

# Manage Null Values

## Contents:

- *Important notes on null values*
- *Locate null values*
- *High percentage of nulls*
- *Fix null values*
- *Null values in transformations*
- *Write null values*

In general terms, a null value is a definition that points to nothing. A container for a value, such as a row-column combination or a variable, exists, but the container points to no actual value.

## Important notes on null values

**NOTE:** In the platform, null values are a subset of the category identifying missing values. For technical reasons, however, Trifacta® displays null values as missing values and visually treats them as the same. Internally, they are understood to be different values.

## Implications:

- Null values are visually represented as missing values.
  - In the data quality bar, null and missing values are represented in the dark bar (missing values).
- Computationally, they are different types of values.
  - Most functions applied to null and missing values return the same results.
    - For example, the `ISMISSING` function returns `true` for null and missing values.
    - However, the `ISNULL` function returns `true` for a null value and `false` for a missing value. See below.
  - If you use a function to generate null values, they are displayed as missing values, although they are recorded as nulls.
    - For example, the following transform generates a column of null values, which are represented as missing values in the data quality bar.

<b>Transformation Name</b>	New formula
<b>Parameter: Formula</b>	<code>NULL( )</code>
<b>Parameter: New column name</b>	<code>nulls</code>

- When a set of results is generated, both null and missing values are written as missing values, unless the output format has a specific schema associated with it.

**NOTE:** When a recipe containing a user-defined function is applied to text data, any null characters cause records to be truncated by the running environment during Photon job execution. In these cases, please execute the job in the Spark running environment.

## Locate null values

Null values are displayed with missing values in the Missing values category of the data quality bar (in black).

You can use the following transform to distinguish between null and missing values. This transform generates a new column of values, which are set to `true` if the value in `isActive` is a null value:

**Tip:** You can use this transform and a subsequent sorting step on the generated column to filter for null values.

<b>Transformation Name</b>	New formula
<b>Parameter: Formula</b>	<code>ISNULL(isActive)</code>
<b>Parameter: New column name</b>	<code>nulls2</code>

## High percentage of nulls

On import, if a column has a high enough percentage of null values, the platform may retype the column as a `String` column, which may yield mismatched values in addition to the missing values that were imported from null values.

## Fix null values

See *Find Missing Data*.

## Null values in transformations

### Functions:

- Applying a null value as an input to a scalar function returns a null value, propagating the null value.
- In aggregate or window functions, null values are ignored, as a single null value could corrupt an entire column of calculations.

### Transforms:

- In a join, a null value in one dataset never matches with a null value in another dataset. Rows with null values in join key columns are never included in the output. See *Join Types*.

## Write null values

If needed, you can write a null value to a set of data. In the following example, all missing values in a column are replaced by nulls, using the `NULL` function.

**NOTE:** The `NULL` function is typically used to pass null values into functions that have been designed to specifically address them.

The following example tests all columns in the range between `column1` and `column255` for whether a missing value is detected. If so, a null value is written. Otherwise, the column value is written back to the column:

<b>Transformation Name</b>	Edit column with formula
<b>Parameter: Columns</b>	<code>column1~column255</code>
<b>Parameter: Formula</b>	<code>IF(ISMISSING([\$col]), null(), \$col)</code>

The above transform writes null values, but these values are converted to missing values on export.