

TRUNC Function

Contents:

- *Basic Usage*
 - *Syntax and Arguments*
 - *numeric_value*
 - *integer_value*
 - *Examples*
 - *Example - Basic TRUNC*
 - *Example - RANDBETWEEN, PI, and TRUNC functions*
-

Removes all digits to the right of the decimal point for any value. Optionally, you can specify the number of digits to which to round. Input can be an Integer, a Decimal, a column reference, or an expression.

Basic Usage

Numeric literal example:

```
derive type:single value: TRUNC(PI())
```

Output: Generates a column with each row's value 3.

Expression example:

```
derive type:single value: TRUNC(length_in * length_in, 2)
```

Output: Generates a column containing the square of the values in `length_in`, truncated to two decimal points.

Syntax and Arguments

```
derive type:single value: TRUNC(numeric_value, integer_value)
```

Argument	Required?	Data Type	Description
<code>numeric_value</code>	Y	string, decimal, or integer	Name of column or Decimal or Integer literal to apply to the function
<code>integer_value</code>	N	integer	Number of digits to which to truncate. <ul style="list-style-type: none">• Default is 0, which truncates to the nearest integer.• Negative integer values can be applied.

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

numeric_value

Name of the column, numeric literal, or numeric expression.

- Missing input values generate missing results.
- Literal numeric values should not be quoted. Quoted values are treated as strings.
- Multiple columns and wildcards are not supported.

Usage Notes:

Required?	Data Type	Example Value
Yes	String (column reference) or Integer or Decimal literal	2 . 5

integer_value


Number of digits to which to round the first argument of the function.

- Positive values truncate to the right of the decimal point.
- Negative values truncate to the left of the decimal point.
- Missing input values generate missing results.

Usage Notes:

Required?	Data Type	Example Value
No	Integer literal	3

Examples

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

Example - Basic TRUNC

Source:

RowId	myVal
r01	1.2345
r02	-1.2345
r03	100.000
r04	10.1
r05	50.029

Transform:

```
derive type:single value:TRUNC(myVal) as: 'trunc_myVal'
```

```
derive type:single value:TRUNC(myVal,2) as: 'trunc_myVal2'
```

```
derive type:single value:TRUNC(myVal,2) as: 'trunc_myVal_2'
```

Results:

RowId	myVal	trunc_myVal	trunc_myVal2	trunc_myVal_2
r01	1.2345	1	1.23	0
r02	-1.2345	-1	-1.23	0
r03	100.000	100	100.00	100

r04	10.1	10	10.1	0
r05	50.029	50	50.02	0

Example - RANDBETWEEN, PI, and TRUNC functions

This example illustrates how you can apply the following functions to generate new and random data in your dataset:

- **RANDBETWEEN** - Generate a random Integer value between two specified Integers. See *RANDBETWEEN Function*.
- **PI** - Generate the value of pi to 15 decimal points. See *PI Function*.
- **ROUND** - Round a decimal value to the nearest Integer or to a specified number of digits. See *ROUND Function*.
- **TRUNC** - Round a value down to the nearest Integer value. See *TRUNC Function*.

Source:

In the following example, a company produces 10 circular parts, the size of which is measured in each product's radius in inches.

prold	radius_in
p001	1
p002	2
p003	3
p004	4
p005	5
p006	6
p007	7
p008	8
p009	9
p010	10

Based on the above data, the company wants to generate some additional sizing information for these circular parts, including the generation of two points along each part's circumference where quality stress tests can be applied.

Transform:

To begin, you can use the following steps to generate the area and circumference for each product, rounded to three decimal points:

```
derive type:single value: ROUND(PI() * (POW(radius_in, 2)), 3) as: 'area_sqin'
```

```
derive type:single value: ROUND(PI() * (2 * radius_in), 3) as: 'circumference_in'
```

For quality purposes, the company needs two tests points along the circumference, which are generated by calculating two separate random locations along the circumference. Since the **RANDBETWEEN** function only calculates using Integer values, you must first truncate the values from `circumference_in`:

```
derive type:single value: TRUNC(circumference_in) as: 'trunc_circumference_in'
```

Then, you can calculate the random points using the following:

```
derive type:single value: RANDBETWEEN(0, trunc_circumference_in) as: 'testPt01_in'
```

```
derive type:single value: RANDBETWEEN(0, trunc_circumference_in) as: 'testPt02_in'
```

Results:

After the `trunc_circumference_in` column is dropped, the data should look similar to the following:

prold	radius_in	area_sq_in	circumference_in	testPt01_in	testPt02_in
p001	1	3.142	6.283	5	5
p002	2	12.566	12.566	3	3
p003	3	28.274	18.850	13	13
p004	4	50.265	25.133	24	24
p005	5	78.540	31.416	0	0
p006	6	113.097	37.699	15	15
p007	7	153.938	43.982	11	11
p008	8	201.062	50.265	1	1
p009	9	254.469	56.549	29	29
p010	10	314.159	62.832	21	21