Tableau Prep Help

Last Updated 9/26/2018

What's New in Tableau Prep

Browse summaries of new features for currently supported versions.

What's new in version 2018.2.3

Connect to Data

- **Connect to data stored in MongoDB Business Intelligence (BI)** below

Examine and Filter Your Data

- **Specify a data role for your field values** on the next page

Join or Union Data

- **Fix mismatched fields directly in the join clause** on page 5

Connect to Data

Connect to data stored in MongoDB Business Intelligence (BI)

If you store your data in Mongo DB Business Intelligence, you can now connect to your data and clean it with Tableau Prep.

For more information about how to connect to your data using Mongo DB Business Intelligence, see MongoDB BI Connector in the Tableau Desktop help.

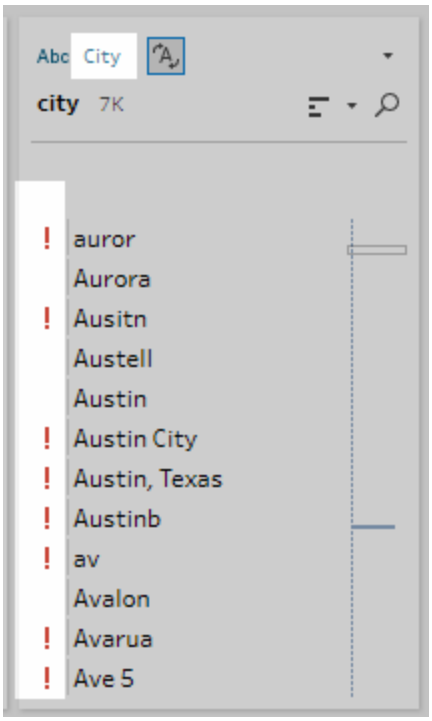
Note: Data connectors are not backward compatible. Flows that include these connectors may open in a prior version of Tableau Prep, but will have errors or can't run unless the data connections are removed.

Examine and Filter Your Data

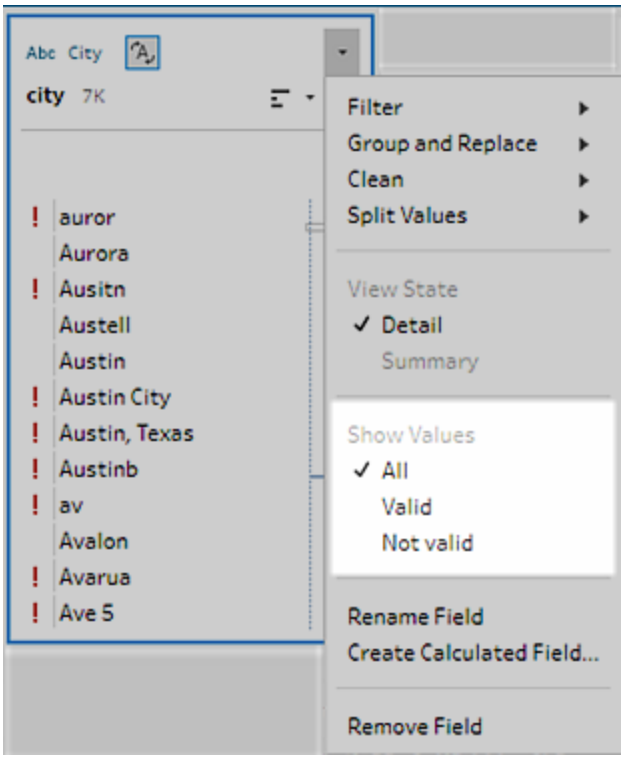
Specify a data role for your field values

You can now assign a data role to your field values and use Tableau Prep to help you find inaccuracies or outliers in your data set.

Data roles tell Tableau Prep what the field values mean or represent, for example email addresses or a geographic role such as city or zip code. When a data role is assigned to a field, Tableau Prep examines the field values and flags the values that don't match so that you can take a closer look.



To view only the values that are valid or not valid, use the new filter option on the drop-down menu:



In this release we support the following data roles:

- Email
- URL
- Geographic roles (Based on current geographic data and is the same data used by Tableau Desktop)
  - Airport
  - Area code (U.S.)
  - CBSA/MSA
  - City
  - Congressional District (U.S.)
  - Country/Region
  - County
  - NUTS Europe
  - State/Province
  - Zip code/Postal code

For more information see **Assign data roles to your data** on page 110.

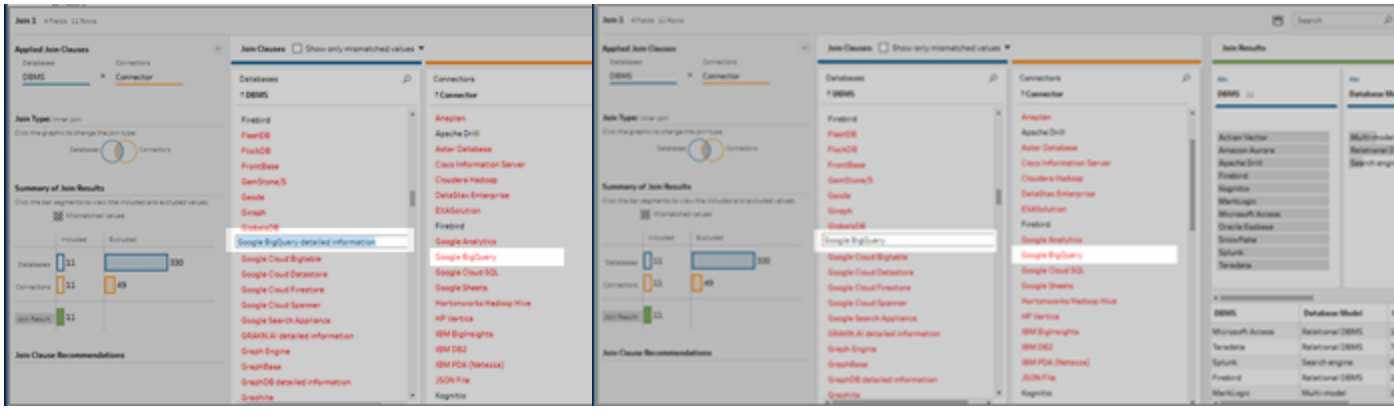
Join or Union Data

Fix mismatched fields directly in the join clause

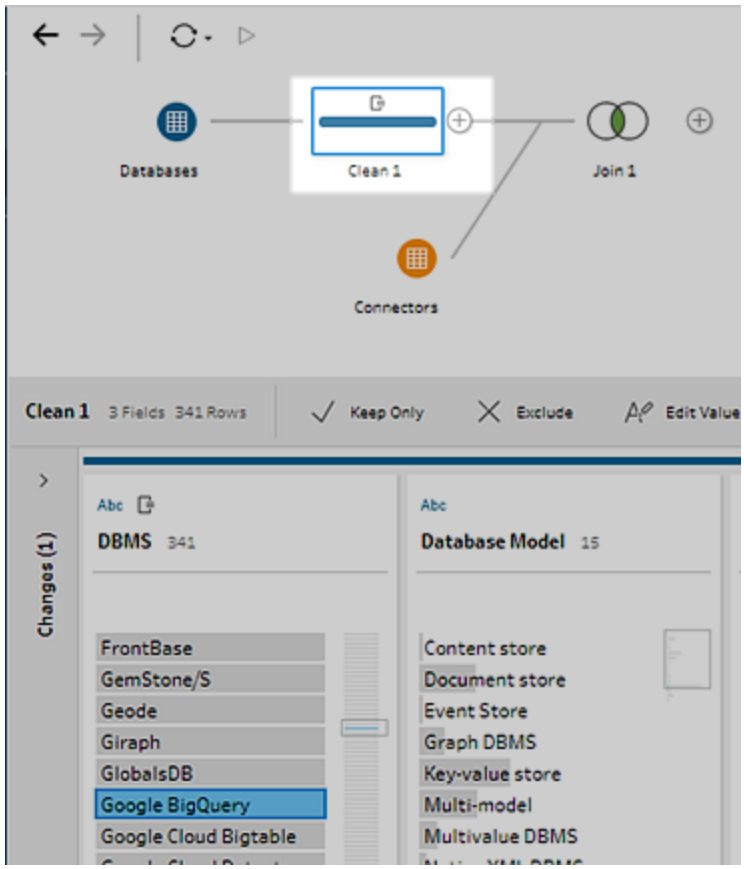
When you join two tables of data you will often have field values that are the same but are mismatched due to data entry errors. Tableau Prep helps identify mismatched fields in your join clauses by turning the mismatched field values red. But wouldn't it be great if you could just fix those field values right in your join clause? Well now you can.

Now when you join two data sets you can edit the field values that you join on, right in the join clause to fix them. Simply double-click in a field value to edit it.

Select field to change **Edit in-line**



The cleaning action is automatically pushed back to the previous cleaning step in the flow for the appropriate join data set.



No cleaning step before the join step? No problem. Tableau Prep automatically adds one for you to capture the cleaning operation from the join.

For information about how to clean field values directly in a join clause, see **Join or Union Data** on page 124.

What's new in version 2018.2.2

Install and Deploy Tableau Prep

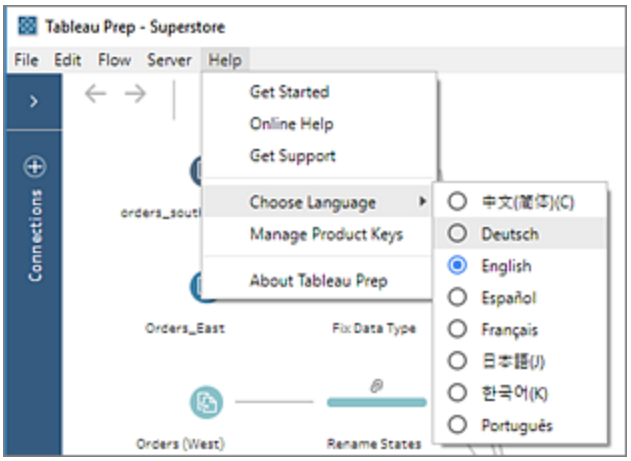
- **Set your display language** on the next page

Connect to Data

Install and Deploy Tableau Prep

Set your display language

When you start up Tableau Prep, it now detects the locale set on your computer and displays the user interface, dates, and number formats in the appropriate language. If you want to change the display language for the user interface you can select from the supported languages from the top menu under **Help** > **Choose Language**.



