

# DOMAIN Function

## Contents:

- *Basic Usage*
- *Syntax and Arguments*
  - *column\_url*
- *Examples*
  - *Example - Filter out internal users*
  - *Example - Domain, Subdomain, Host, and Suffix functions*

---

Finds the value for the domain from a valid URL. Input values must be of URL or String type.

In this implementation, a domain value is all data between 1) the protocol identifier (if present) and the sub-domain and 2) the trailing, top-level domain information (e.g. `.com`).

- For more information, see *Structure of a URL*.
- You can also extract subdomain values by function. See *SUBDOMAIN Function*.

**NOTE:** When the `DOMAIN` function parses a multi-tiered top-level domain such as `co.uk`, the output is the first part of the domain value (e.g. `co`). To return other parts of the domain information, you can use the `HOST` function. See *HOST Function*.

**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

### URL literal examples:

```
domain('http://www.example.com')
```

**Output:** Returns the value `example`.

```
domain('http://www.examp1.e.com')
```

**Output:** Returns the value `e`.

### Column reference example:

```
domain(myURLs)
```

**Output:** Returns the domain values extracted from the `myURLs` column.

## Syntax and Arguments

```
domain(column_url)
```

Argument	Required?	Data Type	Description
column_url	Y	string	Name of column or String or URL literal containing the domain value to extract

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

## column\_url

Name of the column or URL or String literal whose values are used to extract the domain value.

- Missing input values generate missing results.
- Multiple columns and wildcards are not supported.

### Usage Notes:

Required?	Data Type	Example Value
Yes	String literal or column reference (URL)	http://www.example.com

## Examples

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

### Example - Filter out internal users

Here is some example web visitor information, including the name of the individual and the referring URL. You would like to filter out the internal users, whose referrer values include `test-value`.

Name	Referrer
Joe Guy	http://www.example.com
Ian Holmes	http://www.test-value.com/support
Nick Knight	http://www.test-value.com
Axel Adams	http://www.example.com
Teri Towns	http://www.example.com/test-value

### Transformation:

The referrrrer values include `test-value` as a non-domain value and varying URLs from the `test-value.com` domain. So, you should use the `DOMAIN` function to parse only the domain versions of these values. The following evaluates the `Referrer` column values for the `test-value` domain and generates true/false answers in a new column accordingly:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>if(domain(Referrer)=='test-value',true,false)</code>
<b>Parameter: New column name</b>	<code>'isInternal'</code>

Now that these values are flagged, you can filter out the internal names:

<b>Transformation Name</b>	Filter rows
<b>Parameter: Condition</b>	Custom formula
<b>Parameter: Type of formula</b>	Custom single
<b>Parameter: Condition</b>	(isInternal == 'true')
<b>Parameter: Action</b>	Delete matching rows

## Results:

Name	Referrer	isInternal
Joe Guy	http://www.example.com	false
Axel Adams	http://www.example.com	false
Teri Towns	http://www.example.com/test-value	false

## Example - Domain, Subdomain, Host, and Suffix functions

This examples illustrates how you can extract component parts of a URL using the following functions:

- **DOMAIN** - extracts the domain value from a URL. See *DOMAIN Function*.
- **SUBDOMAIN** - extracts the first group after the protocol identifier and before the domain value. See *SUBDOMAIN Function*.
- **HOST** - returns the complete value of the host from an URL. See *HOST Function*.
- **SUFFIX** - extracts the suffix of a URL. See *SUFFIX Function*.
- **URLPARAMS** - extracts the query parameters and values from a URL. See *URLPARAMS Function*.
- **FILTEROBJECT** - filters an Object value to show only the elements for a specified key. See *FILTEROBJECT Function*.

## Source:

Your dataset includes the following values for URLs:

URL
www.example.com
example.com/support
http://www.example.com/products/
http://1.2.3.4
https://www.example.com/free-download
https://www.example.com/about-us/careers
www.app.example.com
www.some.app.example.com
some.app.example.com
some.example.com
example.com
http://www.example.com?q1=broken%20record
http://www.example.com?query=khakis&app=pants

http://www.example.com?q1=broken%20record&q2=broken%20tape&q3=broken%20wrist

### Transformation:

When the above data is imported into the application, the column is recognized as a URL. All values are registered as valid, even the IPv4 address.

