SUBSTRING Function

Contents:

Matches some or all of a string, based on the user-defined starting and ending index values within the string.

- Input must be a string literal value.
- Since the SUBSTRING function matches based on fixed numeric values, changes to the length or structure of a data field can cause your recipe to fail to properly execute.
- The SUBSTRING function requires numerical values for the starting and ending values. If you need to match strings using patterns, you should use the extract transform instead. See *Extract Transform*.

Wrangle vs. SQL: This function is part of Wrangle , a proprietary data transformation language. Wrangle is not SQL. For more information, see *Wrangle Language*.

Basic Usage

```
substring('Hello, World',0,5)
```

Output: Returns the string: Hello.

Syntax and Arguments

```
substring(string_val,start_index,end_index)
```

Argument	Required?	Data Type	Description
string_val	Y	string	String literal to be applied to the function
start_index	Y	integer (non-negative)	Index value for the start character from the source column or value
end_index	Y	integer (non-negative)	Index value for the end character from the source column or value

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

string_val

String constant to be searched.

- Missing string values generate missing string results.
- String constants must be quoted ('Hello, World').
- Multiple columns and wildcards are not supported.

Usage Notes:

Data Type	Required?	Example Value
String	Yes	`'This is my string.'`

start_index

Index value of the character in the string to begin the substring match.

- The index of the first character of the string is `0`.
- Value must be less than `end_index`.
- If this value is greater than the length of the string, a missing value is returned.

Usage Notes:

Data Type	Required?	Example Value
Integer (non-negative)	Yes	0

end_index

Index value of the character in the string that is one after the end the substring match.

- Value must be greater than `start_index`.
- If this value is greater than the length of the string, the end of the string is the end of match. If you know the maximum length of your data, you can use that value here.

Usage Notes:

Data Type	Required?	Example Value
Integer (non-negative)	Yes	5

Examples

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

Example - Sectional Information in Zipcodes

Source:

A US zip code contains five digits with an optional Zip+4 extension consisting of four digits. Valid zip code values can be a mixture of these formats.

Within zip code values, each digit has significance:

- Digit 1: Zip code section
- Digits 2-3: Region within section
- Digits 4-5: area or town within region
- Digits 6-9: Optional Zip+4 identifier within area or town

Here is some example data:

LastName	ZipCode

Able	94101
Baker	23502-1122
Charlie	36845

Transformation:

You are interested in the region and area or town identifiers within a zip code region. You can use the following transformations applied to the `ZipCode` column to extract this information:

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`substring(ZipCode,1,3)`

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`substring(ZipCode,3,5)`

Since the string can be five or ten characters in length, you need to use the `SUBSTRING` function in the second transformation, too. If the data is limited to five-digit zip codes, you could use the `RIGHT` function.

Results:

LastName	ZipCode	substring_ZipCode	substring_ZipCode2
Able	94101	41	01
Baker	23502-1122	35	02
Charlie	36845	68	45