole number no matter what the decimal point.
- TRUNCATE() will return the number truncated to the precision specified.

Example

```
USE bike;
SELECT list_price, FLOOR(list_price), CEILING(list_price),
    TRUNCATE(list_price, 0)
FROM product;
```

Results



list_price	FLOOR(list_price)	CEILING(list_price)	TRUNCATE(list_price, 0)
379.99	379	380	379
749.99	749	750	749
999.99	999	1000	999
2899.99	2899	2900	2899
1320.99	1320	1321	1320

String Functions

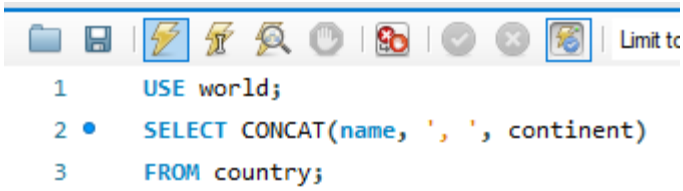
Function	Type	Example	Result
CONCAT(string1[, string2]…)	String	CONCAT(“Salmon”, “Fish”)	Salmon Fish

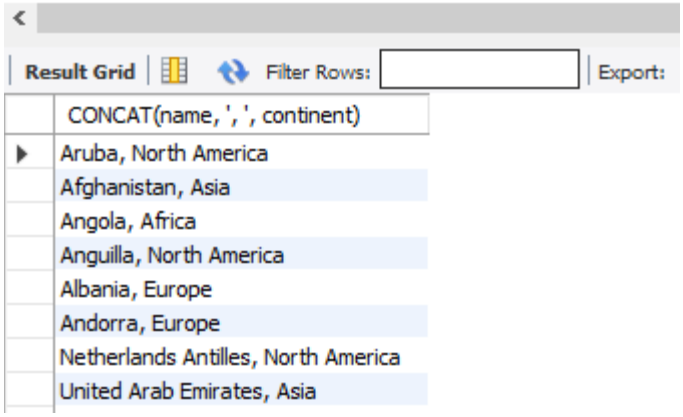
- Can include column values and literal values.
- In MySQL literal values can be enclosed with either single (‘) or double quotes (“) .

Example

```
USE world;
SELECT CONCAT(name, ', ', continent)
FROM country;
```

Results





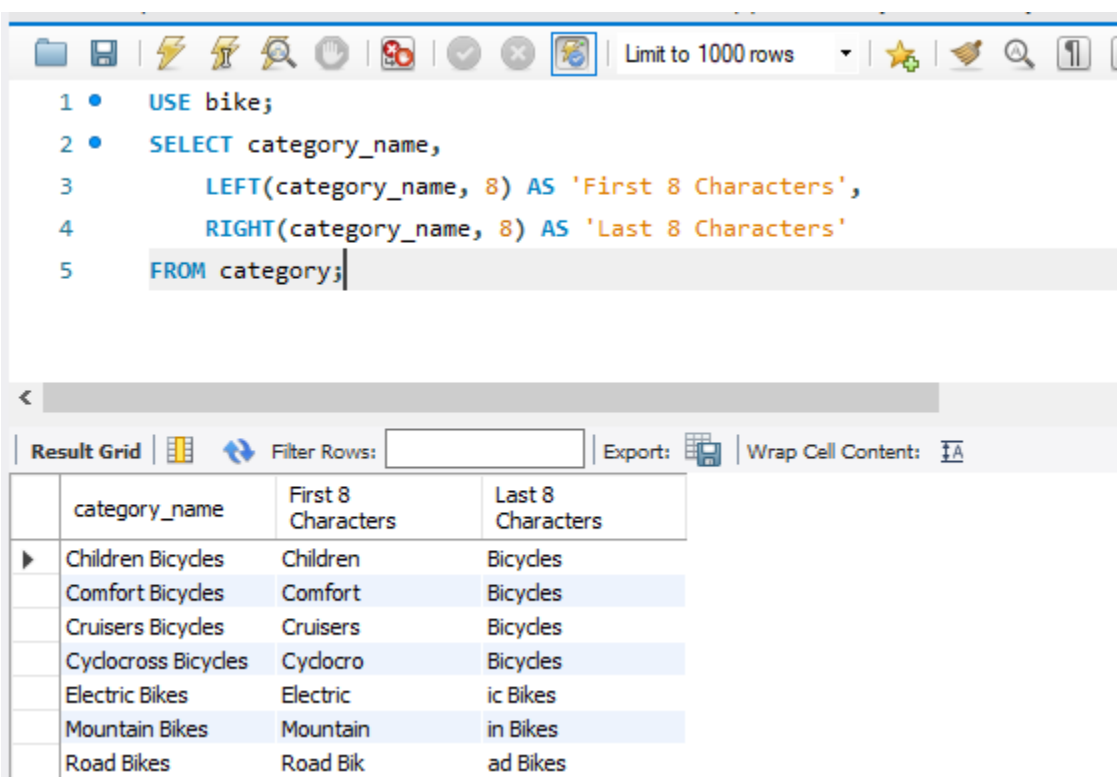
Function	Type	Example	Result
RIGHT(string, num. characters)	string	RIGHT('Salmon', 3)	mon
LEFT(string, num. characters)	string	LEFT('Salmon', 3)	Sal

- The RIGHT and LEFT functions have two parameters. The first is a string and the second is the number of characters to be returned.
- The RIGHT function starts counting from the right side of the string.
- The LEFT function starts counting from the left side of the string.