For more information see Set your display language in the Tableau Desktop and Tableau Prep deployment guide.

Connect to Data

Connect to data stored in Microsoft Access

If you use Microsoft Access for data entry or to store your data tables you can now connect to your access files (from version 2007 or higher) through a file browser and clean your data with Tableau Prep.

This connector requires a 64-bit driver.

For more information about how to connect Tableau Prep to your data, see Access in the Tableau Desktop help.

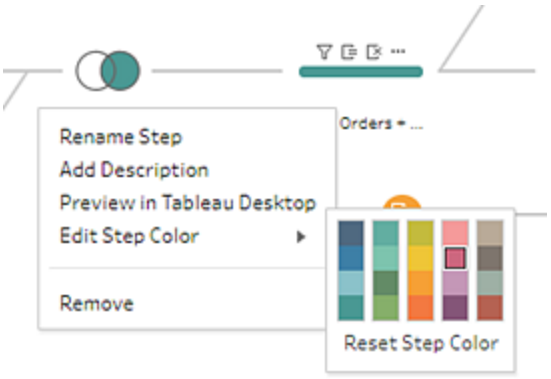
> **Note**: Data connectors are not backward compatible. Flows that include these connectors may open in a prior version of Tableau Prep, but will have errors or can't run unless the data connections are removed.

Clean and Shape Data

Change the color scheme for your flow steps

By default, Tableau Prep assigns each step in your flow a color to help you easily track the changes you make to your data as you build your flow. But you have choices when it comes to this color scheme.

You can now pick from a color palette to change the color scheme for one or more steps. Just select the steps in the Flow pane that you want to change, right-click the selected steps and select **Edit Step Color** from the context menu.

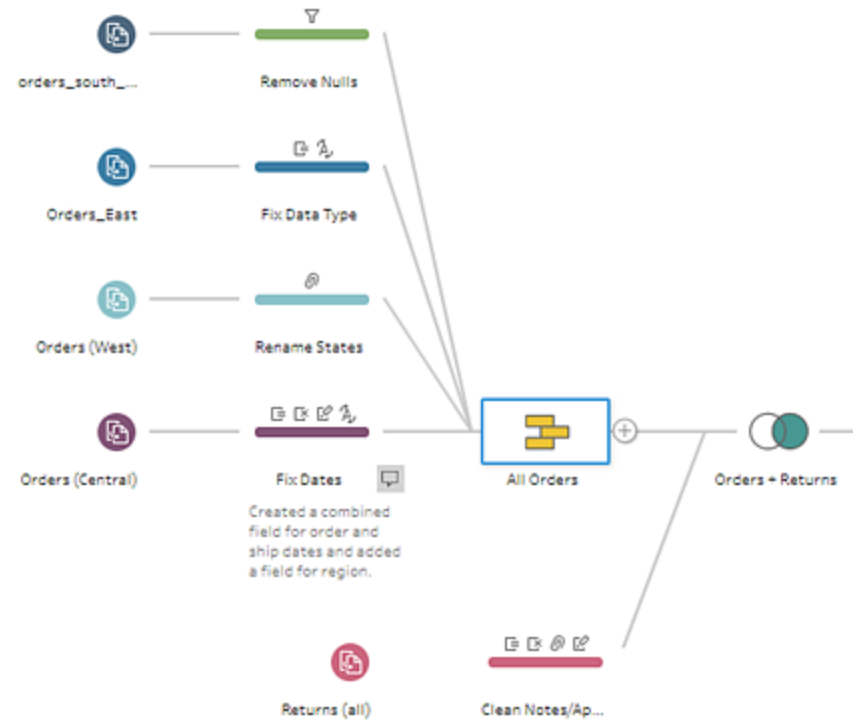


Don't like how it looks? Click **Undo** from the top menu or select **Reset Step Color** from the color palette menu.

For more information, see **Build your flow** on page 87.

Add descriptions to your steps

If you share your flows with others, communicating the changes that you made and why can be cumbersome. To make it easier, you can now add a short description to any individual step in your flow and it displays right in the flow pane.



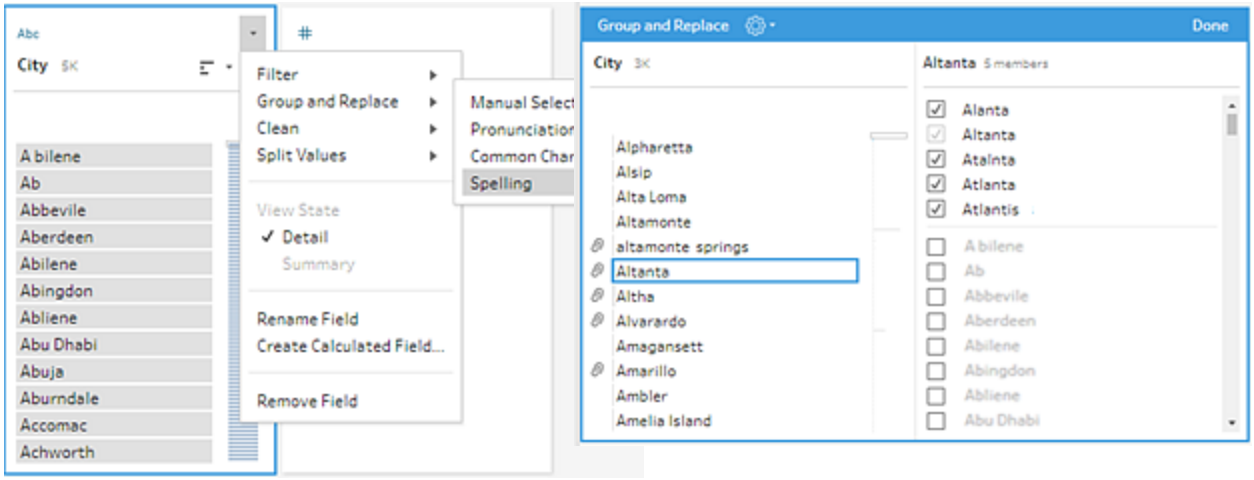
For more information about adding descriptions to flow steps, see **Build your flow** on page 87.

Use fuzzy match to find and fix spelling errors

To help you quickly identify and clean up multiple variations of the same value due to spelling errors, we've added another option to our fuzzy match cleaning feature. Use the new **Spelling** option to find and group text values that differ because of additional or missing letters. This option works in any supported language.

The **Spelling** option uses the Levenshtein distance algorithm to compute an edit distance between two text values and then groups them together when the edit distance is less than a default threshold value.

For more information, see **Cleaning (fixing) variations of the same value** on page 96.



Save and share your work

Run flows from the command line

To keep data fresh in Tableau Prep you run your flow. However, opening Tableau Prep every time you want to run flows can take time. To help streamline this process, you can now run flows from the command line without having to open it in Tableau Prep.

To run a flow from the command line, you'll need:

- The path to the flow (.tfl) file that you want to run.
- A .json file that contains the database credentials for any databases that the flow connects to for its input steps and the credentials for the server where the output is published.

  If your flow connects to or publishes to local files or files that are stored on a network share for inputs or outputs, then this file isn't needed.

  > **Note**: Connecting to or publishing files that are stored on a network share that are password protected isn't supported.

- Administrator permissions on the machine where you are running the flow.

This option is available on both Windows (Task Scheduler is supported) and Mac machines. To use this process, you need an activated version of Tableau Prep and the process must be run on the same machine where Prep is installed.

For information about how to run flows from the command line, see **Refresh output files from the command line** on page 141.

What's new in version 2018.2.1

Install and Deploy Tableau Prep

- **Deactivate Tableau Prep from the command line** on the next page
- **Use virtual desktop support to optimize Tableau Prep Installations** on the next page

Connect to Data

- **Connect to cloud data sources and Hadoop Hive** on page 13

Explore Your Data

Install and Deploy Tableau Prep

Deactivate Tableau Prep from the command line

Like Tableau Desktop, if you no longer need Tableau Prep on your computer you can now deactivate it from the command line using the -return option.

For more information see Deactivate the product key. For more information about other installer properties that are available for Tableau Prep from the command line, see the Installer options and relevant sections in Deploy Tableau Desktop.

Use virtual desktop support to optimize Tableau Prep Installations

Just like Tableau Desktop you can now configure virtual desktop support to optimize your installations of Tableau Prep for non-persistent virtual desktops or for computers that are regularly reimaged. With virtual desktop support, Tableau Prep licenses are automatically

deactivated after a predetermined period of time using a Tableau-hosted “Authorization to Run” (ATR) service, eliminating the need to manually deactivate the product key.

For more information about how to configure this option, see Configure Virtual Desktop Support in the Tableau Desktop and Tableau Prep deployment guide.

Connect to Data

Connect to cloud data sources and Hadoop Hive

We've added support for the following connectors so that you can connect to cloud data and data stored in Hadoop.

- Snowflake.
- Amazon EMR Hadoop Hive
- Cloudera Hadoop (Hive and Impala)
- Hortonworks Hadoop Hive
- MapR Hadoop Hive
- Apache Drill
- SparkSQL

For more information about how to connect Tableau Prep to your data, see the topic for your connector under Supported Connectors in the Tableau Desktop help.

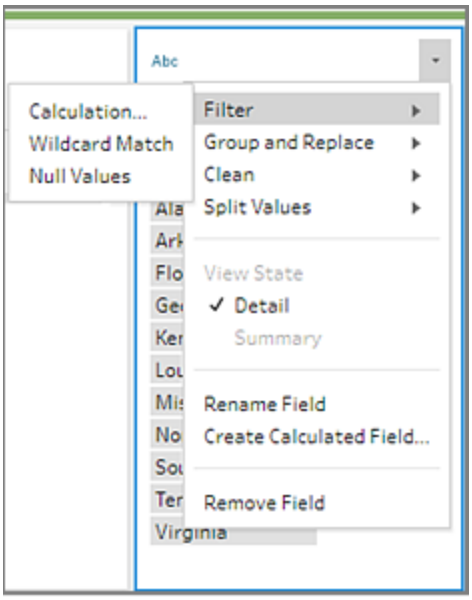
These data connectors are not backward compatible. Flows that include these connectors may open in a prior version of Tableau Prep, but will have errors or can't run unless the data connections are removed.

