

Add Two Columns

Contents:

- *Check Data Types*
 - *Check Values*
 - *Syntax of Math Functions*
 - *Add One Column into Another*
 - *Add Selective Values from One Column into Another*
 - *Add Two Columns into a New Third Column*
 - *Working with More than Two Columns*
 - *Concatenating Columns*
 - *Summing Rows*
-

This section provides an overview of how to perform mathematical operations between columns.

Check Data Types

Before you begin, you should verify that the data types of the two columns match. Check the icon in the upper left of each column to verify that they match.

To change the data type, you can:

- Click the data type icon
- Select **Edit data type** from the column menu. See *Column Menus*.

Check Values

After setting data types, you should address any missing or mismatched values in the column. For example, if you change a column's data type from Decimal to Integer, values that contain decimal points may be reported as mismatched values. Use the `ROUND` function to round them to the nearest integer.

```
set col:myColumn value:ROUND(myColumn)
```

See *ROUND Function*.

Tip: You can use the `FLOOR` or `CEILING` functions to force rounding down or up to the nearest integer.

See *FLOOR Function*.

See *CEILING Function*.

Syntax of Math Functions

You can express mathematical operations using numeric operators or function references. The following two examples perform the same operation, creating a third column that sums the first two.

Numeric Operators:

```
derive type:single value:(colA + colB + colC) as:'colD'
```

Math Functions:

```
derive type:single value:ADD(colA,colB) as:'colD'
```

NOTE: Expressions containing numeric operators can contain more than two column references or values, as well as nested expressions. Math functions support two references only.

For more information, see *Numeric Operators*.

For more information, see *Math Functions*.

Add One Column into Another

To perform math operations, you can use the `set` transform to update values in a column based on a math operation. The following transform multiplies the column by 10 and adds the value of `colB`:

```
set col:colA value:((colA * 10) + colB)
```

All values in `colA` are modified based on this operation. See *Set Transform*.

Add Selective Values from One Column into Another

You can use the `set` transform to perform math operations based on a condition you define. In the following step, the `Cost` column is replaced reduced by 10% if the `Qty` column is more than 100. The expression is rounded down to the nearest integer, so that the type of the column (Integer) is not changed:

```
set col:Cost value:IF(Qty > 100, ROUND(Cost * 0.9), Cost)
```

For rows in which `Qty` is less than 100, the value of `Cost` is written back to the column (no change).

Add Two Columns into a New Third Column

To create a new column in which a math operation is performed on two other columns, use the `derive` transform. The following multiplies `Qty` and `UnitPrice` to yield `Cost`:

```
derive type:single value:MULTIPLY(Qty,UnitPrice) as:'Cost'
```

See *Derive Transform*.

Working with More than Two Columns

If you need to work with more than two columns, numeric operators allow you to reference any number of columns and static values in a single expression.

However, you should be careful to avoid making expressions that are too complex, as they can be difficult to parse and debug.

Tip: When performing complex mathematic operations, you may want to create a new column to contain the innermost computations of your expression. Then, you can reference this column in the subsequent step, which generates the full expression. In this manner, you can build complex equations in a way that is easier to understand for other users of the recipe. The final step is to delete the generated column.

Concatenating Columns

If you are concatenating string-based content between multiple columns, use the `merge` transform. The following creates a third column with a dash between the values of the two source columns:

```
merge col: ColA, ColB with: '-' as: 'ColC'
```

Tip: This method can be used for columns of virtually any type. Change the data type of each column to String and then perform the merge operation.

Array column types can be concatenated with the `ARRAYCONCAT` function. See *ARRAYCONCAT Function*.

See *Merge Transform*.

Summing Rows

You can use aggregate functions to perform mathematic operations on sets of rows. Aggregated rows are collapsed and grouped based on the functions that you apply to them. See *Aggregate Functions*.