Remove Data

Contents:

- *Considerations when removing data*
- *Drop columns*
- *Delete rows*
  - *Delete rows based on selections*
  - *Filter rows based on matching conditions*
  - *Delete rows based on multiple blank cells*
- *Remove values*
  - *Using regular expressions*

Through simple selections, you can identify columns to remove, values on which to base row deletion, or strings to remove from your dataset. As needed, these transformations can be modified for more sophisticated removal transformations.

Considerations when removing data

Please keep in mind:

- When data is removed from your dataset, no actual deletion is performed.
  - Trifacta® Self-Managed Enterprise Edition does not modify source data. All recipe executions generate new sets of data based on the transformations you define, which are applied to a generated version of the source data.
  - Transformation steps are previewed and can be undone on sampled data in the Transformer page, so you should feel free to experiment with data removal.
- In large volume datasets, be careful applying patterns or regular expressions to your data. You should limit your application of these pattern-based changes to the minimum range of columns, rows, or strings required to complete the task.

Drop columns

To drop a column from your dataset, click the column drop-down and select **Drop**. The data is no longer available in the data grid or subsequent recipe steps.

Tip: To simply remove columns from display, use the **Hide** command. The hidden column still appears in the output.

To drop multiple columns, you can specify comma-separated column names in your Delete Columns transformation:

Transformation Name	`Delete columns`
Parameter: Columns	`ColA,ColC,ColE`
Parameter: Action	`Delete selected columns`

To drop a range of columns, use the tilde (~) character between the start and end column names:

Transformation Name	`Delete columns`
Parameter: Columns	`ColA~ColE`

Parameter: Action	`Delete selected columns`

For more information, see *Remove Data*.

Tip: You can also drop multiple columns through the Column Browser. See *Column Browser Panel*.

Delete rows

You can delete rows in your dataset based on conditional patterns that you specify. The easiest method is to select a string in the appropriate column and then choose the Delete suggestion card.

Delete rows based on selections

Steps:

In the following example, each row contains an entry for a different business, and you want to remove all of the business entries from the city of Tempe.

1. In this case, you could use the column histogram to select the value `Tempe` in the `city` column, or you can use the Filters panel to filter for rows containing the value `Tempe`.
2. Then, select the Delete suggestion card.

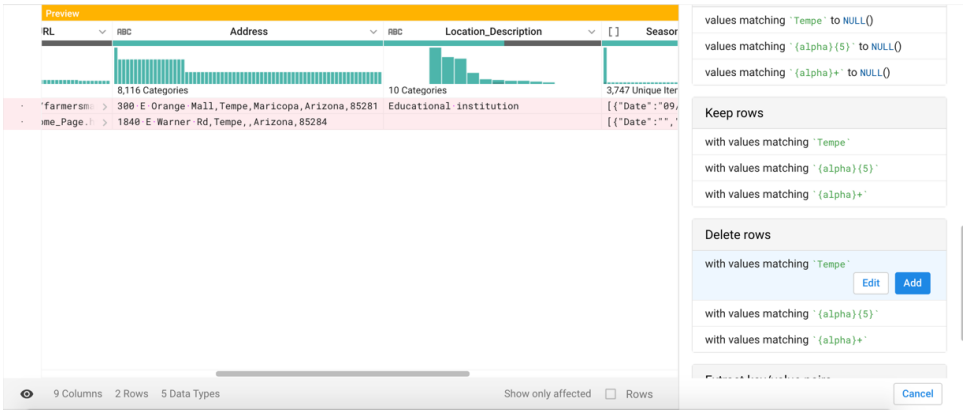

Figure: Select Tempe in the City column to remove all entries for that city

3. After selecting `Delete`, the application evaluates your selected value and attempt your intention with the selection. Is it a string literal or a pattern? If it's a pattern, what does the pattern represent? You may select one of the variants in the Delete card to find the right match.

   NOTE: Be sure to scroll up and down in the data grid to review the values that are affected. In some cases, your selection may turn into a pattern, which could apply to more than just the desired values. In the previous example, selecting `Tempe` may yield a matching pattern of `{alpha}{5}`, which would match any five-letter city name, including `Tempe`. Select other variants in the Delete card to change the matching pattern. Click **Modify** to review the matching string.

4. After defining and modifying your Filter Rows transformation, you can use the preview to see the rows that will be removed, prior to adding the transformation to your recipe.

Tip: You can also use the Filter Rows to retain rows based on a specified condition, effectively deleting the rows that do not match. See *Filter Data*.

Filter rows based on matching conditions

You can delete or keep rows in your dataset based on one or more matching conditions you define.

1. In the Search panel, enter `filter`.
2. Select the type of conditional. You can filter based on:
    a. Type: missing or mismatched values.
    b. Matches: literal or pattern matches that are exact matches, partial matches, or matches with the beginning or ending of column values.
    c. Ranges: Less than (or equal to), greater than (or equal to), or combinations.
    d. Custom formula: Specify an expression that evaluates to `true` or `false`. If `true`, then the data is filtered.
3. Specify the other parameters, including whether to drop or keep the matching rows.

For more information, see *Filter Data*.

Delete rows based on multiple blank cells

If you have rows in your dataset that contain no data, you can use the following two steps to remove them. Assuming that you know the starting (`col1`) and ending (`colN`) column names of your dataset, try the following:

NOTE: If at a later time, you reorder or remove the starting or ending columns in a step before this one, these steps are broken.

Transformation Name	`New formula`
Parameter: Formula type	`Single row formula`
Parameter: Formula	`MERGE([column1~columnN])`
Parameter: New column name	`'all_blank_vals'`

Transformation Name	`Delete rows when value is missing`
Parameter: Column	`all_blank_vals`
Parameter: Action	`Delete selected columns`

The above merges all values into a single value in the `all_blank_vals` column. The second step removes the row if the value in the merged column is blank.

Remember to delete the `all_blank_vals` column after you are done.

For more information, see *Filter Data*.

Remove values

To delete values from a column, select the values in the data grid. In the suggestion cards, select the `Replace` card. In the following example, the `city` column is removed of all values matching `Tempe`:

Transformation Name	`Replace text or patterns`
Parameter: Column	`city`
Parameter: Find	`'Tempe'`

Parameter: Replace with	`''`
Parameter: Match all occurrences	`true`

The Replace transformation applies only to string values. The rest of a matching row is unaffected.

The above transformation matches all values in the column, even partial values, the match string is removed from the column value. For example, an entry `Tempest` would be turned into `st` if the above transformation was added.

To ensure that only full-column value matches are applied, you can add Trifacta patterns to indicate the start and end of the column value as in the following:

Transformation Name	`Replace text or patterns`
Parameter: Column	`city`
Parameter: Find	`` `{start}Tempe{end}` ``
Parameter: Replace with	`''`
Parameter: Match all occurrences	`true`

In the above case, only values of `Tempe` that are the entire column value are matched. For more information on this pattern-based matching, see *Text Matching*.

Using regular expressions

For more sophisticated matching, you can apply regular expressions to your `replace` command. In the following example, all integers from 0-99 are matched in the `qty` column. Because there is no replacement value, they are deleted.

Regular expressions are very powerful pattern matching tools. You should be careful in your use of them. See *Text Matching*.

Character	Definition
^	Beginning of string. Required to prevent matching on the last digit of any numeric value.
$	End of string. Required to prevent a 2-digit match on three-digit numbers.
\d	A single digit
|	Logical or. In this case, it is used to define separate regexes for 1- and 2-digit values.