Explore Your Data

New filter options to keep only the data you want

No more writing complex calculations to keep or exclude Null values or to set up a wildcard match on text values. Instead select one of the new filter options on a field to see the impact of

your changes right away without having to first write a calculation and then revert your change if it doesn't give you the results you want.



For more information, see **Filter values** on page 118.

Clean and Shape Data

Use the ISO-8601 date standard in calculated fields

Creating calculated fields to support European calendars just got easier. Tableau Prep now supports the ISO-8601 international date standard for the following date parts:

- "iso-year"
- "iso-quarter"
- "iso-week"
- "iso-weekday"

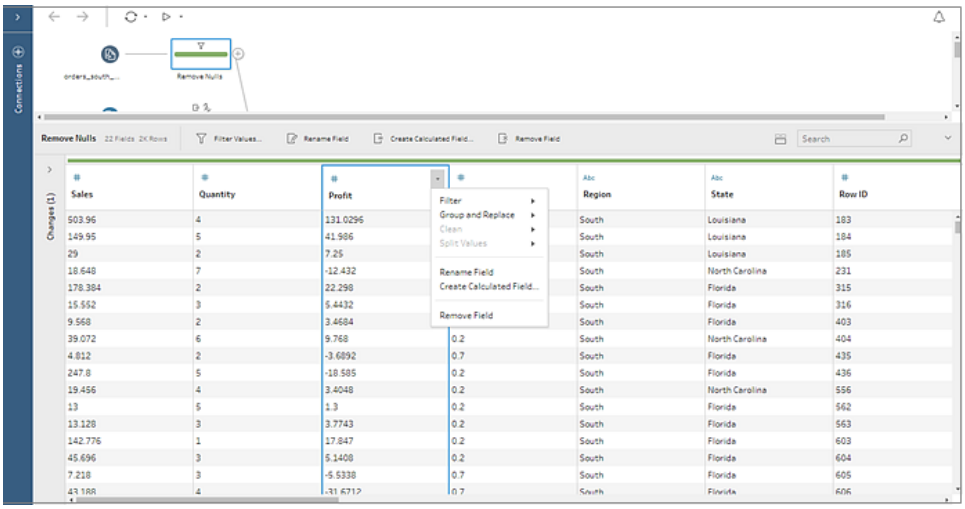
Use these date parts in functions DATEPART, DATETRUNC, DATENAME, DATEDIFF, and DATEADD.

For example `Week Number = STR(DATEPART('iso-year', [Week Date])) + "-" + STR(DATEPART('iso-week', [Week Date]))`

For more information about how to work with date functions in calculated fields, see Date Functions in the Tableau Desktop help.

Apply cleaning operations in the data grid

In prior versions, the data grid showed you a preview of your data, but had few cleaning options available. Now you can act on your data anywhere. If you want to work with the detailed values in the data grid, collapse the Profile pane and perform the same cleaning operations that are available in the Profile pane in the data grid.

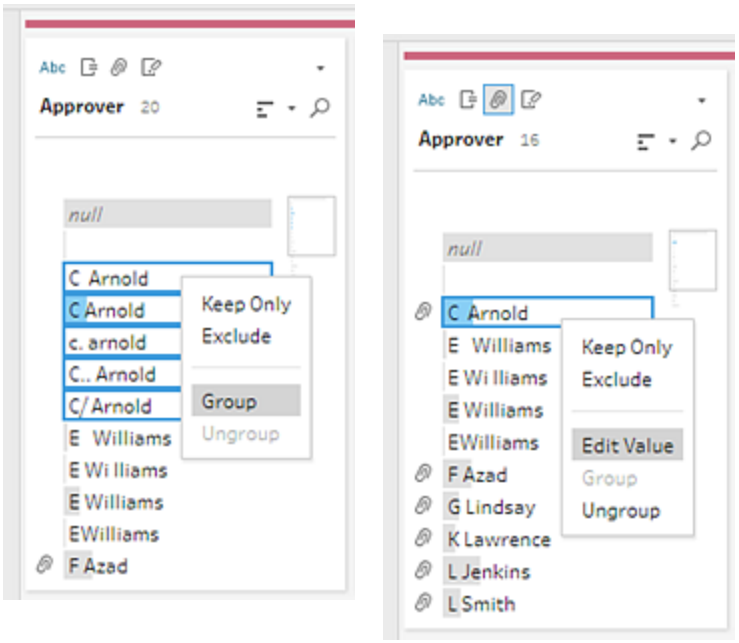


For more information, see **Clean and Shape Data** on page 87.

Use multi-select to group values in the Profile pane

To quickly group a set of values for a field, you can now multi-select the values in the Profile card, then right-click to open the menu and select **Group**. The values are grouped under the field value that you select when you right-click to open the menu. A paperclip icon shows next to the grouped value.

Right-click the grouped values to open the menu again to ungroup or edit the values.



For more information about grouping values, see **Cleaning (fixing) variations of the same value** on page 96

Other enhancements

We've also made the following enhancements to improve usability:

- New icons show on the menu when adding steps to your flow to provide visual cues and help you learn the visual language of Tableau Prep.

- New animations in the Profile pane help catch your eye to better see the impact of your changes.

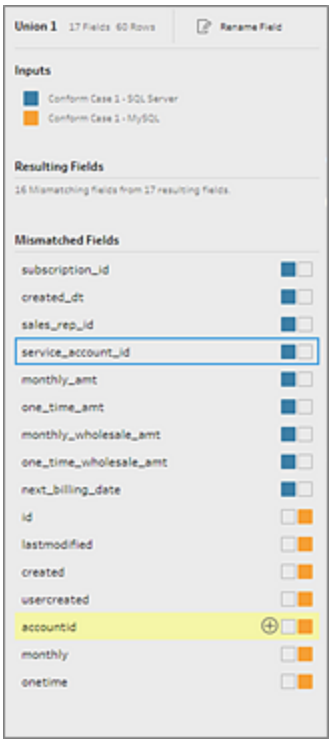
Join or Union Data

Use union recommendations to clean mismatched fields

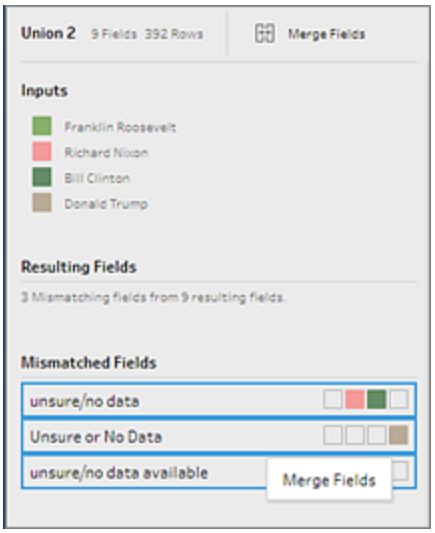
Fixing mismatched fields after unioning two or more files just got easier. A new **Mismatched Fields** section in the **Union summary** pane shows a list of fields that don't match and the data source where they originated. Merge your mismatched fields directly in this section using one of the following options:

- Select a field in the list. If Tableau Prep identifies a field with similar characteristics, based on similar data types and field names, it highlights the field in yellow, suggesting a match.

  Click the plus ⊕ button that appears on hover on the suggested matched field to merge the two fields.



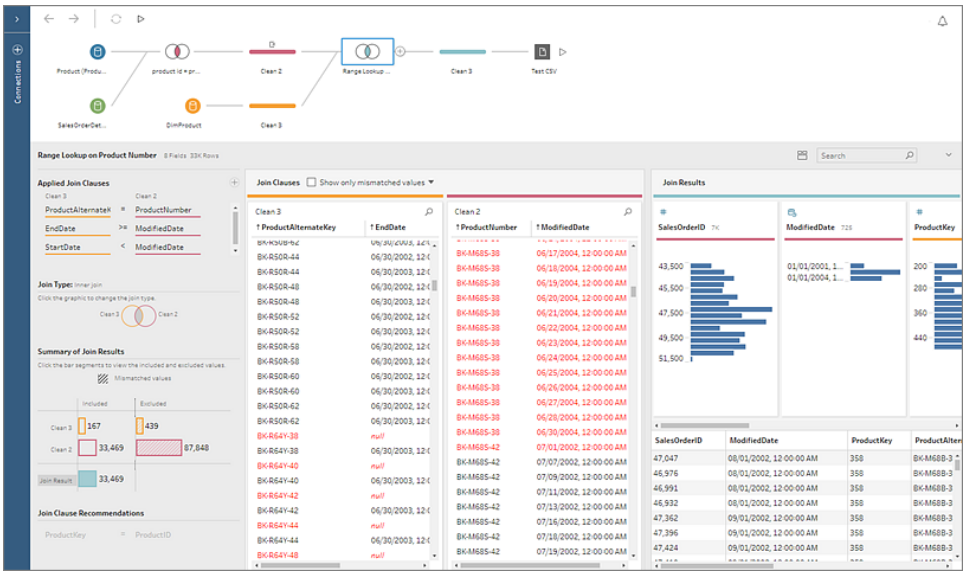
- Select two or more fields in the list, right-click on one of them and select **Merge Fields**.



For more information about unioning data and resolving mismatched fields, see **Union your data** on page 129.

Identify mismatched fields for all join types

Easily identify fields that don't match no matter how you join tables. In prior versions the **Join Clauses** tables showed field values that were excluded only when one field equaled another. But now you can see this data using any operator when matching join clauses, such as "End Date >= Modified Date".



For more information, see **Join your data** on page 124.

What's new in version 2018.1.2

Install and Deploy Tableau Prep

- **Activate and register Tableau Prep from the command line (Windows)** on the next page

Connect to Data

- **Connect to data stored in statistical files or on Presto** on the next page
- **Union sub-tables found by Data Interpreter in the Input step** on the next page
- **Better feedback when loading tables** on page 21

Explore Your Data

- **Reorder fields in the Profile pane and the Data grid** on page 22

Clean and Shape Data

- **Use drag-select to remove multiple steps in your flow** on page 22
- **Pivot multiple groups of fields in a single action** on page 23
- **Improved field naming when merging fields** on page 24
- **Other enhancements** on page 24

Install and Deploy Tableau Prep

Activate and register Tableau Prep from the command line (Windows)

Like Tableau Desktop, you can now activate and register Tableau Prep from the command line by including a command line with the following properties:

- ACTIVATE_KEY="<key>". The installer runs -activate to apply the license key.
- REGISTER="1". During the installation process, the installer will run the -register process and add the registration information.

