String commands

This lesson introduces important SQL commands that can be applied to strings. These commands are used to manipulate or extract strings.

As you may remember from the Excel lessons, strings are treated differently than numbers.

Strings are text data, and SQL has a separate set of commands specifically for manipulating strings.

There are lots of uses for string commands in SQL.

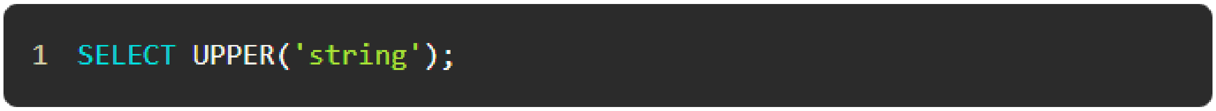
- One use is to bring consistency to a column of data by removing extra spaces in the text or ensuring that all letters have the same case.
- You may want to extract certain characters from a string to create shorter values or to exclude unnecessary characters.
- You can even combine two separate strings to form a new string!

There are numerous string commands in SQL, but this course focuses on the most used commands. These commands may remind you of string functions in Excel—that's because the concepts are the same.

Shown here: **UPPER**, **LOWER**, **TRIM**, **CONCAT**, **LEFT**, **RIGHT**.

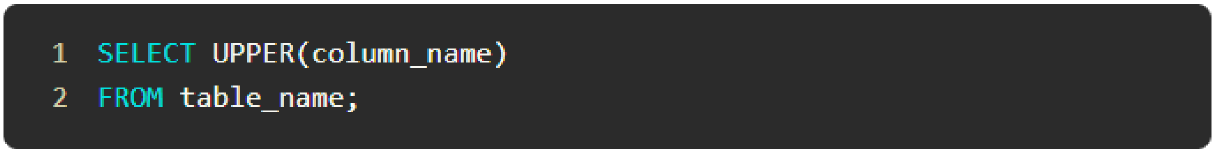
UPPER

The UPPER command in SQL capitalizes all letters in the string. The syntax is straightforward:



```sql
SELECT UPPER('string');
```

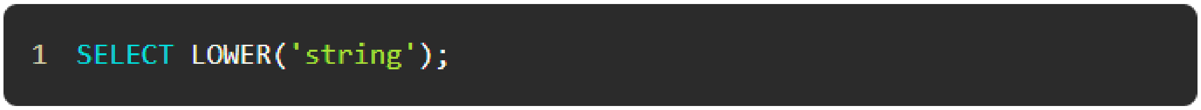
To apply UPPER to an entire column, use the following structure:



```sql
SELECT UPPER(column_name)
FROM table_name;
```

LOWER

The **LOWER** command in SQL is the inverse of the **UPPER** command: it makes all letters in a string lowercase. The syntax is exactly like that of the **UPPER**.

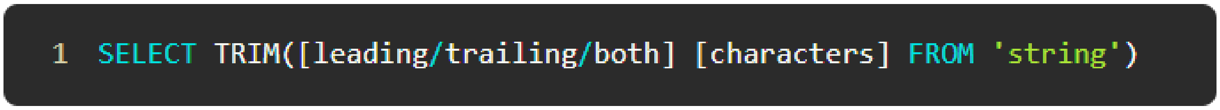


```sql
SELECT LOWER('string');
```

TRIM

Sometimes, data contains unnecessary spaces or other unwanted extra characters. If you want to remove unwanted characters—such as extra spaces or symbols—from a string, use the TRIM command in SQL.

The structure of TRIM is as follows:

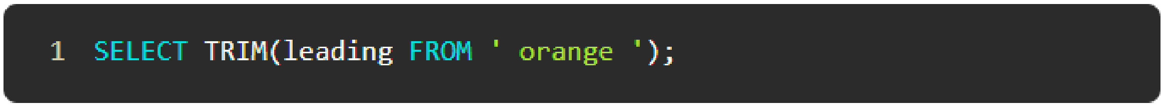


```
SELECT TRIM([leading/trailing/both] [characters] FROM 'string')
```

Here are a few things to keep in mind when using TRIM:

- The word leading is used when the characters to be removed are at the beginning of the string.
- The word trailing is used when the characters to be removed are at the end of the string.
- The word both is used when the characters to be removed are at the beginning and the end of the string.
- If the characters argument is left blank, then only spaces are removed.
- The string argument can be a string or a column name in a table.

Example:



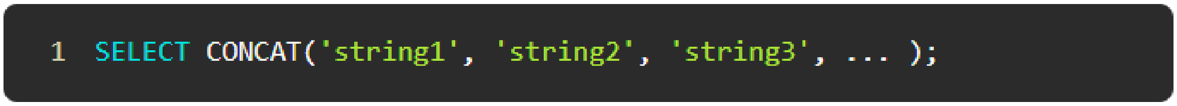
```
SELECT TRIM(leading FROM ' orange ');
```

CONCAT

There are scenarios where you may want to combine strings.

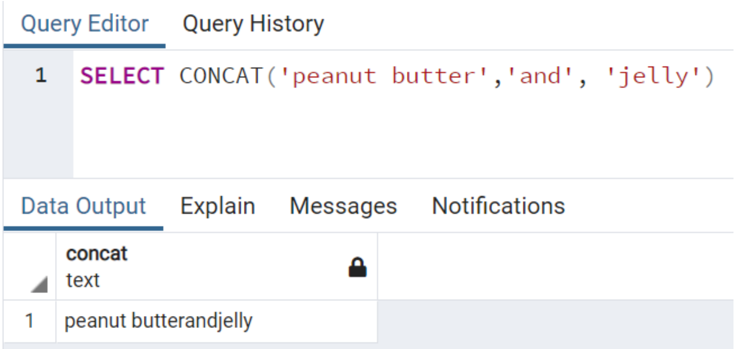
For example, you may want to combine a first name and last name to create a full name. Similarly, you may want to combine a street address with a city, state, and zip code to create a full address. Or you may want to create a unique code for every row, by combining two or more values together.

