

### 3.3

## String Functions

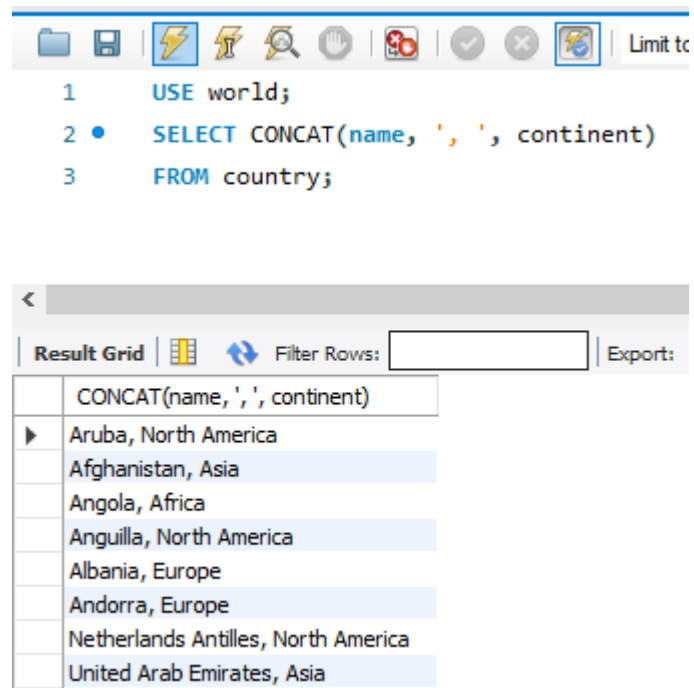
### CONCAT

- Combines a list of strings into a single string.
- 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:*



The screenshot shows a SQL IDE interface. The top toolbar contains icons for file operations, execution, and navigation. Below the toolbar, the SQL query is displayed in a text area, numbered 1 to 3. The query is: `USE world;`, `SELECT CONCAT(name, ', ', continent)`, and `FROM country;`. Below the query, the results are shown in a table. The table has one column, `CONCAT(name, ', ', continent)`, and several rows of data. The first row is highlighted with a blue background.

CONCAT(name, ', ', continent)
Aruba, North America
Afghanistan, Asia
Angola, Africa
Anguilla, North America
Albania, Europe
Andorra, Europe
Netherlands Antilles, North America
United Arab Emirates, Asia

## RIGHT, LEFT

- 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.

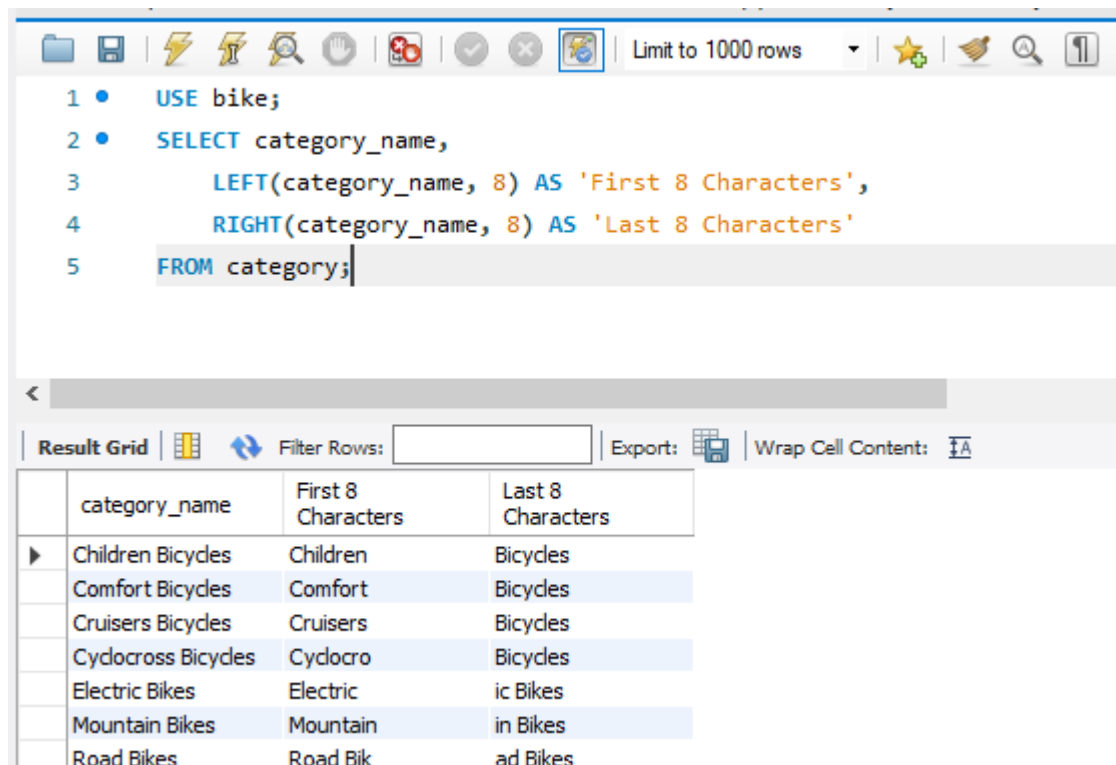
Table 7. RIGHT, LEFT functions

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

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:



The screenshot shows a database query editor with a toolbar at the top. The SQL query is entered in the main area:

```
1 • USE bike;
2 • SELECT category_name,
3       LEFT(category_name, 8) AS 'First 8 Characters',
4       RIGHT(category_name, 8) AS 'Last 8 Characters'
5 FROM category;
```

Below the query editor, the results are displayed in a table grid. The grid has four columns: category\_name, First 8 Characters, and Last 8 Characters. The data is as follows:

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

## TRIM, LTRIM, RTRIM

- 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.

