

Query Ninja: String Manipulation with MapInfo Pro SQL - Part 1

This article shows how to manipulate text (string) data with various string functions. For example, you can extract certain characters from the beginning, middle or end of a string. One example of how this might be used is to extract a part of a postal code out of a full postal code, for example the first five digits of a US nine digit post code or in the U.K. the Postal area and District (everything up to the space) from a full unit postcode.

Jun 28, 2017•How To

Instructions / Information

The MapInfo Pro™ Monthly Journal

Be the Query Ninja: String Manipulation with MapInfo Pro SQL - Part 1



The information in this article pertains to both the 32 and 64 bit versions of MapInfo Pro.

From time to time, you may get some data from other parties that might not be as clean as you could want it to be. You might have to erase certain parts of a string, you might have to split one string into several or maybe you need to replace certain words with different words.

Luckily, the MapBasic language comes with a number of functions that can be used for searching thru strings and extracting parts from strings.

These MapBasic functions can also be used when querying tables and when updating table.

Here is a list of some of the MapBasic string functions:

- Find a sub-string: InStr()
- Extract part of a string: Left\$(), Right\$(), Mid\$()
- Upper/Lower case: UCase\$(), LCase\$(), Proper\$()
- Trim blanks from a string: LTrim\$(), RTrim\$()
- Determine string length: Len()
- Repeat a string sequence: Space\$(), String\$()

Query statements

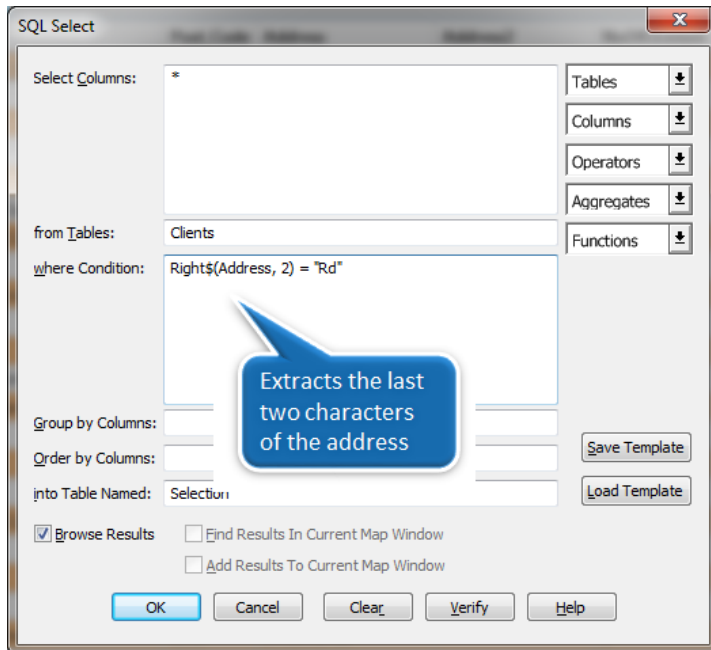
Let have a look at some practical query statement to see how you can use these functions in an SQL statement.

In many cases, you might want to find the records that meet a certain condition. Maybe the addresses of your data have been entered with abbreviations and you want to change these to the full words, for example changing "12 High St" to "12 High Street", "234 Main Rd" to "234 Main Road".

<https://li360.pitneybowes.com/s/article/Query-Ninja-String-Manipulation-with-MapInfo-Pro-SQL-Part-1>

First you need to locate the records that are using the abbreviation. We can use a condition like this:

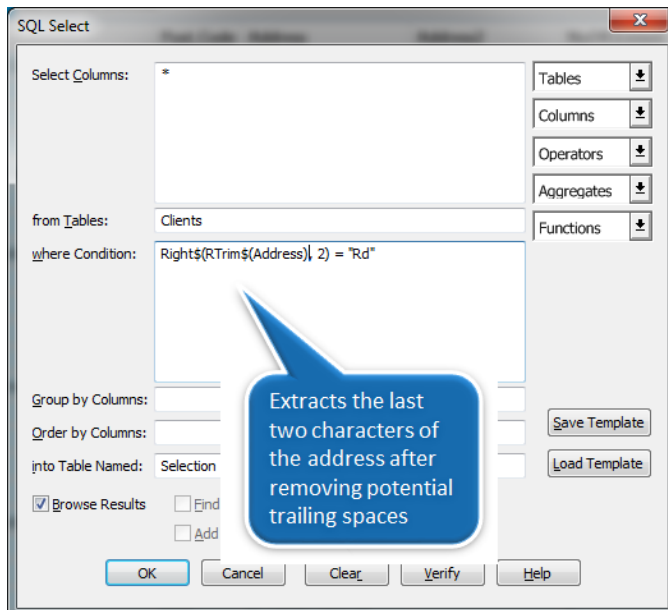
```
Right$(RTrim$(Address), 2) = "Rd"
```



The `Right$()` function extracts a number of characters from the right of the text. In the example here, we extract the two characters to the right.

