EXAMPLE - Change data type transformation

This example illustrates how to clean up data by changing its data type to String, manipulating it using String functions, and then retyping the data to its proper data type.

Functions:

Item	Description
IF Function	The `IF` function allows you to build if/then/else conditional logic within your transforms.
LEN Function	Returns the number of characters in a specified string. String value can be a column reference or string literal.
MERGE Function	Merges two or more columns of String type to generate output of String type. Optionally, you can insert a delimiter between the merged values.

Source:

The following example contains customer ID and Zip code information in two columns. When this data is loaded into the Transformer page, it is initially interpreted as numeric, since it contains all numerals.

The four-digit `ZipCode` values should have five digits, with a `0` in front.

CustId	ZipCode
4020123	1234
2012121	94105
3212012	94101
1301212	2020

Transformation:

CustId column: This column needs to be retyped as String values. You can set the column data type to String through the column drop-down, which is rendered as the following transformation:

Transformation Name	`Change column data type`
Parameter: Columns	`CustId`
Parameter: New type	`String`

While the column is now of String type, future transformations might cause it to be re-inferred as Integer values. To protect against this possibility, you might want to add a marker at the front of the string. This marker should be removed prior to execution.

The basic method is to create a new column containing the customer ID marker (`C`) and then merge this column and the existing `CustId` column together. It's useful to add such an indicator to the front in case the customer identifier is a numeric value that could be confused with other numeric values. Also, this merge step forces the value to be interpreted as a String value, which is more appropriate for an identifier.

Transformation Name	`Merge columns`
Parameter: Columns	`'C',CustId`

You can now delete the `CustId` columns and rename the new column as `CustId`.

ZipCode column: This column needs to be converted to valid Zip Code values. For ease of use, this column should be of type String:

Transformation Name	`Change column data type`
Parameter: Columns	`ZipCode`
Parameter: New type	`Zipcode`

The transformation below changes the value in the `ZipCode` column if the length of the value is four in any row. The new value is the original value prepended with the numeral `0`:

Transformation Name	`Edit column with formula`
Parameter: Columns	`ZipCode`
Parameter: Formula	`if(len($col) == 4, merge(['0',$col]), $col)`

This column might now be re-typed as Zipcode type.

Results:

CustId	ZipCode
C4020123	01234
C2012121	94105
C3212012	94101
C1301212	02020

Remember to remove the `C` marker from the `CustId` column. Select the `C` value in the `CustId` column and choose the `replace` transform. You might need to re-type the cleaned data as String data.