For more information about these installer properties and how to activate and register Tableau Prep from the command line, see the Installer options and relevant sections in Deploy Tableau Desktop.

Connect to Data

Connect to data stored in statistical files or on Presto

We've added two new connectors to help you connect to data from more locations.

- Statistical files. Connect to to SAS (*.sas7bdat), SPSS (*.sav), and R (*.rdata) data files.
- Presto. For more information about how to configure your connection to Presto, see Presto in the Tableau Desktop help.

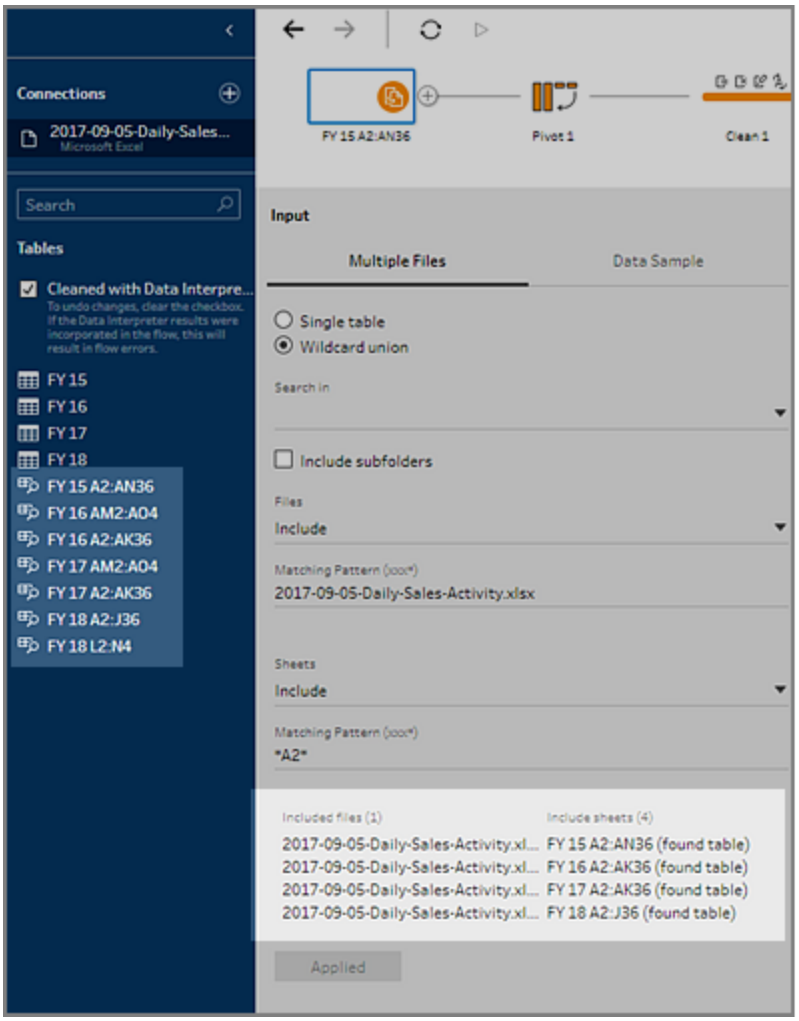
These data connectors are not backward compatible. Flows that include these connectors will open in a prior version of Tableau Prep, but will have errors or can't run unless the data connections are removed.

Union sub-tables found by Data Interpreter in the Input step

Using Data Interpreter to clean your Microsoft Excel data and now you want to union the resulting sub-tables? You can now use wildcard union to union all of the found sub-tables in the Input step.

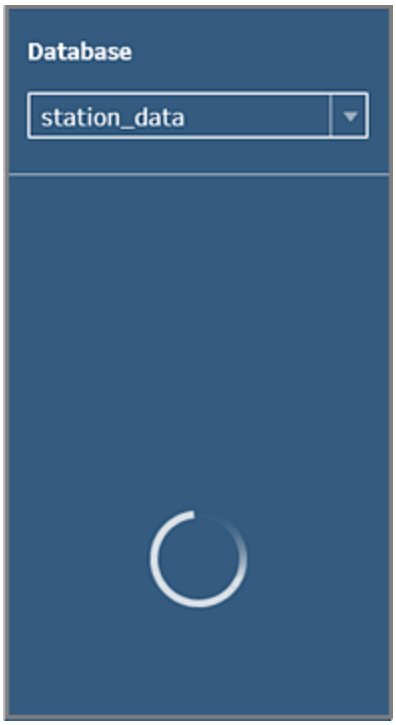
Simply drag one of your sub-tables to the **Flow** pane, and then use the wildcard search criteria to select the remaining sub-tables to union the data and include all the sub-table data in the Input step.

For more information see **Union files in the Input step** on page 77.



Better feedback when loading tables

When you connect to a database for the first time, it can sometimes seem like nothing happened. Now a new indicator tells you that the data is still loading.



Explore Your Data

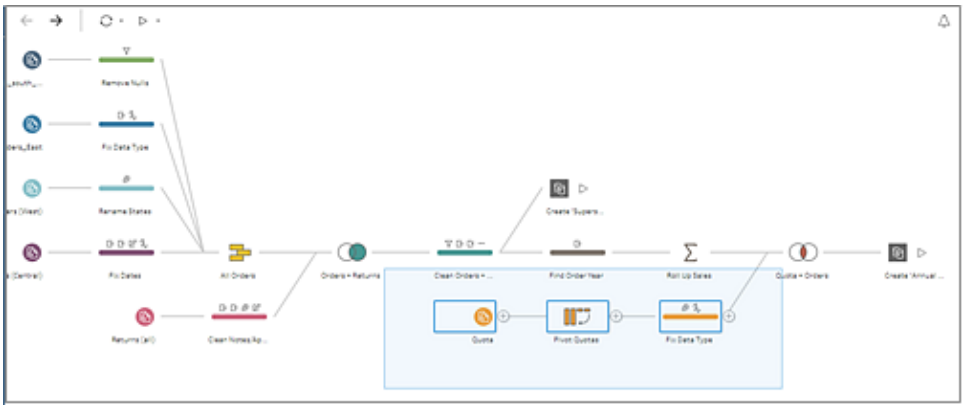
Reorder fields in the Profile pane and the Data grid

You can now drag and drop to reorder your fields in both the **Profile** pane and the **Data** grid and the two panes will stay in sync. We also maintain the field order even when you rename a field. Fields are no longer reordered automatically.

Clean and Shape Data

Use drag-select to remove multiple steps in your flow

Removing whole sections of your flow just got easier. You can now click in the **Flow** pane and use your mouse to drag and select the section of the flow that you want to remove. Then right-click to remove all of the selected steps at once. For more information see **Build your flow** on page 87.



Pivot multiple groups of fields in a single action

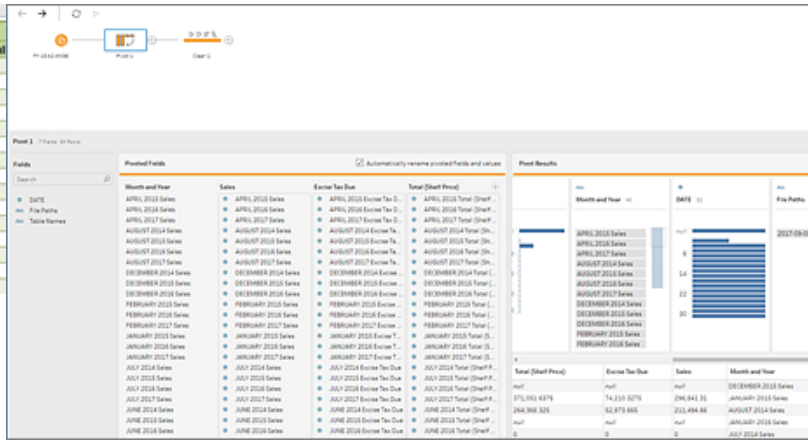
The pivot option in Tableau Prep has been expanded to better support more complex spreadsheets or text files. You can now perform either a single pivot or select groups of fields to pivot.

For example to pivot a spreadsheet to see sales, tax, and totals by month and year you can pivot each group of fields to get the results you want. For more information, see **Pivot your data** on page 91.

Original Spreadsheet

DATE	DECEMBER 2014			JANUARY 2015		
	Sales (Shelf Price)	Excise Tax Due	Total (Shelf Price	Sales (Shelf Price	Excise Tax Due	Total
1	$ 448,111	$ 112,028	$ 560,139	$ 296,841	$ 74,210	$
2	$ 425,472	$ 106,368	$ 531,840	$ 754,061	$ 188,515	$
3	$ 435,525	$ 108,881	$ 544,406	$ 482,497	$ 120,624	$
4	$ 634,765	$ 158,691	$ 793,456	$ 332,228	$ 83,057	$
5	$ 695,425	$ 173,856	$ 869,282	$ 601,529	$ 150,382	$
6	$ 436,720	$ 109,180	$ 545,899	$ 527,374	$ 131,843	$
7	$ 238,481	$ 59,620	$ 298,101	$ 560,102	$ 140,026	$
8	$ 421,422	$ 105,356	$ 526,778	$ 539,974	$ 134,993	$
9	$ 543,816	$ 135,954	$ 679,770	$ 683,408	$ 170,852	$
10	$ 616,271	$ 154,068	$ 770,339	$ 442,352	$ 110,588	$
11	$ 756,542	$ 189,135	$ 945,677	$ 288,605	$ 72,151	$
12	$ 726,270	$ 181,567	$ 907,837	$ 674,121	$ 168,530	$
13	$ 477,208	$ 119,302	$ 596,510	$ 526,451	$ 131,613	$
14	$ 245,898	$ 61,475	$ 307,373	$ 573,842	$ 143,461	$
15	$ 456,254	$ 114,064	$ 570,318	$ 658,952	$ 164,738	$

