

FILTEROBJECT Function

Contents:

- *Basic Usage*
- *Syntax and Arguments*
 - *obj_col*
 - *keys*
- *Examples*
 - *Example - Parsing query parameters from URLs*

Filters the keys and values from an Object data type column based on a specified key value.

- A single field value of an Object data type must have unique keys. Values may, however, be repeated.
- The order of key-value pairs is not guaranteed.

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

Object literal reference example:

```
filterobject('q','hello','r','there': 'q','world');
```

Output: Returns an Object of key-value pairs for the `q` key:

```
{"q":["hello", "world"]}
```

Column reference example:

```
filterobject(myObjects, ['k1,k2'])
```

Output: Returns an Object of key-value pairs for all instances of the `k1` and `k2` keys.

Syntax and Arguments

```
filterobject(obj,keys)
```

Argument	Required?	Data Type	Description
obj_col	Y	String or Object	Name of column, function returning an Object, or Object literal to be filtered
keys	Y	Array	Array representing the keys to filter. Each element can be a String, function returning a String, or a reference to a column of String values.

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

obj_col

Object literal, name of the Object column, or function returning an Object whose keys you want to extract into an array.

Usage Notes:

Required?	Data Type	Example Value
Yes	Object literal, function, or column reference	myObj

keys

This parameter contains an Array of Strings, each of which represents a key whose values are to be returned with the key as the output of the function.

- For a single key, this value can be a regular String value.
- For multiple keys, this value is an Array of String values.

Usage Notes:

Required?	Data Type	Example Value
Yes	String or Array	['key1 ' , 'key2 ' , 'key3 ']

Examples

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

Example - Parsing query parameters from URLs

This examples illustrates how you can extract component parts of a URL using specialized functions for the URL data type.

Functions:

Item	Description
DOMAIN Function	Finds the value for the domain from a valid URL. Input values must be of URL or String type.
SUBDOMAIN Function	Finds the value a subdomain value from a valid URL. Input values must be of URL or String type.
HOST Function	Finds the host value from a valid URL. Input values must be of URL or String type and can be literals or column references.
SUFFIX Function	Finds the suffix value after the domain from a valid URL. Input values must be of URL or String type.
URLPARAMS Function	Extracts the query parameters of a URL into an Object. The Object keys are the parameter's names, and its values are the parameter's values. Input values must be of URL or String type.
FILTEROBJECT Function	Filters the keys and values from an Object data type column based on a specified key value.

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 numeric address.

To extract the domain and subdomain values:

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

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

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

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

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

Transformation Name	Extract text or pattern
Parameter: Column to extract from	URL
Parameter: Option	Custom text or pattern
Parameter: Text to extract	`{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*.

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

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:

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

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

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	URLPARAMS(URL)

Parameter: New column name	'urlParams'
-----------------------------------	-------------

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

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	FILTEROBJECT(urlParams, 'q1')
Parameter: New column name	'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"}