Example

```
USE bike;
SELECT category_name,
    LEFT(category_name, 8) AS 'First 8 Characters',
    RIGHT(category_name, 8) AS 'Last 8 Characters'
FROM category;
```

Results



category_name	First 8 Characters	Last 8 Characters
Children Bicycles	Children	Bicycles
Comfort Bicycles	Comfort	Bicycles
Cruisers Bicycles	Cruisers	Bicycles
Cyclocross Bicycles	Cyclocro	Bicycles
Electric Bikes	Electric	ic Bikes
Mountain Bikes	Mountain	in Bikes
Road Bikes	Road Bik	ad Bikes

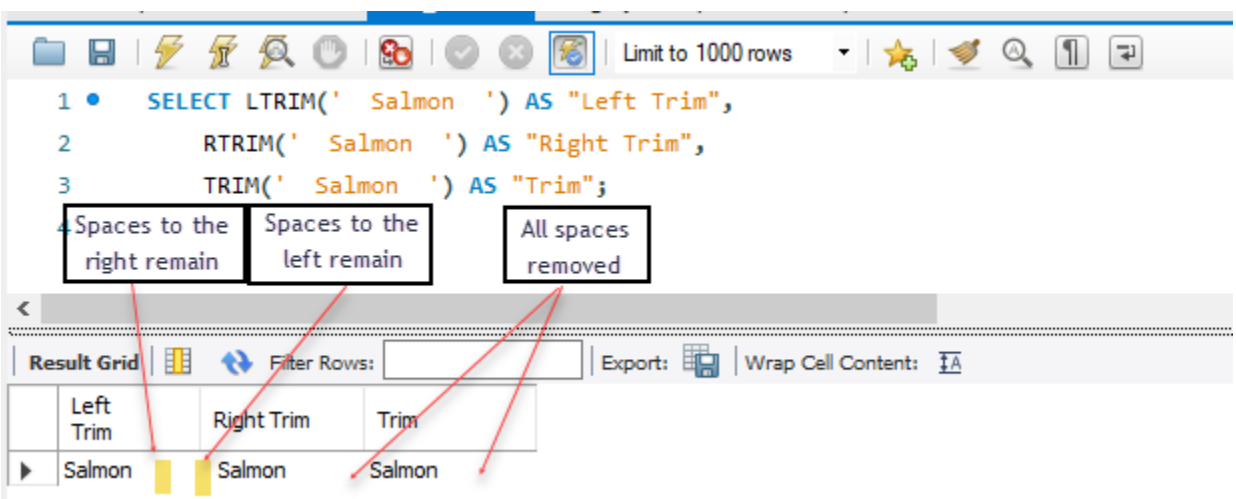
Function	Type	Example	Result
TRIM(string)	string	TRIM(‘ Salmon ‘)	‘salmon’
LTRIM(string)	string	LEFT(‘Salmon ‘)	‘salmon ’
RTRIM(string)	string	RIGHT(‘ Salmon‘)	‘ salmon’

- The TRIM function will remove leading and trailing spaces from a string.
- The LTRIM function will remove leading spaces from a string.
- The RTRIM function will remove trailing spaces from a string.

Example

```
SELECT LTRIM('  Salmon  ') AS "Left Trim",
     RTRIM('  Salmon  ') AS "Right Trim",
    TRIM('  Salmon  ') AS "Trim";
```