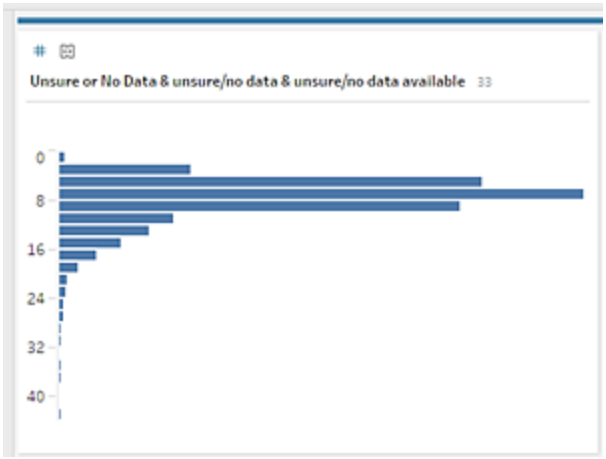
After pivoting on multiple groups of fields



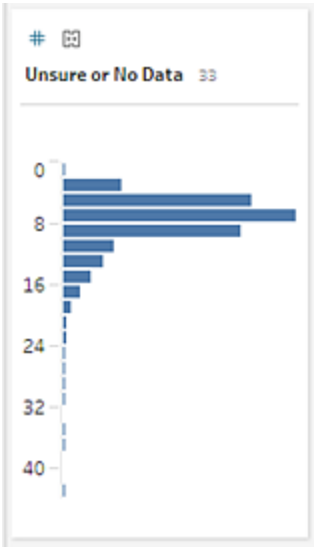
Improved field naming when merging fields

When you merge fields, the field names are no longer concatenated. Instead the field name of the target field persists. For more information, see **Merge fields** on page 95.

Concatenated field naming



Simplified field naming



Other enhancements

We've also made the following enhancements to improve usability and performance:

- A new edit icon shows on hover on entries in the **Changes** pane to help you quickly see which items are editable.

- Renaming a step will no longer automatically run the flow so your authoring experience is seamless.
- The **Connections** pane automatically opens when you connect to a Microsoft Excel file with multiple sheets to help you quickly get to your data.

Get Started with Tableau Prep

This tutorial introduces you to the common operations that are available in Tableau Prep. Using the sample data sets that come with Tableau Prep, you will walk through creating a flow for Sample Superstore.

Watch for tips along the way to gain insights into how Tableau Prep helps you clean and shape your data for analysis.

To install Tableau Prep before continuing with this tutorial, see Install Tableau Prep in the Tableau Desktop and Tableau Prep Deployment guide. Otherwise you can download the free trial.

Note: To complete the tasks in this tutorial, you need to install Tableau Prep, and you need the sample Superstore data files located here:

- (Windows) `C:\Program Files\Tableau\Tableau Prep <version>\help\Samples\en_US\Superstore Files`
- (Mac) `/Applications/Tableau Prep <version>.app/Contents/help/Samples/en_US/Superstore Files`

In this article

Here's the story...

You work at the headquarters for a large retail chain. Your boss wants to analyze product sales and profits over the last four years for the company. You suggest that he use Tableau Desktop to do that. Your boss thinks that's a great idea and wants you to get right on that.

As you start gathering all the data you'll need, you notice that the data has been collected and tracked differently for each region. You also notice a lot of creative data entry in the different files, and that one region even has a separate file for each year!

Before you can start analyzing the data in Tableau, you'll have to do some serious data cleaning first, and it's going to be a long night.

As you rummage for restaurant menus to order some dinner, you remember that Tableau just introduced a new product called Tableau Prep that might help you with your Herculean data cleaning task.

You sign up for a free trial and decide to give it a try.

1. Connect to data

The first thing you see when you open Tableau Prep is a Start page with a **Connections** pane, just like Tableau Desktop.

To get started, the first step is to connect to your data and create an Input step. From there you will start building a workflow or "flow", as it's called in Tableau Prep, and add more steps to take action on your data as you go.

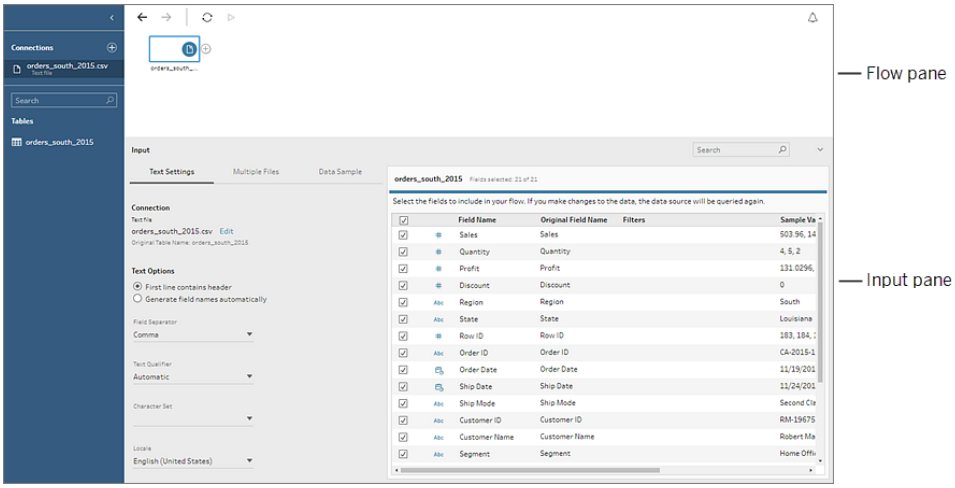
> **Tip**: The Input step is the ingestion point for your data and the starting point for your flow. You can have multiple Input steps and some might include multiple data files. For more information about connecting to data, see **Connect to Data** on page 71.

Your sales data files for the different regions are stored in different formats, and your orders from the South are actually multiple files. You check out the **Connections** pane and see that you have a lot of choices to connect to data. Great!

Since your other regions have one file for all four years worth of data, you decide to tackle the files from the South first.

1. On the **Connections** pane, click the **Add connection** button.
2. The files are .csv files, so select **Text file** in the list of connections.
3. Navigate to the directory for your files, select the first file **orders_south_2015.csv** and click **Open** to add it to your flow. (For file location, see **Wrap up and resources** on page 59.)

   After you connect to your first file, the Tableau Prep workspace opens and you see it is divided into two main sections. The **Flow** pane at the top and the **Input** pane at the bottom.



   In Tableau Prep, the **Flow** pane is a canvas, much like Tableau Desktop, where you can interact with your data visually and build your flow. The **Input** pane contains configuration options about how the data is ingested. It also shows you the fields, data types, and sample values for your data set.

   We'll look at how you can interact with this data in the next section.

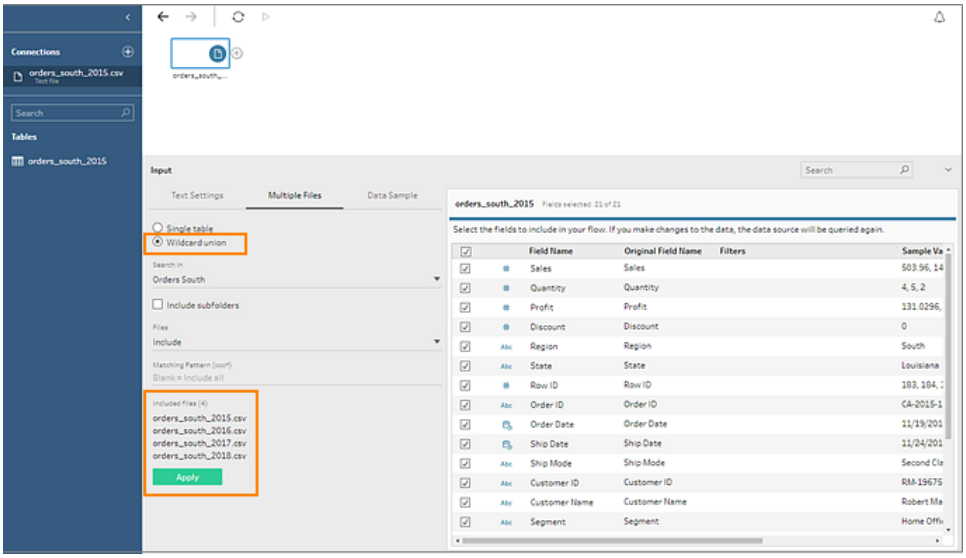
   > **Tip:** For single tables, Tableau Prep automatically creates an Input step for you in the **Flow** pane when you add data to your flow. Otherwise you can use drag-and-drop to add tables to the **Flow** pane.

4. You have three other files for your orders in the South. You could add each file individually, but you want to combine all the files together into one Input step, so you click the **Multiple Files** tab in the **Input** pane.

5. You see an option for **Wildcard union**. Select it.

   You notice that the directory where you selected your file is already populated and the other files you need are listed in the **Included files** section in the Input pane.

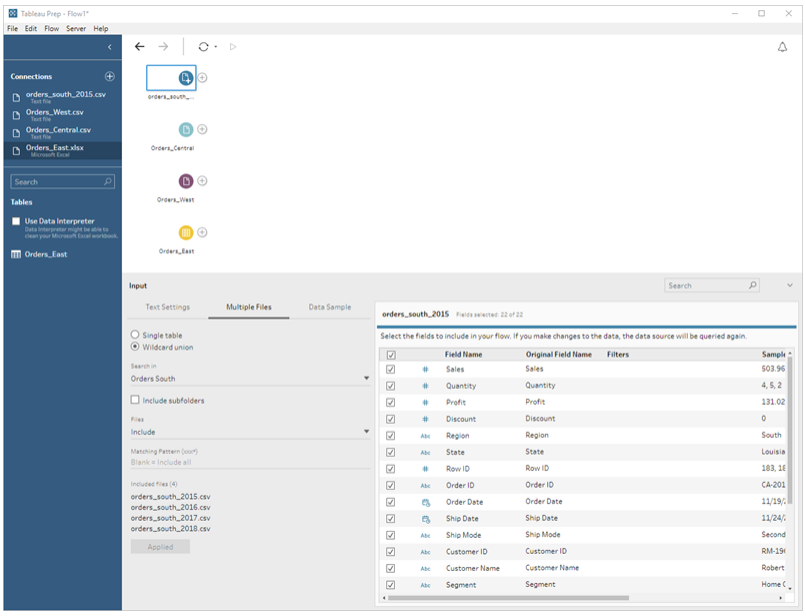
   > **Tip**: Using a wildcard union is a great way to connect to multiple files from a single data source with a similar name and structure. To use this option, the files must be in the same parent or child directory. If you don't see the files you need right away, change your search criteria. For more information, see **Union files in the Input step** on page 77.