To extract the domain and subdomain values:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	DOMAIN(URL)
<b>Parameter: New column name</b>	'domain_URL'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	SUBDOMAIN(URL)
<b>Parameter: New column name</b>	'subdomain_URL'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	HOST(URL)
<b>Parameter: New column name</b>	'host_URL'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	SUFFIX(URL)
<b>Parameter: New column name</b>	'suffix_URL'

You can use the Pattern in the following transformation to extract protocol identifiers, if present, into a new column:

<b>Transformation Name</b>	Extract text or pattern
<b>Parameter: Column to extract from</b>	URL
<b>Parameter: Option</b>	Custom text or pattern
<b>Parameter: Text to extract</b>	`{start}%*://`

To clean this up, you might want to rename the column to protocol\_URL.

To extract the path values, you can use the following regular expression:

**NOTE:** Regular expressions are considered a developer-level method for pattern matching. Please use them with caution. See *Text Matching*.

<b>Transformation Name</b>	Extract text or pattern
<b>Parameter: Column to extract from</b>	URL
<b>Parameter: Option</b>	Custom text or pattern
<b>Parameter: Text to extract</b>	/[!^*:\/\ ]\./.*\$/

The above transformation grabs a little too much of the URL. If you rename the column to `path_URL`, you can use the following regular expression to clean it up:

<b>Transformation Name</b>	Extract text or pattern
<b>Parameter: Column to extract from</b>	URL
<b>Parameter: Option</b>	Custom text or pattern
<b>Parameter: Text to extract</b>	/[!^\/].*\$/

Delete the `path_URL` column and rename the `path_URL1` column to the deleted one. Then:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	URLPARAMS(URL)
<b>Parameter: New column name</b>	'urlParams'

If you wanted to just see the values for the `q1` parameter, you could add the following:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	FILTEROBJECT(urlParams, 'q1')
<b>Parameter: New column name</b>	'urlParam_q1'

**Results:**

For display purposes, the results table has been broken down into separate sets of columns.

Column set 1:

URL	host_URL	path_URL
www.example.com	www.example.com	

example.com/support	example.com	/support
http://www.example.com/products/	www.example.com	/products/
http://1.2.3.4	1.2.3.4	
https://www.example.com/free-download	www.example.com	/free-download
https://www.example.com/about-us/careers	www.example.com	/about-us /careers
www.app.example.com	www.app.example.com	
www.some.app.example.com	www.some.app.example.com	
some.app.example.com	some.app.example.com	
some.example.com	some.example.com	
example.com	example.com	
http://www.example.com?q1=broken%20record	www.example.com	
http://www.example.com?query=khakis&app=pants	www.example.com	
http://www.example.com?q1=broken%20record&q2=broken%20tape&q3=broken%20wrist	www.example.com	

Column set 2:

URL	protocol_URL	subdomain_URL	domain_URL	suffix_URL
www.example.com		www	example	com
example.com/support			example	com
http://www.example.com/products/	http://	www	example	com
http://1.2.3.4	http://			
https://www.example.com/free-download	https://	www	example	com
https://www.example.com/about-us/careers	https://	www	example	com
www.app.example.com		www.app	example	com
www.some.app.example.com		www.some.app	example	com
some.app.example.com		some.app	example	com
some.example.com		some	example	com
example.com			example	com
http://www.example.com?q1=broken%20record	http://	www	example	com
http://www.example.com?query=khakis&app=pants	http://	www	example	com
http://www.example.com?q1=broken%20record&q2=broken%20tape&q3=broken%20wrist	http://	www	example	com

Column set 3:

URL	urlParams	urlParam_q1
www.example.com		
example.com/support		
http://www.example.com/products/		
http://1.2.3.4		

https://www.example.com/free-download		
https://www.example.com/about-us/careers		
www.app.example.com		
www.some.app.example.com		
some.app.example.com		
some.example.com		
example.com		
http://www.example.com?q1=broken%20record	{"q1":"broken record"}	{"q1":"broken record"}
http://www.example.com?query=khakis&app=pants	{"query":"khakis","app":"pants"}	
http://www.example.com?q1=broken%20record&q2=broken%20tape&q3=broken%20wrist	{"q1":"broken record", "q2":"broken tape", "q3":"broken wrist"}	{"q1":"broken record"}