Table 8. TRIM functions

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

Example:

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

Results:

The screenshot shows a database query editor with the following SQL query:

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

Below the query, there are three annotations in boxes:

- Spaces to the right remain**: Points to the 'Left Trim' result.
- Spaces to the left remain**: Points to the 'Right Trim' result.
- All spaces removed**: Points to the 'Trim' result.

The results grid shows the following data:

Left Trim	Right Trim	Trim
salmon	salmon	salmon

## FORMAT

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

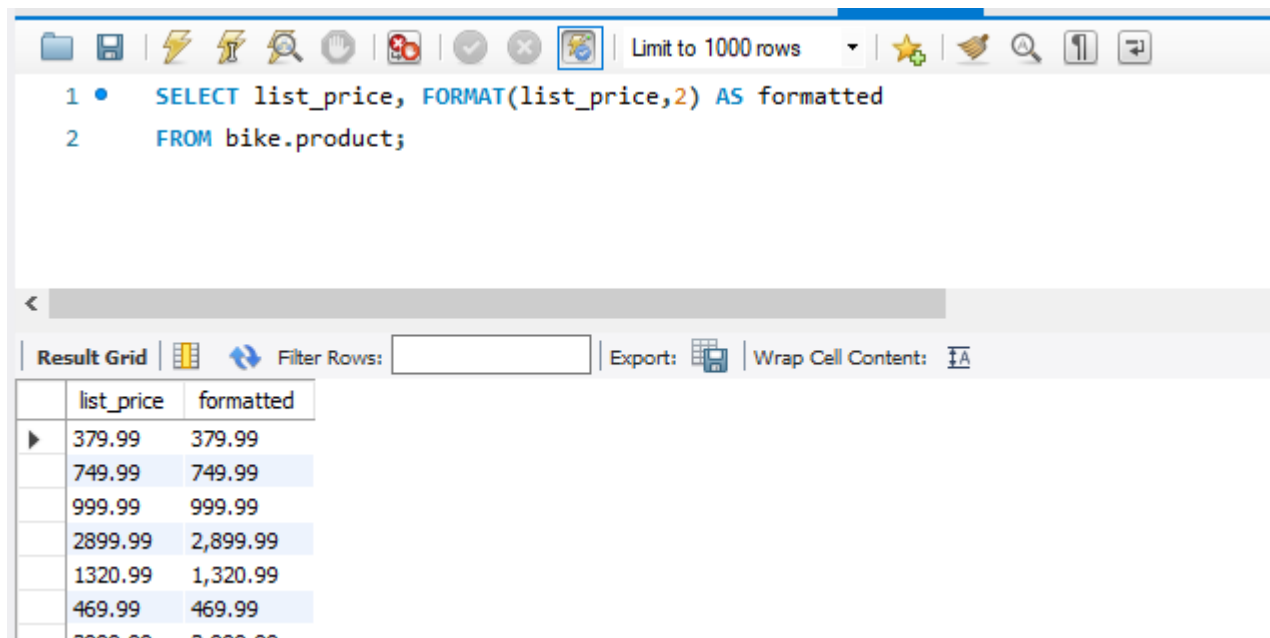
Table 9. FORMAT functions

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

Code Sample:

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

Results:



list_price	formatted
379.99	379.99
749.99	749.99
999.99	999.99
2899.99	2,899.99
1320.99	1,320.99
469.99	469.99
2000.00	2,000.00

## LOWER, UPPER

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

Table 9. LOWER, UPPER functions

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

Example:

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

Results:

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The query editor contains the following SQL code:

```
1 • SELECT UPPER('Salmon'),
2     LOWER('Salmon');
3
4
5
```

Below the editor is the 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table:

	UPPER('Salmon')	LOWER('Salmon')
▶	SALMON	salmon

## LOCATE, LENGTH, SUBSTRING

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

Table 9. LOCATE. LENGTH, SUBSTRING functions

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'

*Example:*

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

*Results:*

Limit to 1000 rows

```
1 • SELECT LOCATE('a1','salmon',1),
2     LENGTH('salmon'),
3     SUBSTRING('salmon',3,999);
4
5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [A](#)

	LOCATE('a1','salmon',1)	LENGTH('salmon')	SUBSTRING('salmon',3,999)
▶ 2		6	lmon



This content is provided to you freely by BYU-I Books.

Access it online or download it at [https://books.byui.edu/learning\\_mysql/string\\_functions](https://books.byui.edu/learning_mysql/string_functions).