Logical Operators

Contents:

Logical operators (and, or, not) enable you to logically combine multiple expressions to evaluate a larger, more complex expression whose output is `true` or `false`.

```
(left-hand side) (operator) (right-hand side)
```

These evaluations result in a Boolean output. The following operators are supported:

Operator Name	Symbol	Example Expression	Output	Notes
and	`&&`	`((1 == 1) && (2 == 2))`	`true`	
		`((1 == 1) && (2 == 3))`	`false`	
or	`||`	`((1 == 1) || (2 == 2))`	`true`	Exclusive or (xor) is not supported.
		`((1 == 2) || (2 == 3))`	`false`	
not	!	`!(1 == 1)`	`false`	
		`!(1 == 2)`	`true`	

The above examples apply to integer values only. Below, you can review how the comparison operators apply to different data types.

Usage

Logical operators are used to perform evaluations of expressions covering a variety of data types. Typically, they are applied in evaluations of values or rows.

Example data:

X	Y
true	true
true	false
false	true
false	false

Transforms:

```
derive type:single value:(X && Y) as: 'col_and'
```

```
derive type:single value:(X || Y) as: 'col_or'
```

```
derive type:single value:!(or) as: 'col_not_and'
```

```
derive type:single value:!(or) as: 'col_not_or'
```

Results:

Your output looks like the following:

X	Y	col_and	col_or	col_not_and	col_not_or
true	true	true	true	false	false
true	false	false	true	true	false
false	true	false	true	true	false
false	false	false	false	true	true

Examples

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

and

Column Type	Example Transform	Output	Notes
Integer /Decimal	`set col:InRange value:((Input >= 10) && (Input <= 90))`	• Set the value of the `InRange` column to `true` if the value of the `Input` column is between 10 and 90, inclusive. • Otherwise, `InRange` column is `false`.	
Datetime	`delete row: ((Date >= DATE(2014, 01, 01)) && (Date <= DATE(2014, 12, 31))`	Delete all rows in which the `Date` value falls somewhere in 2014.	
String	`derive type:single value:((LEFT (USStates,1) == "A") && (RIGHT (USStates,1) == "A"))`	For U.S. State names, the generated column contains `true` for the following values: • `Alabama` • `Alaska` • `Arizona` For all other values, the generated value is `false`.	• See *LEFT Function*. • See *RIGHT Function*.

or

Column Type	Example Transform	Output	Notes

Integer /Decimal	`set col:BigOrder value:((Total > 1000000) || (Qty > 1000))`	• In the `BigOrder` column, set the value to `true` if the value of `Total` is more than 1,000,000 or the value of `Qty` is more than 1000. • Otherwise, the value is `false`.	
Datetime	`delete row: ((Date <= DATE(1950, 01, 01)) || (Date >= DATE(2050, 12, 31))`	Delete all rows in the dataset where the `Date` value is earlier than 01/01/1950 or later than 12/31/2050.	
String	`derive type:single value:((Brand == 'subaru') || ('Color' == 'green')) as:'good_car'`	• Generate the new `good_car` column containing `true` if the `Brand` is `subaru` or the `Color` is `green`. • Otherwise, the `good_car` value is `false`.	

not

Column Type	Example Transform	Output	Notes
Integer /Decimal	`keep row:!((sqft < 1300) && (bath < 2) && (bed < 2.5))`	Keep all rows for houses that do not meet any of these criteria: • smaller than 1300 square feet, • less than 2 bathrooms, • less than 2.5 bedrooms.	
Datetime	`keep row:!(YEAR(Date) == '2016')`	Keep all rows in the dataset where the year of the `Date` value is not 2016.	
String	`delete row:!(status == 'Keep_It')`	Delete all rows in which the value of the `status` column is not `Keep_It`.	