

# KEYS Function

## Contents:

- *Basic Usage*
- *Syntax and Arguments*
  - *obj\_col*
- *Examples*
  - *Example - Basic keys example*
  - *Example - Create an Object of product properties*

---

Extracts the key values from an Object data type column and stores them in an array of String values.

## Basic Usage

### Column reference example:

```
derive type:single value:KEYS('object1') as:'objKeys'
```

**Output:** Generates a new `objKeys` column containing an array of all of the keys found in the key-value Objects found in the `object1` column.

### Array literal reference example:

See the example below.

## Syntax and Arguments

```
derive type:single value:KEYS(obj_col)
```

Argument	Required?	Data Type	Description
obj_col	Y	String or Object	Name of column or Object literal whose keys are to be extracted into an array

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

### obj\_col

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

### Usage Notes:

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

## Examples

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

## Example - Basic keys example

### Source:

Following dataset contains configuration blocks for individual features. These example blocks are of Object type.

Code formatting has been applied to the Object data to improve legibility.

FeatureName	Configuration
Whiz Widget	<pre>{   "enabled": "true",   "maxRows": "1000",   "maxCols": "100" }</pre>
Magic Button	<pre>{   "enabled": "false",   "startDirectory": "/home",   "maxDepth": "15" }</pre>
Happy Path Finder	<pre>{   "enabled": "true" }</pre>

### Transform:

The following transform extracts the key values from the Object data in the Configuration column.

```
derive type:single value: KEYS(Configuration) as: 'keys_Configuration'
```

### Results:

The keys\_Configuration column contains the arrays of the key values.

FeatureName	Configuration	keys_Configuration
Whiz Widget	<pre>{   "enabled": "true",   "maxRows": "1000",   "maxCols": "100" }</pre>	<pre>["enabled", "maxRows", "maxCols"]</pre>
Magic Button	<pre>{   "enabled": "false",   "startDirectory": "/home",   "maxDepth": "15" }</pre>	<pre>["enabled", "startDirectory", "maxDepth"]</pre>

Happy Path Finder	<pre>{   "enabled": "true" }</pre>	<pre>[ "enabled" ]</pre>
-------------------	------------------------------------	--------------------------

### Example - Create an Object of product properties

#### Source:

Your dataset contains master product data with product properties stored in two arrays of keys and values.

ProdId	ProdCategory	ProdName	ProdKeys	ProdProperties
S001	Shirts	Crew Neck T-Shirt	["type", "color", "fabric", "sizes"]	["crew","blue","cotton","S,M,L","in stock","padded"]
S002	Shirts	V-Neck T-Shirt	["type", "color", "fabric", "sizes"]	["v-neck","white","blend","S,M,L,XL","in stock","discount - seasonal"]
S003	Shirts	Tanktop	["type", "color", "fabric", "sizes"]	["tank","red","mesh","XS,S,M","discount - clearance","in stock"]
S004	Shirts	Turtleneck	["type", "color", "fabric", "sizes"]	["turtle","black","cotton","M,L,XL","out of stock","padded"]

#### Transform:

When the above data is loaded into the Transformer page, you might need to clean up the two array columns.

Using the following transform, you can map the first element of the first array as a key for the first element of the second, which is its value. You might notice that the number of keys and the number of values are not consistent. For the extra elements in the second array, the default key of `ProdMiscProperties` is used:

```
derive type:single value: ARRAYSTOMAP(ProdProperties, ProdValues, 'ProdMiscProperties')
as: 'prodPropertyMap'
```

You can now use the following steps to generate a new version of the keys:

```
drop col:ProdKeys
```

```
derive type:single value:KEYS(prodPropertyMap) as:'ProdKeys'
```

#### Results:

ProdId	ProdCategory	ProdName	ProdKeys	ProdProperties	prodPropertyMap
S001	Shirts	Crew Neck T-Shirt	["type", "color", "fabric", "sizes", "ProdMiscProperties"]	["crew","blue","cotton","S,M,L","in stock","padded"]	<pre>{   "type": [ "crew" ],   "color": [ "blue" ],   "fabric": [ "cotton" ],   "sizes": [ "S,M,L" ],   "ProdMiscProperties": [ "in stock", "padded" ] }</pre>

S002	Shirts	V-Neck T-Shirt	["type", "color", "fabric", "sizes", "ProdMiscProperties"]	["v-neck", "white", "blend", "S,M,L,XL", "in stock", "discount - seasonal"]	{ "type": [ "v-neck" ], "color": [ "white" ], "fabric": [ "blend" ], "sizes": [ "S", "M", "L", "XL" ], "ProdMiscProperties": [ "in stock", "discount - seasonal" ] } }
S003	Shirts	Tanktop	["type", "color", "fabric", "sizes", "ProdMiscProperties"]	["tank", "red", "mesh", "XS,S,M", "discount - clearance", "in stock"]	{ "type": [ "tank" ], "color": [ "red" ], "fabric": [ "mesh" ], "sizes": [ "XS", "S", "M" ], "ProdMiscProperties": [ "discount - clearance", "in stock" ] } }
S004	Shirts	Turtleneck	["type", "color", "fabric", "sizes", "ProdMiscProperties"]	["turtle", "black", "cotton", "M,L,XL", "out of stock", "padded"]	{ "type": [ "turtleneck" ], "color": [ "black" ], "fabric": [ "cotton" ], "sizes": [ "M", "L", "XL" ], "ProdMiscProperties": [ "out of stock", "padded" ] } }