ADD Function

Returns the value of summing the first argument and the second argument. Equivalent to the + operator.

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

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

Basic Usage

```
derive type:single value: ADD(2,3) as:'five'
```

Output: Sums the values 2 and 3 and stores the value in a new column called `five`.

Syntax and Arguments

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

Argument	Required?	Data Type	Description
value1	Y	string	The first value must be an Integer or Decimal literal, column reference, or expression that evaluates to one of those two numeric types.
value2	Y	string	The first value must be an Integer or Decimal literal, column reference, or expression that evaluates to one of those two numeric types.

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

value1, value2

Integer or Decimal expressions, column references or literals to sum 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 or Decimal value	`myScore * 10`

Examples

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

Example - Numeric Functions

This example demonstrate the following numeric functions:

- See *ADD Function*.
- See *SUBTRACT Function*.
- See *MULTIPLY Function*.
- See *DIVIDE Function*.
- See *MOD Function*.
- See *NEGATE Function*.
- See *LCM Function*.

Source:

ValueA	ValueB
8	2
10	4
15	10
5	6

Transform:

Execute the following transforms:

```
derive type:single value:ADD(ValueA, ValueB) as:'add'
```

```
derive type:single value:SUBTRACT(ValueA, ValueB) as:'subtract'
```

```
derive type:single value:MULTIPLY(ValueA, ValueB) as:'multiply'
```

```
derive type:single value:DIVIDE(ValueA, ValueB) as:'divide'
```

```
derive type:single value:MOD(ValueA, ValueB) as:'mod'
```

```
derive type:single value:NEGATE(ValueA) as:'negativeA'
```

```
derive type:single value:LCM(ValueA, ValueB) as:'lcm'
```

Results:

With a bit of cleanup, your dataset results might look like the following:

ValueA	ValueB	lcm	negativeA	mod	divide	multiply	subtract	add
8	2	8	-8	0	4	16	6	10
10	4	20	-10	2	2.5	40	6	14
15	10	30	-15	5	1.5	150	5	25
5	6	30	-5	5	0.833333333	30	-1	11