EQUAL Function

Returns `true` if the first argument is equal to the second argument. Equivalent to the = operator.

- Each argument can be a literal String, Integer or Decimal number, a function, or a column reference.

Since the function returns a Boolean value, it can be used as a function or a conditional.

> **NOTE:** Within an expression, you might choose to use the corresponding operator, instead of this function. For more information, see *Comparison Operators*.

Basic Usage

```
derive type:single value: IF(EQUAL(errorCount, 0),'ok','Error_recorded') as:'status'
```

Output: If the value in the `errorCount` column is zero, then the `status` column value is `ok`. Otherwise, the value is `Error_recorded`.

Syntax and Arguments

```
derive type:single value:EQUAL(value1, value2)
```

Argument	Required?	Data Type	Description
value1	Y	string	The first value. This value can be a String, a number, a function, or a column reference.
value2	Y	string	The second value. This value can be a String, a number, a function, or a column reference.

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

value1, value2

Names of the columns, expressions, or literals to compare.

- Missing values generate missing string results.

Usage Notes:

Required?	Data Type	Example Value
Yes	Column reference, function, or numeric or String value	`myColumn`

Examples

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

Example - Basic Equal and Notequal Functions

This example demonstrate the following comparison functions.

- See *EQUAL Function*.
- See *NOTEQUAL Function*.
- See *ISEVEN Function*.
- See *ISODD Function*.

In this example, the dataset contains current measurements of the sides of rectangular areas next to the size of those areas as previously reported. Using these functions, you can perform some light analysis of the data.

Source:

sideA	sideB	reportedArea
4	14	56
6	6	35
8	4	32
15	15	200
4	7	28
12	6	70
9	9	81

Transform:

In the first test, you are determining if the four-sided area is a square, based on a comparison of the measured values for `sideA` and `sideB`:

```
derive type:single value:EQUAL(sideA, sideB) as:'isSquare'
```

Next, you can use the reported sides to calculate the area of the shape and compare it to the area previously reported:

```
derive type:single value:NOTEQUAL(sideA * sideB, reportedArea) as:'isValidData'
```

You can also compute if the reportedArea can be divided into even square units:

```
derive type:single value:ISEVEN(reportedArea) as:'isReportedAreaEven'
```

You can test if either measured side is an odd number of units:

```
derive type:single value:IF((ISODD(sideA) == true) OR (ISODD(sideB) == true),TRUE,FALSE)
as:'isSideOdd'
```

Results:

sideA	sideB	reportedArea	isSquare	isValidData	isReportedAreaEven	isSideOdd
4	14	56	FALSE	FALSE	TRUE	FALSE
6	6	35	TRUE	TRUE	TRUE	FALSE
8	4	32	FALSE	FALSE	TRUE	FALSE
15	15	200	TRUE	TRUE	TRUE	TRUE
4	7	28	FALSE	FALSE	TRUE	TRUE
12	6	70	FALSE	TRUE	TRUE	FALSE

9	9	81	TRUE	FALSE	FALSE	FALSE