If you are uncertain about how clean the data is, you can use `RTrim$()` to remove potential trailing spaces from the address:

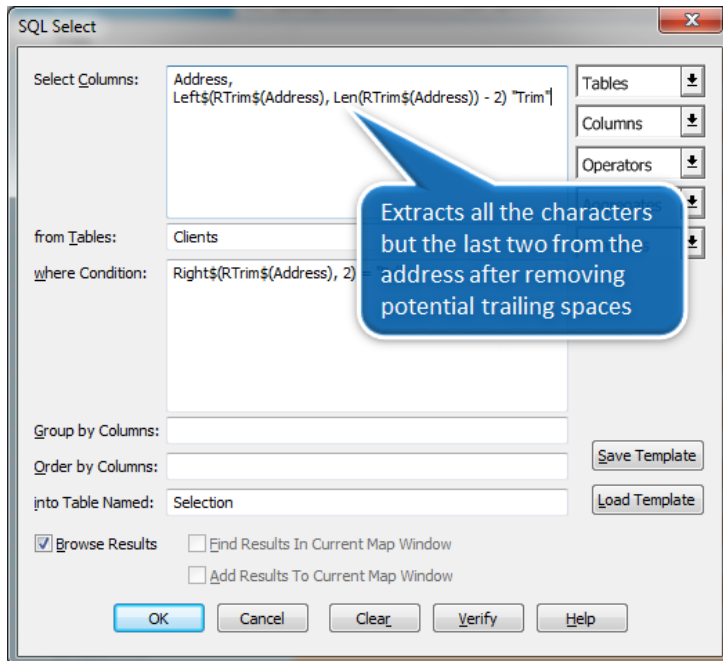
```
Right$(RTrim$(Address), 2) = "Rd"
```



The `RTrim$()` function removes any potential spaces at the end of the string. There is a similar function for removing spaces at the beginning of the string - it is called `LTrim$()`.

Here is another technique:

<https://li360.pitneybowes.com/s/article/Query-Ninja-String-Manipulation-with-MapInfo-Pro-SQL-Part-1>



Using one of the two methods above, we can find the addresses that end with the text "Rd".

Now let us remove the "Rd" from the string and compare that with the original address to validate it looks fine. To do so, we will add some expressions to the Select Columns field:

```
Address,
Left$(RTrim$(Address), Len(RTrim$(Address)) - 2) "Trim"
```

The Left\$ () function works similar to the Right\$ () function. It just reads from the beginning of the string.

The Len () function is used to "measure" the length of the string - the number of characters in the string.

I have also included the RTrim\$ () function in the expression to remove potential trailing spaces from address.

In the browser window below, you can see the original addresses on the left and the trimmed versions on the right.

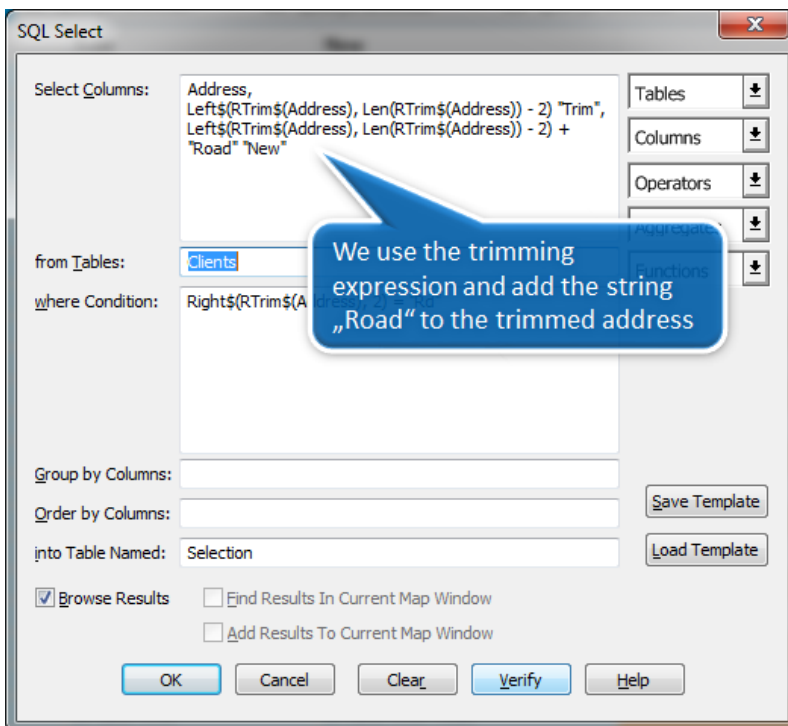
Address	Trim
69 Broad Creek Rd	69 Broad Creek
27 Lanes Rd	27 Lanes
53/158 Roostreet Rd	53/158 Roostreet
41 /60 Back Rd	41 /60 Back
92 Back Rd	92 Back
77 Broad Creek Rd	77 Broad Creek
73 Broad Creek Rd	73 Broad Creek
73 Broad Creek Rd	73 Broad Creek
37 Broad Creek Rd	37 Broad Creek
73 Broad Creek Rd	73 Broad Creek
31 Broad Creek Rd	31 Broad Creek
31 Arve Rd	31 Arve
38 Broad Creek Rd	38 Broad Creek
101 Roostreet Rd	101 Roostreet
33 Kermandie Rd	33 Kermandie
74 Golden Valley Rd	74 Golden Valley
1/61 The Narrows Rd	1/61 The Narrows
69 Broad Creek Rd	69 Broad Creek
40 Paddys Flats Rd	40 Paddys Flats
42/455 Roostreet Rd	42/455 Roostreet
10 Lake River Rd	10 Lake River
518 Roostreet Rd	518 Roostreet

