

# LCM Function

## Contents:

- *Basic Usage*
- *Syntax and Arguments*
  - *value1, value2*
- *Examples*
  - *Example - Basic LCM function*
  - *Example - Padding values*

---

Returns the least common multiple shared by the first and second arguments.

- Each argument can be a literal Integer number, a function returning an Integer, or a reference to a column containing Integer values.

## Basic Usage

```
derive type:single value: LCM(10,4) as:'twenty'
```

**Output:** The least common multiple between values 10 and 4 is 20 and is stored in a new column called `twenty`.

## Syntax and Arguments

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

Argument	Required?	Data Type	Description
value1	Y	string	The first value must be an Integer literal, column reference, or expression that evaluates to an Integer value.
value2	Y	string	The first value must be an Integer literal, column reference, or expression that evaluates to an Integer value.

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

## value1, value2

Integer expressions, column references or literals to multiply together.

- Missing or mismatched values generate missing string results.

## Usage Notes:

Required?	Data Type	Example Value
Yes	Literal, function, or column reference returning an Integer value	15

## Examples

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

### Example - Basic LCM function

#### Source:

string	repeat_count
ha	0
ha	1
ha	1.5
ha	2
ha	-2

#### Transform:

```
derive type:single value: repeat(string, repeat_count) as: 'repeat_string'
```

#### Results:

string	repeat_count	repeat_string
ha	0	
ha	1	ha
ha	1.5	
ha	2	haha
ha	-2	

### Example - Padding values

In the following example, the imported `prodId` values are supposed to be eight characters in length. Somewhere during the process, however, leading 0 characters were truncated. The steps below allow you to re-insert the leading characters.

#### Source:

prodName	prodId
w01	1
w02	10000001
w03	345
w04	10402

#### Transform:

First, we need to identify how many zeroes need to be inserted for each `prodId`:

```
derive type:single value: 8 - len(prodId) as: 'len_prodId'
```

Use the REPEAT function to generate a pad string based on the above values:

```
derive type:single value: repeat('0', len_prodId) as: 'padString'
```

Merge the pad string and the original prodId column:

```
merge col: padString,prodId as: 'column2'
```

### Results:

When you delete the intermediate columns and rename column2 to prodId, you have the following table:

prodName	prodId
w01	00000001
w02	10000001
w03	00000345
w04	00010402