6. Click **Apply** to add these files to add the data from these files to the **orders_south_2015** input step.

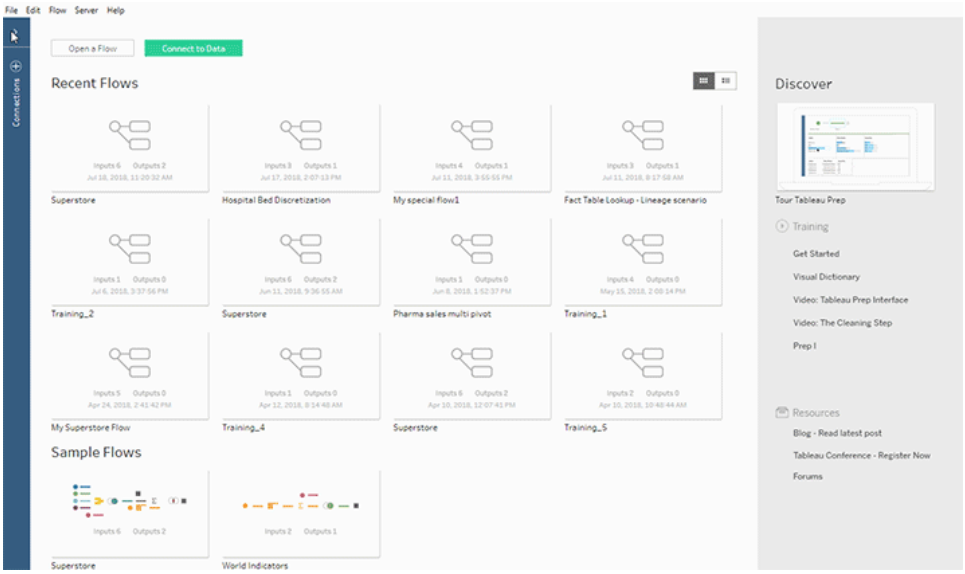
   The files for the other regions are all single table files, so you can select all of the files at once and add them to your flow.

7. Open File Explorer or Finder and navigate to the directory for the files. Ctrl+click (Command+click on Mac) to select the following files and drag-and-drop them onto the **Flow** pane to add them to your flow. (For file location, see **Wrap up and resources** on page 59.)
   - Orders_Central.csv
   - Orders_East.xlsx
   - Orders_West.csv



Check your work: Watch "Connect to data" in action.

Click the image to replay it



2. Explore your data

Now that you have the data files loaded into Tableau Prep, you're pretty sure that you want to combine the files together. But before you do that, it might be a good idea to take a look at them first and see if you can spot any issues.

When you select an Input step in the **Flow** pane, you can see the settings used to bring in the data, the fields that are included, and some sample values.

This is a good place to decide how much data you want to include in your flow and remove or filter fields that you don't want. You can also change any data types that were assigned incorrectly.

> **Tip**: If you are working with large data sets, Tableau Prep will automatically bring in a sample of the data to maximize performance. If you don't see the data you expect, you might need to adjust the sample. You can do this on the **Data Sample** tab. For more information about configuring your data options and sample size, see **Configure your data set** on page 81.

In the **Flow** pane, as you select each step and look over each data set, you notice a few things that you want to fix later and one thing that you can fix now in the **Input** step.

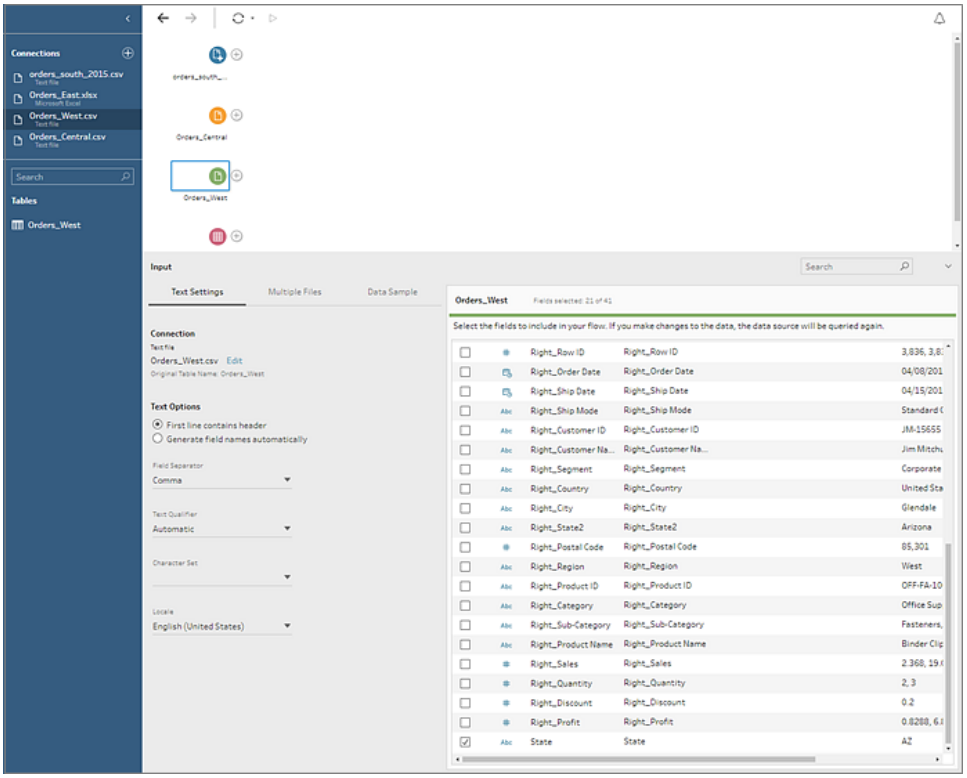
- In the **Flow** pane, click the **Orders_Central** Input step to select it. In the **Input** pane, you notice the following issues:
    - The order dates and ship dates are separated out into fields for month, day, and year.
    - Some of the fields have different data types than the same fields in other files.
    - There is no field for **Region**.

  You'll need to do some cleaning on these fields before you can combine this file with the others files. But you can't fix that here in the **Input** step, so you make a note to do this later.
- Select the **Orders_East** Input step.

  The fields in this file look like they align pretty well with the other files. But the **Sales** values all seem to have the currency code included. You'll need to fix that later, too.
- Select the **Orders_West** Input step. There are some issues in this file too.

- The **State** field uses abbreviations for the state name. Other files spell this out, so you'll need to fix that later.
- There are a lot of fields that start with **Right_**. These fields appear to be duplicates of the other fields. You don't want to include these duplicate fields in your flow. This is something you can fix here in the **Input** step:

  To fix this now, clear the check box for all fields that start with **Right_**. This tells Tableau Prep to ignore these fields and not to include them in the flow.



Now that you've identified a few troublemakers in your data sets, the next step is to examine your data a bit more closely and clean up any issues that you find so that you can combine and shape your data and generate an output file that you can use for analysis.

3. Clean your data

In Tableau Prep, examining and cleaning your data is an iterative process. After you decide on the data set that you want to work with, the next step is to examine and take action on that data by applying various cleaning, shaping, and combining operations to it. You apply these operations by adding steps to your flow.

Steps come in many flavors, depending on what you are trying to do. For example, add a cleaning step (**Add Step**) any time you want to apply cleaning operations to your fields like filter, merge, split, rename, and so on. Add an aggregation step (**Add Aggregate**) to group and aggregate fields and change the level of detail of your data. For more information about the different step types and their uses, see **Build your flow** on page 87.

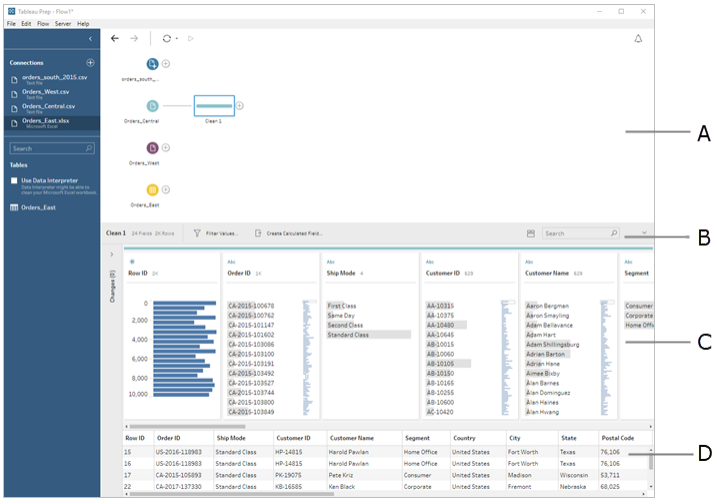
> **Tip**: As you add steps to your flow, a flow line is automatically added to connect the steps to one another. You can move these flow lines around and remove or add them as needed.
>
> When you run your flow, these connection points are required so Tableau Prep knows which steps are connected and in which order the steps apply in the flow. If a flow line is missing, the flow will be broken and you'll get an error.

Clean Orders_Central

To address the issues you noticed earlier and to see if there are any other issues, you start by adding a cleaning step to the **Orders_Central** Input step.

1. In the **Flow** pane, select **Orders_Central**, click the plus ⊕ icon and select **Add Step**.

   When you add a cleaning step to your flow, the workspace changes and you see the details of your data.



A. Flow pane, B. Toolbar, C. Profile pane, D. Data grid

The workspace is now split into three parts: the **Flow** pane, the **Profile** pane with a toolbar, and the **Data** grid. The **Profile** pane shows you the structure of your data, summarizing the field values into bins so that you can quickly see related values and spot outliers and null values. This is where you will perform most of your cleaning tasks.

