

RANDBETWEEN Function

Generates a random integer between a low and a high number. Two inputs may be Integer or Decimal types, functions returning these types, or column references.

- Range is inclusive of the two parameter values.
- The first parameter must be the lower value in the range.

Basic Usage

```
derive type:single value: RANDBETWEEN(1,10) as:'r10'
```

Output: For each row, generate a random Integer value between 1 and 10 in the new `r10` function.

Syntax

```
derive type:single value: RANDBETWEEN(value1,value2) as:'random'
```

| Argument | Required? | Data Type | Description |
|----------|-----------|--------------------|--|
| value1 | Y | Integer or Decimal | Integer or Decimal literal, function returning one of these data types, or a column reference for the lower boundary of the range. Range is inclusive of this value. |
| value2 | Y | Integer or Decimal | Integer or Decimal literal, function returning one of these data types, or a column reference for the upper boundary of the range. Range is inclusive of this value. |

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

value1, value2

Literals, functions, or column references to Integer or Decimal values that are used as the lower and upper bounds, respectively, for the range.

- Missing input values generate missing results.
- Multiple columns and wildcards are not supported.

Usage Notes:

| Required? | Data Type | Example Value |
|-----------|---|---------------|
| Yes | Integer or Decimal literal, function, or column reference | 100 |

Examples

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.

| prodlid | 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:

| prodlid | 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 |