Results



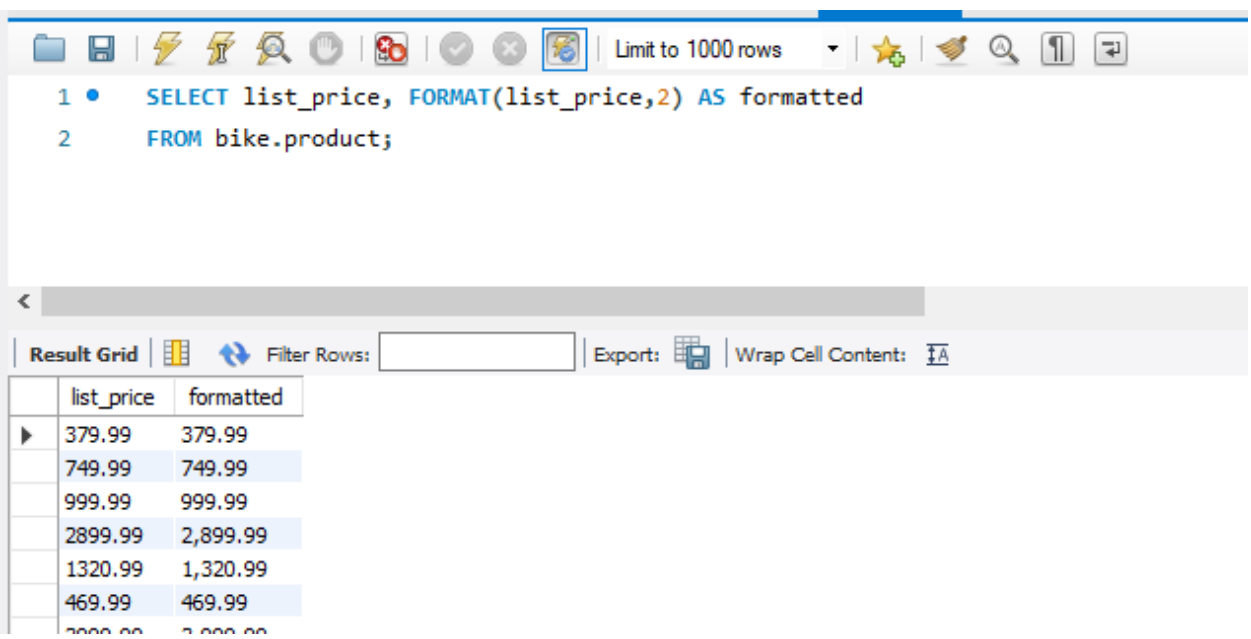
Function	Type	Example	Result
FORMAT(number, decimal)	string	FORMAT(1234.342, 2)	-356

- FORMAT() accepts a decimal but returns a comma formatted string.

Example

```
SELECT FORMAT(list_price,2)
FROM bike.product;
```

Results



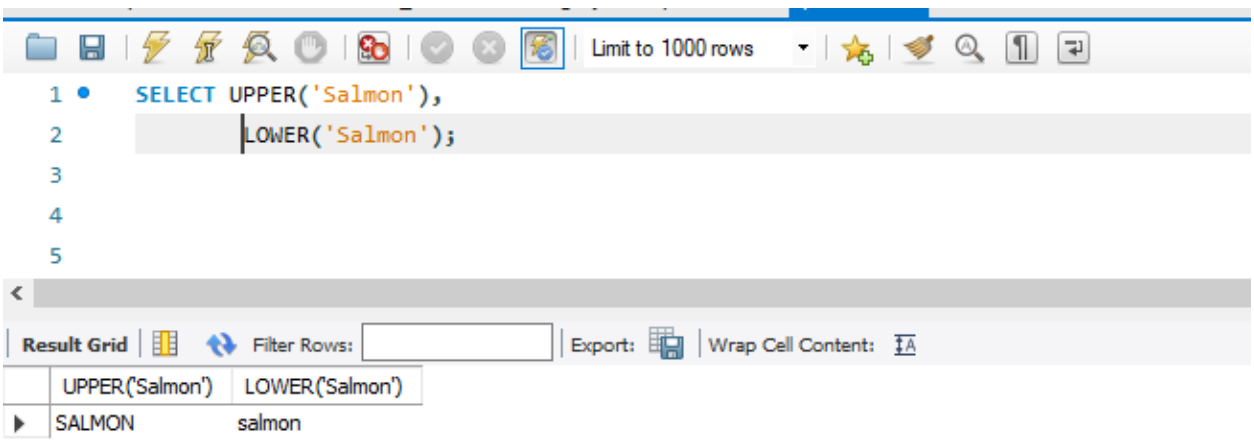
Function	Type	Example	Result
LOWER(string)	string	LOWER(‘Salmon ‘)	‘salmon’
UPPER(string)	string	UPPER(‘Salmon‘)	‘SALMON’

- LOWER() converts all characters to lower case.
- UPPER() converts all characters to upper case.

Example

```
SELECT UPPER('Salmon'),
       LOWER('Salmon');
```

Results



Function	Type	Example	Result
LOCATE(find,search[,start])	string	LOCATE('al','salmon',1)	2
LENGTH(str)	string	LENGTH('salmon')	6
SUBSTRING(str,start[,length])	string	SUBSTRING('salmon',3,999)	'lmon'

- LOCATE(), and LENGTH() accept a string but return an integer.
- SUBSTRING() accepts a string and returns a string.

Example

```
SELECT LOCATE('al','salmon',1),
       LENGTH('salmon'),
      SUBSTRING('salmon',3,999);
```

Results