

Case Transform

Contents:

- *Basic Usage*
- *Syntax and Parameters*
 - *if*
 - *then*
 - *else*
 - *col*
 - *colCases*
 - *cases*
 - *default*
 - *as*
- *Examples*

NOTE: Transforms are a part of the underlying language, which is not directly accessible to users. This content is maintained for reference purposes only. For more information on the user-accessible equivalent to transforms, see *Transformation Reference*.

Performs conditional transformation of data with a single statement using if-then-else logic or with multiple statements using case logic. Results are generated in a new column.

NOTE: If you are running your job on Spark, avoid creating single conditional transformations with deeply nested sets of conditions. On Spark, these jobs can time out, and deeply nested steps can be difficult to debug. Instead, break up your nesting into smaller conditional transformations of multiple steps.

There are function equivalents to this transform:

- *IF Function*
- *CASE Function*

Basic Usage

Example - if/then/else

This example illustrates a single if/then/else construction:

```
case if: testScore >= 60 then: 'yes' else: 'no' as: 'passedTest'
```

Output: If a value in the `testScore` is greater than or equal to 60, a value of `yes` is written into the new `passedTest` column. Otherwise, a value of `no` is written.

Example - Case (single column)

This example shows how to step through a sequence of case tests applied to a single column.

```
case col: custName colCases: ['Big Co',0.2],['Little Guy Ltd',0.05] default: 0 as:  
'discountRate'
```

Output: Checks names in the `custName` column and writes discount values based on exact matches of values in the column:

custName value	discountRate
Big Co	0.2
Little Guy Ltd	0.05
default (if no matches)	0

Example - Case (custom conditions)

The following example illustrates how to construct case transforms with multiple independent conditions. Tests can come from arbitrary columns and expressions.

- The first case is tested:
 - If `true`, then the listed value is written to the new column.
 - If `false`, then the next case is tested.
- If none of the stated cases evaluates to `true`, then the default value is written.

```
case cases: [totalOrdersQ3 < 10, true], [lastOrderDays > 60, true] default: false as: 'sendCheckinEmail'
```

Output: If the total orders in Q3 < 10 OR the last order was placed more than 60 days ago, then write `true` in the `sendCheckinEmail`. Otherwise, write `false`.

Logic	Test	SendCheckinEmail
if	<code>totalOrdersQ3 < 10</code>	<code>true</code>
if above is false	<code>lastOrderDays > 60</code>	<code>true</code>
if above is false	write default	<code>false</code>

Syntax and Parameters

```
case [if: if_expression] [then:'str_if_true'] [else:'str_if_false'] [col:col1] [colCases: [[Match1,Val1]], [[Match2,Val2]]] [cases: [[Exp3,Val3]], [[Exp4,Val4]]] [default:default_val] as: 'new_column_name'
```

Token	Required?	Data Type	Description
<code>case</code>	Y	transform	Name of the transform
<code>if</code>	N	string	(For single if/then/else) Expression that is tested must evaluate to <code>true</code> or <code>false</code> .
<code>then</code>	N	string	(For single if/then/else) Value written to the new column if the if expression is <code>true</code> .
<code>else</code>	N	string	(For single if/then/else) Value written to the new column if the if expression is <code>false</code> .
<code>col</code>	N	string	(For single-column case) Name of column whose values are to be tested.
<code>colCases</code>	N	comma-separated arrays	(For single-column case) Matrix of string-value pairs: <ul style="list-style-type: none"> • First entry is the value to match. • Second entry is the value written to the new column if a match appears

cases	N	comma-separated arrays	(For custom conditions case) Matrix of expression-value pairs: <ul style="list-style-type: none"> • First entry is the expression to evaluate. • Second entry is the value to write if the expression is <code>true</code>.
default	N	any	(For single-column case and custom condition case) If no matches are made, this value is written to the new column.
as	Y	string	Name of the new column where results are written.

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

if

For if-then-else condition types, this value is an expression to test. Expression must evaluate to `true` or `false`.

Usage Notes:

Required?	Data Type
Required for if-the-else condition type	String (expression)

then

For if-then-else condition types, this value is a literal value to write in the output column if the expression evaluates to `true`.

Usage Notes:

Required?	Data Type
Required for if-the-else condition type	String or other literal type

else

For if-then-else condition types, this value is a literal value to write in the output column if the expression evaluates to `false`.

Usage Notes:

Required?	Data Type
Required for if-the-else condition type	String or other literal type

col

For single-case condition types, this value identifies the column to test.

Usage Notes:

Required?	Data Type
Required for single-case condition type	String (column name)

colCases

For single-case condition types, this parameter contains a comma-separated set of two-value arrays.

- **Array value 1:** A literal value to match in the specified column.
- **Array value 2:** If the value is matched, this value is written into the output column.

You can specify one or more cases as comma-separated two-value arrays.

Usage Notes:

Required?	Data Type
Required for single-case condition type	Array (comma-separated list)

cases

For multi-case condition types, this parameter contains a comma-separated set of two-value arrays.

- **Array value 1:** An expression to test, which must evaluate to `true` or `false`.
- **Array value 2:** If the value is matched, this value is written into the output column.

You can specify one or more cases as comma-separated two-value arrays.

Usage Notes:

Required?	Data Type
Required for single-case condition type	Array (comma-separated list)

default

For single-case and multi-case condition types, this parameter defines the value to write in the new column if none of the cases yields a `true` result.

Usage Notes:

Required?	Data Type
Required for single-case condition type	Literal of any data type

as

Name of the new column that is being generated. If the `as` parameter is not specified, a default name is used.

Usage Notes:

Required?	Data Type
Yes	String (column name)

Examples

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

See above.