

FILTEROBJECT Function

Contents:

- *Basic Usage*
 - *Syntax*
 - *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.
- For more information, see *Object Data Type*.

Basic Usage

Object literal reference example:

```
derive type:single value:FILTEROBJECT('{ "q": "hello", "r", "there": "q", "world" }', 'q') as: 'q'
```

Output: Generates a new `q` column containing an Object of key-value pairs for the `q` key:

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

Column reference example:

```
derive type:single value:FILTEROBJECT(myObjects, '[k1,k2]') as: 'k-values'
```

Output: Generates a new `k-values` column containing an Object of key-value pairs for all instances of the `k1` and `k2` keys.

Syntax

```
derive type:single value:FILTEROBJECT(obj, 'keys')
```

Argument	Required?	Data Type	Description
<code>obj_col</code>	Y	String or Object	Name of column, function returning an Object, or Object literal to be filtered
<code>keys</code>	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

Example - Parsing query parameters from URLs

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

Transform:

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:

```
derive type:single value: DOMAIN(URL) as: 'domain_URL'
```

```
derive type:single value: SUBDOMAIN(URL) as: 'subdomain_URL'
```

```
derive type:single value: HOST(URL) as: host_URL'
```

```
derive type:single value: SUFFIX(URL) as: 'suffix_URL'
```

You can use the Trifacta® pattern in the following transform to extract protocol identifiers, if present, into a new column:

```
extract col:URL on: `{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*.

```
extract col: URL on: /^[^*:\//\]\.\.*$/
```

The above transform 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:

```
extract col:path_URL on: /[^^\//].*$/
```

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

```
derive type:single value: URLPARAMS(URL) as: 'urlParams'
```

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

```
derive type:single value: FILTEROBJECT(urlParams,'q1') as: '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"}