The CONCAT command in SQL will combine strings for you so that you don't have to type them out yourself! The structure of CONCAT is as follows:



```
SELECT CONCAT('string1', 'string2', 'string3', ... );
```

Try this in pgAdmin now. Create and execute a query that combines the strings peanut butter, and, and jelly. If you execute the query with these three strings, this is the output that you'll get:



```
SELECT CONCAT('peanut butter','and', 'jelly')
```

```
  concat
  text
1 peanut butterandjelly
```

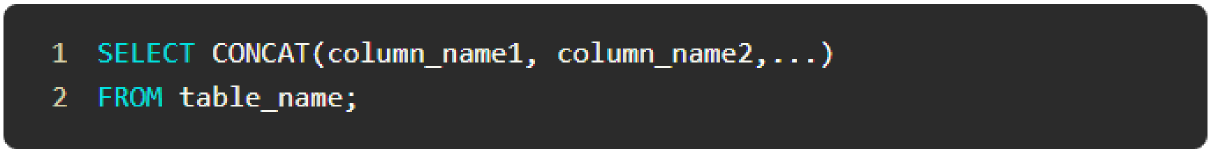
Notice that this query didn't add spaces when it combined peanut butter, and, and jelly.

CONCAT combines the strings that you specify, without adding any additional spaces or characters.

So, if you have a phrase like this, remember to include spaces where appropriate.

CONCAT with Multiple Columns

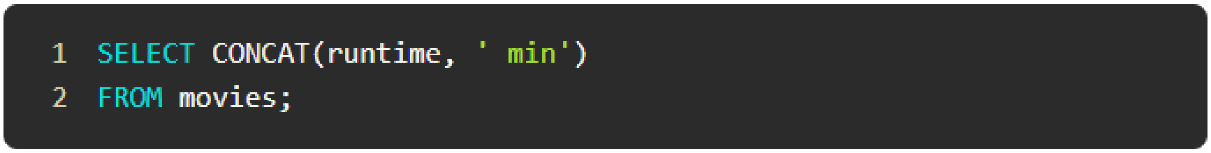
You can use CONCAT to combine values in multiple columns in a data table as well. The structure remains the same, with column names used in place of string names. Take a look:



```
SELECT CONCAT(column_name1, column_name2,...)
FROM table_name;
```

The movies table contains a column called runtime. This column contains the length of each movie in minutes, but it doesn't specify the units. Use CONCAT to add min to the end of each value. Remember to add any additional characters or spaces as string arguments in your CONCAT statement.

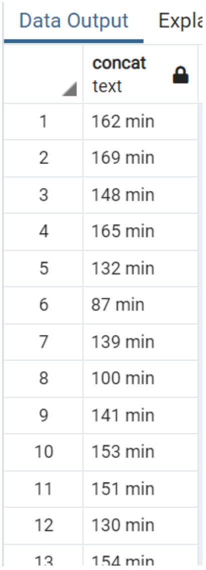
Your query should look like this:



```sql
SELECT CONCAT(runtime, ' min')
FROM movies;
```

Notice that an extra space is added in front of min to ensure that there is a space between the number of minutes and the text min. Also, although runtime is an integer data type, CONCAT treats it like a string.

Your output will show the following:

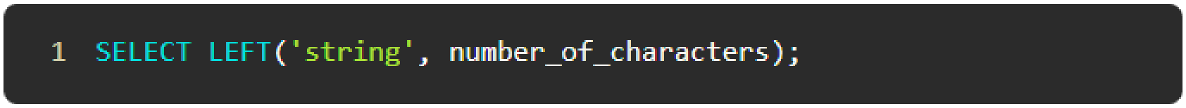


	concat text
1	162 min
2	169 min
3	148 min
4	165 min
5	132 min
6	87 min
7	139 min
8	100 min
9	141 min
10	153 min
11	151 min
12	130 min
13	154 min

You just created a new column by combining an existing column and text! As you can see, CONCAT is powerful and can be used to easily combine strings together.

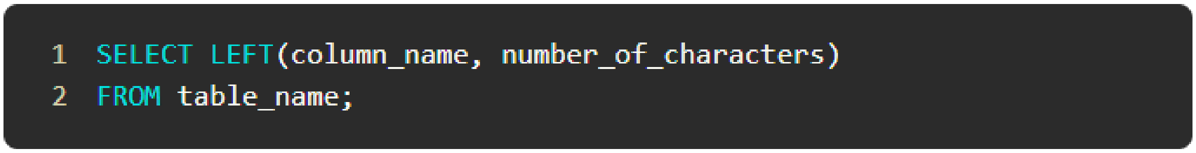
LEFT

Say a column contains long strings and you only need the first few characters. The LEFT command lets you extract a specified number of characters from the beginning of the string. The structure of this command is as follows:



```
SELECT LEFT('string', number_of_characters);
```

From a column:



```
SELECT LEFT(column_name, number_of_characters)
FROM table_name;
```

RIGHT is exactly the same as the **LEFT**.