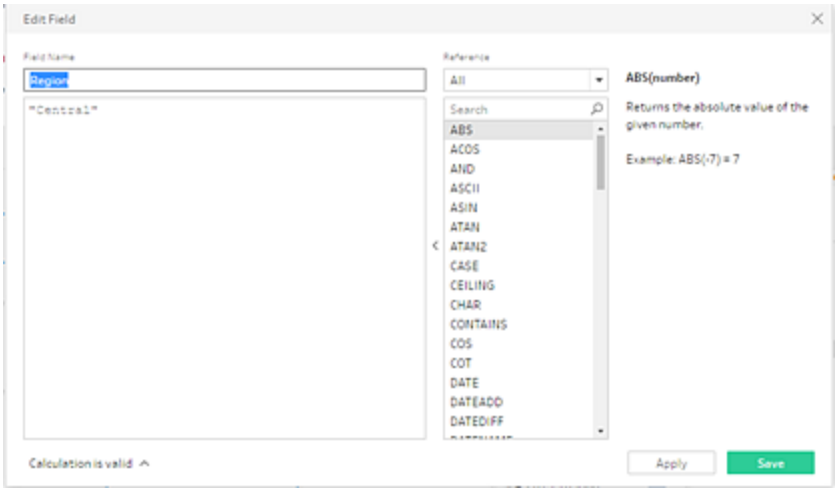
The **Data** grid shows you the row level detail for your fields.

> **Tip**: Each field in the **Profile** pane is shown on a profile card. Use the drop-down arrow on each card to see and select the different cleaning options that are available for that field type. You can also sort the field values, change the data type, or drag and drop the profile cards and the columns in the **Data** grid to rearrange them.

Clean data with calculated fields

This data set is missing a field for **Region**. Since the other data sets have this field you'll need to add it so that you can combine your data later. You'll need to use a calculated field to do this.

2. In the toolbar, click **Create Calculated Field**.
3. Name the calculated field **Region**. Then enter **"Central"** (including the quotes) and click **Save**.



You love the flexibility of being able to use calculated fields to shape you data. You are pleased to see that Tableau Prep uses the same calculation editor language as Tableau Desktop.

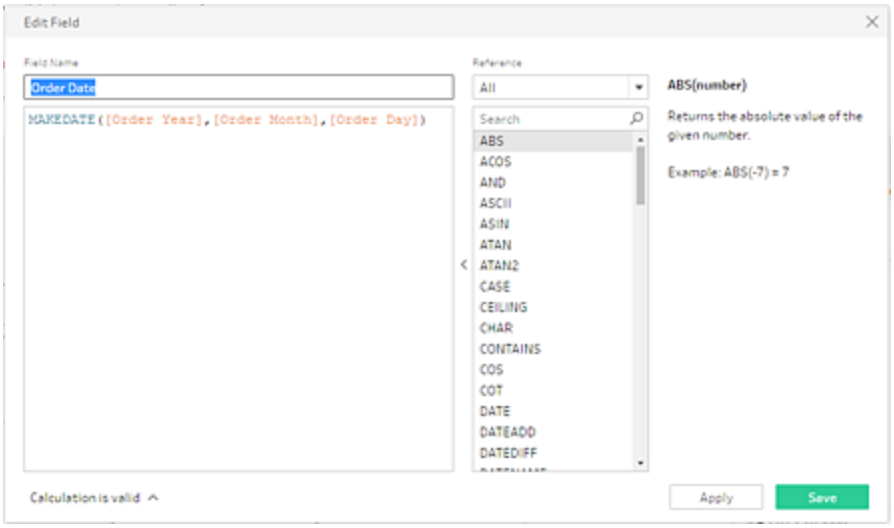
> **Tip**: When you make changes to your fields and values, Tableau Prep keeps track of them in the **Changes** pane. An icon representing the change is also added to the cleaning step in the flow and to the field in the **Profile** pane. We'll look at the **Changes** pane after making more changes.

Next you want to address the separate order date and ship date fields. You want to combine them into two single fields, one for **Order Date** and one for **Ship Date** so they align with the same fields in the other data sets.

You can use a calculated field again to do this in one easy step.

4. In the toolbar, click **Create Calculated Field** to combine the **Order Year**, **Order Month**, and **Order Day** fields into one field with the format "MM/DD/YYYY".
5. Name the calculated field **Order Date**. Then enter the following calculation into the Calculation editor and click **Save**:

```
MAKEDATE([Order Year],[Order Month],[Order Day])
```



Now that you have a new field for your order date, you want to remove the existing fields, as you no longer need them.

You have a lot of fields in the **Profile** pane. You notice a **Search** box in the top right corner on the toolbar. You wonder if you can use that to quickly find the fields that you want to remove. You decide to give it a try.

6. In the **Profile** pane, in the search box, type **Order**.

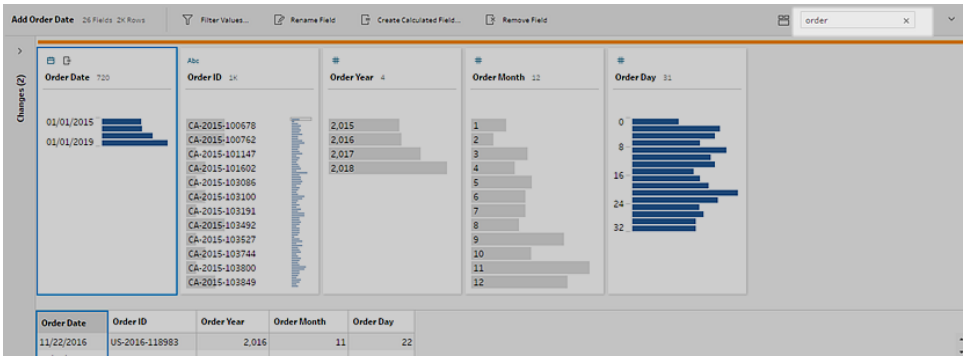
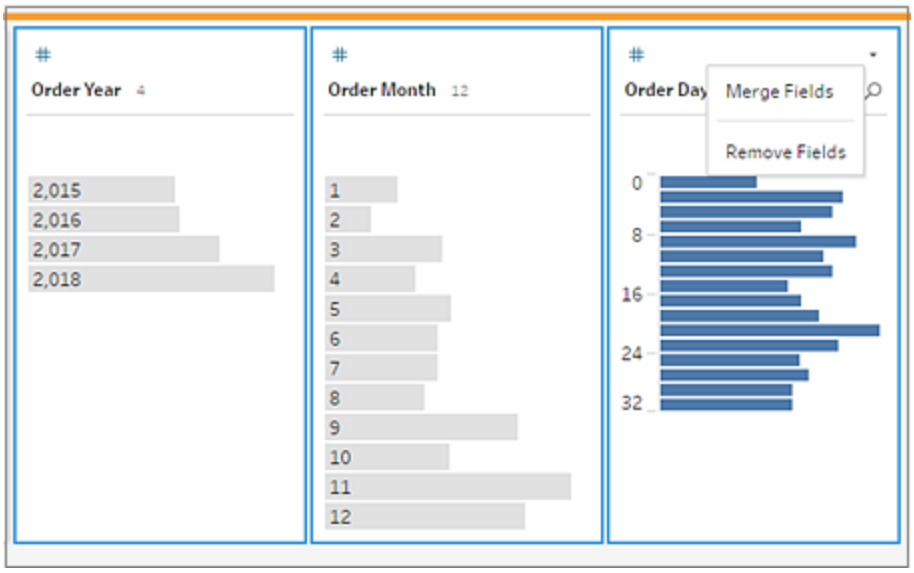


Tableau Prep quickly scrolls all the fields with **Order** in the name into view. Cool!

7. Ctrl+click (Command+click on Mac) to select the fields for **Order Year**, **Order Month**, and **Order Day**. Then right-click on the selected fields and select **Remove Field** from the menu to remove them.



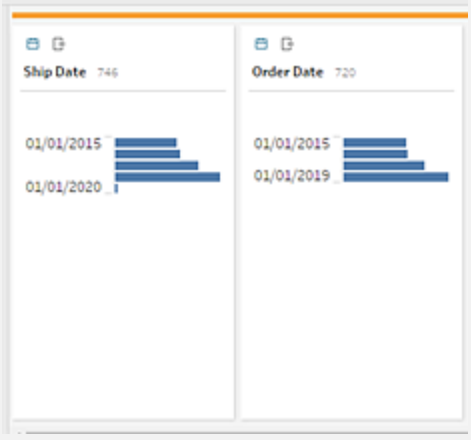
8. Now repeat steps 4 though 7 above to create a single field for **Ship Date**. Try it on your own or use the steps below to help you.

- In the toolbar, click **Create Calculated Field** to combine the **Ship Year**, **Ship Month**, and **Ship Day** fields into one field with the format "MM/DD/YYYY".
- Name the calculated field **Ship Date** and enter the calculation `MAKEDATE ([Ship Year],[Ship Month],[Ship Day])`. Then click **Save**.
- Remove the **Ship Year**, **Ship Month**, and **Ship Day** fields. Search for the fields, select them, and select **Remove Field** from the menu.

Tip: Tableau Prep summarizes the data in the Profile pane into bins to help you quickly see the shape of your data, find outliers, spot relationships between fields, and so on.

In this scenario, the order and ship dates can now be summarized by year. Each bin represents a year from January of the beginning year to January of the following year and is labeled accordingly. Because there are sales dates and ship dates that fall in the latter part of 2018 and 2019, we get a bin for that data that is labeled with the ending year 2019 and 2020 accordingly.

To change this view to the actual dates, click the drop-down arrow in the Profile card and select **Detail**.



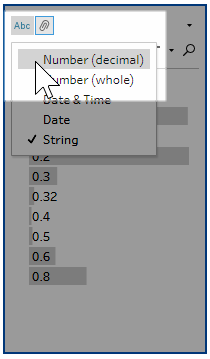
Interact directly with fields to clean your data

Your data is starting to look good. But, as you finish removing the extra fields for the order and ship dates, you notice that the **Discounts** field has a couple of issues.

- It's assigned to a **String** data type instead of a **Number (decimal)** data type.
- There's a field value **None** instead of a numeric value for no discount.

This will cause a problem when you combine the files, so you better fix that too.

