

# SUBSTITUTE Function

## Contents:

- *Basic Usage*
- *Syntax and Arguments*
  - *string\_source*
  - *string\_pattern*
  - *string\_replacement*
  - *pattern\_before*
  - *pattern\_after*
- *Examples*
  - *Example - Partial obfuscation of credit card numbers*

---

Replaces found string literal or pattern from a source with a string.

Input can be specified as a column reference, a function returning a string, or a string literal, although string literal usage is rare.

- A column reference can refer to a column of String type.
- If no match is found, the function returns the source string.
- If multiple matches are found in a single string, all replacements are made.

## Basic Usage

### Column reference example:

```
set col:myURL value:SUBSTITUTE(myURL,`{ip-address}`,myDomain)
```

**Output:** Searches the `myURL` column values for sub-strings that match valid IP addresses. Where matches are found, they are replaced with the corresponding value in the `myDomain` column.

### Function reference example:

```
set col:companyName value:SUBSTITUTE(UPPER(companyName),'ACME','New ACME')
```

**Output:** Searches the uppercase version of values from the `companyName` column for the string literal `ACME`. When found, these matches are replaced by `New ACME` in the `companyName` column.

## Syntax and Arguments

```
derive type:single value:SUBSTITUTE(string_source,string_pattern,replacement_string)
```

Argument	Required?	Data Type	Description
<code>string_source</code>	Y	string	Name of the column, a function returning a string, or string literal to be applied to the function
<code>string_pattern</code>	Y	string	String literal or pattern to find
<code>string_replacement</code>	Y	string	String literal to use as replacement
<code>pattern_before</code>	N	string	String literal or pattern to find before finding the <code>string_pattern</code> value.
<code>pattern_after</code>	N	string	String literal or pattern to find after finding the <code>string_pattern</code> value.

For more information on syntax standards, see *Language Documentation Syntax Notes*.

### string\_source

Name of the item to be searched. Valid values can be:

- String literals must be quoted ( 'Hello, World' ).
- Column reference to any type that can be inferred as a string, which encompasses all values
- Functions that return string values

Multiple values and wildcards are not supported.

#### Usage Notes:

Required?	Data Type	Example Value
Yes	String literal or column reference (String, Array, or Object)	myColumn

### string\_pattern

String literal or pattern to find. This value can be a string literal, a Trifacta® pattern, or a regular expression.

- String literals must be quoted ( 'Hello, World' ).
- Multiple values and wildcards are not supported.

#### Usage Notes:

Required?	Data Type	Example Value
Yes	String literal or pattern	'Hello'

### string\_replacement

Value with which to replacement any matched patterns. Value can be a string, a function returning string values, or a column reference containing strings.

- String literals must be quoted ( 'Hello, World' ).
- column reference to any type that can be inferred as a string, which encompasses all values.

Multiple values and wildcards are not supported.

#### Usage Notes:

Required?	Data Type	Example Value
Yes	String literal or column reference (String, Array, or Object)	'###REDACTED##'

### pattern\_before

String literal or pattern to find in a position before the pattern to match.

**Tip:** Use this parameter if there are potentially multiple instances of the pattern to match in the source.

**Usage Notes:**

Required?	Data Type	Example Value
No	String literal or pattern	<code>`{digit}{3}`</code>

**pattern\_after**

String literal or pattern to find in a position after the pattern to match.

**Tip:** Use this parameter if there are potentially multiple instances of the pattern to match in the source.

**Usage Notes:**

Required?	Data Type	Example Value
No	String literal or pattern	<code>' '</code>

**Examples**

**Tip:** For additional examples, see *Common Tasks*.

**Example - Partial obfuscation of credit card numbers**

**Source:**

Suppose you have the following transactional data, which contains customer credit card numbers.

TransactionId	CreditCardNum	AmtDollars
T001	4111-1111-1111-1111	100.29
T002	5500-0000-0000-0004	510.21
T003	3400-0000-0000-009	162.13
T004	3000-0000-0000-04	294.12

For security purposes, you wish to redact the first three sets of digits, so only the last set of digits appears.

**Transform:**

To make the substitution, you must first change the type of the column to be a string:

```
settype col: CreditCardNum type: 'String'
```

You can then use the following transform to perform the pattern-based replacement of four-digit sets that end in a dash with `XXXX`:

```
set col: CreditCardNum value: substitute(CreditCardNum, `{digit}+\-`, 'XXXX-')
```

To indicate that the column no longer contains valid information, you might choose to rename it like in the following:

rename mapping: [CreditCardNum, 'CreditCardNumOBSCURED']

**Results:**

TransactionId	CreditCardNumOBSCURED	AmtDollars
T001	XXXX-XXXX-XXXX-1111	100.29
T002	XXXX-XXXX-XXXX-0004	510.21
T003	XXXX-XXXX-XXXX-009	162.13
T004	XXXX-XXXX-XXXX-04	294.12