The trimmed address without „Rd“

The original address with „Rd“ at the end

We can now also add "Road" to the string and see if that looks fine, too. We just have to add another expression to the Select Columns field:

```
Address,
Left$(RTrim$(Address), Len(RTrim$(Address)) - 2) "Trim",
Left$(RTrim$(Address), Len(RTrim$(Address)) - 2) + "Road" "New"
```



Here we add - or concatenate - the string "Road" onto the trimmed version of the address. We keep the other two columns or expressions as well. This makes it easier to visually check if the result looks correct.

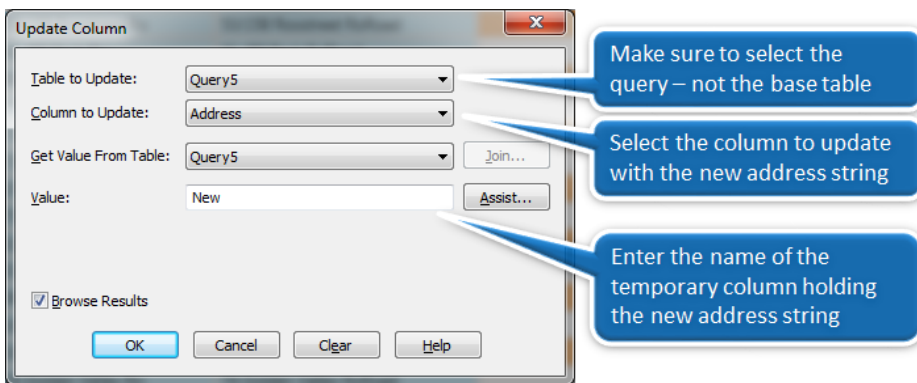
Address	Trim	New
69 Broad Creek Rd	69 Broad Creek	69 Broad Creek Road
27 Lanes Rd	27 Lanes	27 Lanes Road
53/158 Roostreet Rd	53/158 Roostreet	53/158 Roostreet Road
41 /60 Back Rd	41 /60 Back	41 /60 Back Road
92 Back Rd	92 Back	92 Back Road
77 Broad Creek Rd	77 Broad Creek	77 Broad Creek Road
73 Broad Creek Rd	73 Broad Creek	73 Broad Creek Road
73 Broad Creek Rd	73 Broad Creek	73 Broad Creek Road
37 Broad Creek Rd	37 Broad Creek	37 Broad Creek Road
73 Broad Creek Rd	73 Broad Creek	73 Broad Creek Road
31 Broad Creek Rd	31 Broad Creek	31 Broad Creek Road
31 Arve Rd	31 Arve	31 Arve Road
38 Broad Creek Rd	38 Broad Creek	38 Broad Creek Road
101 Roostreet Rd	101 Roostreet	101 Roostreet Road
33 Kermandie Rd	33 Kermandie	33 Kermandie Road
74 Golden Valley Rd	74 Golden Valley	74 Golden Valley Road
1/61 The Narrows Rd	1/61 The Narrows	1/61 The Narrows Road
69 Broad Creek Rd	69 Broad Creek	69 Broad Creek Road
40 Paddys Flats Rd		40 Paddys Flats Road
42/455 Roostreet Rd		42/455 Roostreet Road
10 Lake River Rd		10 Lake River Road
518 Roostreet Rd	518 Roostreet	518 Roostreet Road

The original address with „Rd“ at the end

The trimmed address without „Rd“

The trimmed address with „Road“ added

When you have verified that the New address looks fine, you can update the original address column with the value from the new address column using Update Column.



Be careful, that you select the latest query and not the base table when updating.

You can also use the expression that trims the address and adds "Road" to the address directly in the Update Column dialog. This can be used directly on the first or second query where we select the records that needs to be updated.