

# Relational DB & SQL - C11

[Dashboard](#) / [Courses](#) / [Miscellaneous](#) / [RDB & SQL - C11](#) / [Functions](#) / [String Functions](#)

## String Functions

**To do:** Go through the activity to the end

### LEN(), CHARINDEX(), PATINDEX() Functions

#### LEN()

The **LEN()** function returns the number of characters of a string (excluding spaces end of the text). The return type of the result is an integer.

Here is an example of the **LEN()** function:

query:

```
1 SELECT LEN('this is an example') AS sample
```

result:

```
1 sample
2 -----
3 18
```

If string is **NULL** value, the length function returns **NULL**. If the value specified inside the function is numeric, the **LEN()** function returns the length of a string representation of the value. That means the numeric value is converted into a string and then the number of characters of it is calculated. Such as:

query:

```
1 SELECT LEN(NULL) AS col1, LEN(10) AS col2, LEN(10.5) AS col3
```

result:

```
1 col1  col2  col3
2 ----  ----  ----
3 NULL  2      4
```

#### CHARINDEX(substring, string [, start location])

**CHARINDEX()** function takes a string and a substring of it as arguments and returns an integer that indicates the position of the substring, which is the first character of the substring. **CHARINDEX()** function finds the first occurrence of substring and returns a value of integer type.

**CHARINDEX()** function takes three arguments:



**substring:** The substring to be searched withing string

**string:** The string to be searched

**[start location]:** The index that the function will start searching (optional).

If the substring is not found, `CHARINDEX()` function returns 0.

### Examples:

❶ The following example searches the string 'Reinvent yourself' to find the substring 'yourself'. It returns the index number of the first character of 'yourself' which is 'y'.

query:

```
1 SELECT CHARINDEX('yourself', 'Reinvent yourself') AS  
   start_position;
```

result:

```
1 start_position  
2 -----  
3 10
```

❷ `CHARINDEX()` function works case-sensitively. The following query returns the index number of the first occurrence of the substring 'r' not 'R'.

query:

```
1 SELECT CHARINDEX('r', 'Reinvent yourself') AS motto;
```

result:

```
1 motto  
2 ----  
3 13
```

❸ The following query finds the first occurrence of the substring 'self' and returns its first character's index number. As you see that there are two 'self's. `CHARINDEX()` function only finds the first occurrence which is the 'self' inside the 'yourself'.

query:

```
1 SELECT CHARINDEX('self', 'Reinvent yourself and ourself') AS  
   motto;
```

result:

```
1 motto  
2 -----  
3 14
```



④ But the following query find second 'self' by using the optional parameter [start location]

query:

```
1 SELECT CHARINDEX('self', 'Reinvent yourself and ourself', 15)
   AS motto;
```

result:

```
1 motto
2 -----
3 26
```

**PATINDEX(%pattern%, input string)**

The **PATINDEX()** function returns the starting position of the first occurrence of a pattern in a specified expression, or zeros if the pattern is not found, on all valid text and character data types.

**PATINDEX()** function takes two arguments:

**pattern:** Is a character expression that contains the sequence to be found. The **%** character must come before and follow pattern (except when you search for first or last characters).

**input string:** Is a character string data that is searched for the specified pattern

#### 💡 Tips:

- If either pattern or expression is **NULL**, **PATINDEX()** returns **NULL**.
- The starting position for **PATINDEX()** is 1.
- **PATINDEX** works just like **LIKE**, so you can use any of the wildcards. You do not have to enclose the pattern between percents. Unlike **LIKE**, **PATINDEX()** returns a position, similar to what **CHARINDEX()** does..

Examples:

query:

```
1 SELECT PATINDEX('%ern%', 'this is not a pattern') AS sample
```

result:

```
1 sample
2 -----
3 19
```

If we don't use the **%** character end of the pattern:

query:

```
1 SELECT PATINDEX('%ern', 'this is not a pattern') AS sample
```



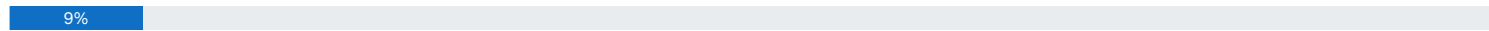
result:

|   |        |  |
|---|--------|--|
| 1 | sample |  |
| 2 | -----  |  |
| 3 | 19     |  |

Previous

Next

You have completed 9% of the lesson



Jump to... 

[Reset user tour on this page](#)

  
WAY TO REINVENT YOURSELF

© 2021 Copyright: Clarusway.com