



SQL Standard

SQL REGEXP_REPLACE

7 months ago • by John Otieno

Text data or strings, as the developers call them, is a major building block for any functional program. This is no different when it comes to storing the data. Nearly all databases contain some form of text information such as names, logs, etc.

As such, string manipulation is a common task which involves manipulating and transforming the string values to a specific format.

One of the most powerful functions in SQL that deals with string operations is the REGEXP_REPLACE() function. This function allows us to perform the regular expression-based search and replace. If you are familiar with regular expression, you know how powerful this function can be.

In this tutorial, we will learn how we can use this function to search and replace the strings in SQL database.

SQL REGEXP_REPLACE

The SQL REGEXP_REPLACE() is a function that allows us to perform the regular expression-based pattern matching and replacement within a given string.

Regular expression, or regex, is a set pattern and placeholders that allows us to match and manipulate the strings or substrings that follow a specific pattern.

It is good to keep in mind that each database engine may slightly implement the syntax and functionality of the function.

However, we can express its syntax as follows:

```
REGEXP_REPLACE(input_string, pattern, replacement [, flags])
```

The function parameters are expressed in the following:

1. `input_string` – This specifies the string within which we wish to search and replace.

2. Pattern – This specifies the regular expression pattern that we want to match within the input string.
3. Replacement – This specifies the string that will replace the matched substrings.
4. Flags – This is a set of optional flags that can help to modify the functionality of the regular expression. For example, we can enable the global search, case-insensitive matching, etc. This feature varies depending on the database engine.

Examples:

To better understand how this function works, let us look at some examples on how to use it.

Example 1: Basic Usage

Suppose we have a table that contains the employee information as shown in the following example output:

Consider a case where we want to replace the occurrence of the “Charlie” string to “Matthew”. We can use the query as follows:

```
SELECT  
REGEXP_REPLACE(first_name, 'Charlie', 'Matthew') AS new_name  
FROM  
employees;
```

The given example demonstrates a basic search and replace to find the “Charlie” string from the “first_name” column and replace it with “Matthew”.

Output:

Example 2: Case Insensitive Replacement

In some cases, you may want to perform a case insensitive search. This means that the function will only look at the string content and not the actual casing of the alphanumeric letters.

In such a case, we use the “i” as the function flag as follows:

```
SELECT REGEXP_REPLACE(product_description, Samsung, Apple, 'i') AS modified  
FROM products;
```

By setting the flag to “i”, the function matches all words that match “Samsung”, regardless of the casing.

Conclusion

In this example, we explored how to use and work with the REGEXP_REPLACE() function to perform the regular expression pattern-based search and replace.

ABOUT THE AUTHOR



John Otieno

My name is John and am a fellow geek like you. I am passionate about all things computers from Hardware, Operating systems to Programming. My dream is to share my knowledge with the world and help out fellow geeks. Follow my content by subscribing to LinuxHint mailing list

[View all posts](#)

RELATED LINUX HINT POSTS

Add a Column to the Table in SQL
Compare Two Tables in SQL
Combine Two Columns in SQL
SQL Having Clause
SQL Subtract
SQL Select All Except
SQL Outer Join

Linux Hint LLC, editor@linuxhint.com
1210 Kelly Park Circle, Morgan Hill, CA
95037
[Privacy Policy](#) and [Terms of Use](#)