9. Clear your search and enter **disc** in the search box to find the field.
10. Select the **Discounts** field, double-click the field value **None**, and change it to the numeric value **0**.
11. Change the data type for the **Discount** field from **String** to **Number (decimal)**. Click **Abc** and select **Number (decimal)** from the drop-down menu.



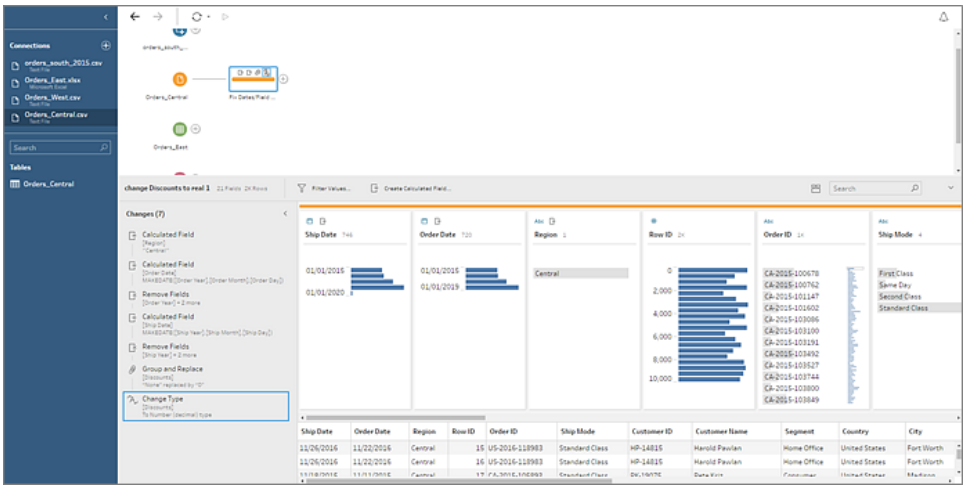
12. Finally name your step to help keep track of what you did in this step. In the **Flow** pane, double-click the step name **Clean 1** and type in **Fix dates/field names**.

Review your changes

You made a lot of changes to this data set and you start to worry that you won't remember everything you did. As you look over your work, you see a column on the left of the **Profile** pane called **Changes**.

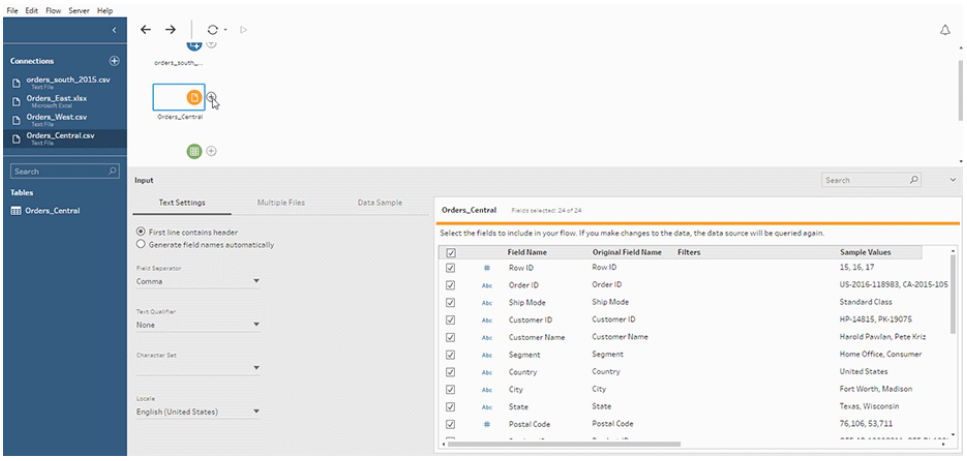
You click the arrow to open it and are delighted to see a list of every change you just made. As you scroll through the changes in the list, you notice that you can delete or edit your changes or even move them around to change the order that you did them in.

You love that you can easily find the changes you made in any step as you build your flow and experiment with the order of those changes to get the most out of your data.



Check your work: Watch "Clean Orders_Central" in action.

Click the image to replay it



Now that you've cleaned one file, you take a look at the other files to see what other issues you need to fix.

You decide to look at the Excel file for **Orders_East** next.

Clean Orders_East

As you look over the fields for the **Orders_East** file, most of the fields look like they align with the other files, except for **Sales**. To take a closer look and see if there are any other issues to address, you add a cleaning step to the **Orders_East** Input step.

1. In the **Flow** pane, select **Orders_East**, click the plus ⊕ icon and select **Add Step**.

   Looking at the **Sales** field you quickly see that the **USD** currency code has been included with the sales numbers, and Tableau Prep interpreted these field values as a string.

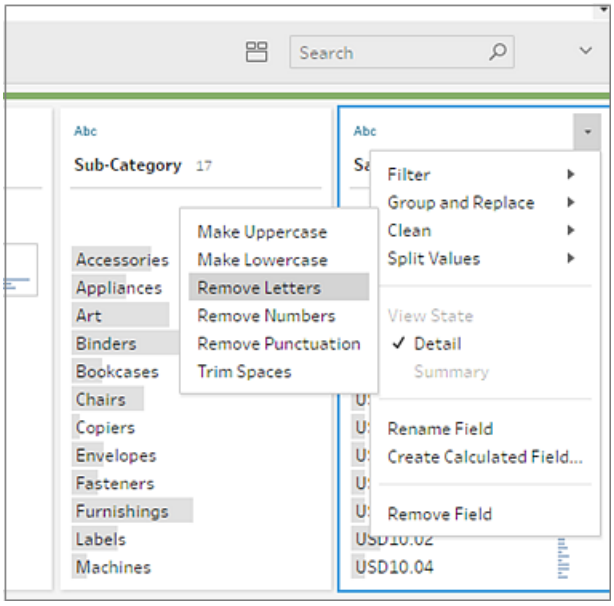
   You'll need to remove the currency code from this field and change the data type if you want to get accurate sales data.

   Fixing the data type is easy, you already know how to do that. But there are over 2ooo unique rows of sales data and fixing every individual row to remove the currency code seems cumbersome.

   But this is Tableau Prep, and you decide to check out the drop-down menu to see if there is an option to fix this.

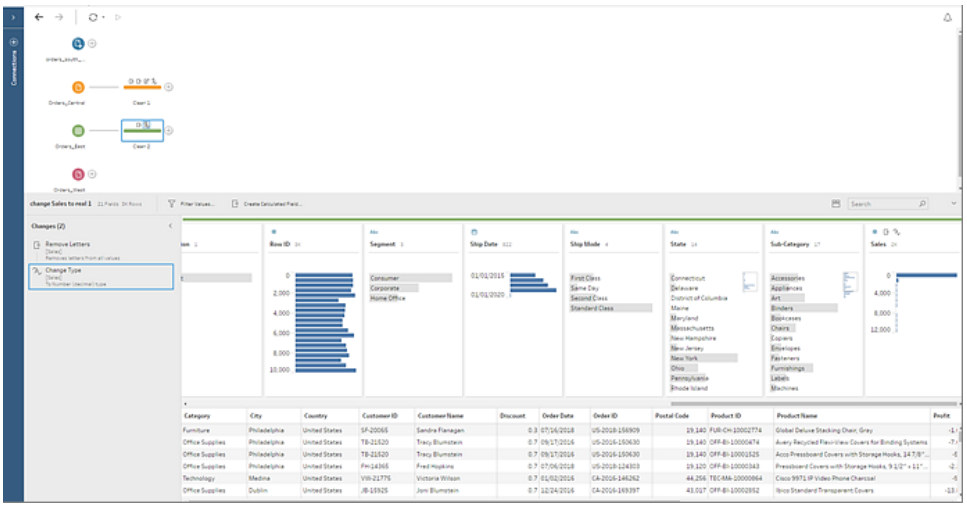
   When you click the drop-down arrow for the **Sales** field, you see a menu option called **Clean** and an option under that to remove letters. You decide to give that a try and see what it does.

2. Select the **Sales** field. Click the drop-down arrow and select **Clean** > **Remove Letters**.



   Wow! That cleaning option instantly removed the currency code from every field. Now you just need to change the data type from **String** to **Number (decimal)** and this file is looking good.

3. Click the data type and select **Number (decimal)** from the drop-down list.



4. The rest of the file looks pretty good. Name your cleaning step to keep track of your work. For example, **Change data type**.

Next you look at your last file for **Orders_West** to see if there are any issues there that you need to fix.

Clean Orders_West

As you look over the fields for the **Orders_West** file, most of the fields look like they align with the other files, but you remember seeing that the **States** field used abbreviations for the values instead of spelling out the state name. To combine this file with the other files, you'll need to fix this. So you add a cleaning step to the **Orders_West** Input step.

1. In the **Flow** pane, select **Orders_West**, click the plus ⊕ icon and select **Add Step**.

   Scroll or use Search to find the **State** field. You see that all the state name values use the short abbreviation. There are only 11 unique values for this field. You could manually change each one, but maybe Tableau Prep has another way to do this?

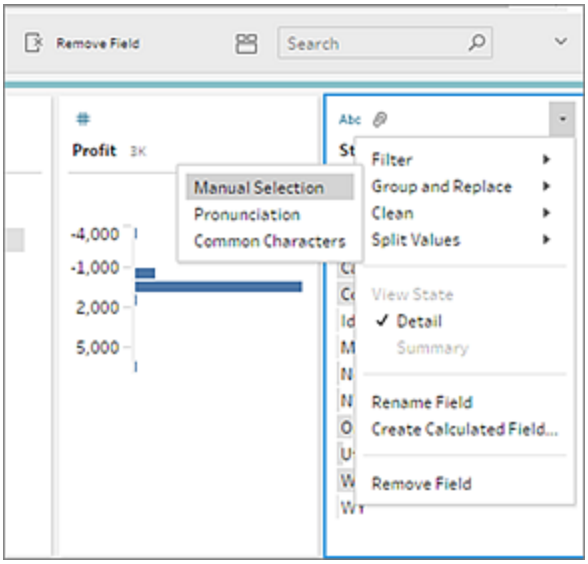
   You click the drop-down menu for the field and see an option called **Group and Replace**. When you select it you see three options:

   - Manual Selection
   - Pronunciation
   - Common Characters

The state names don't sound alike and don't share the same characters, so you decide to try the **Manual Selection** option.

> **Tip**: You can double-click a field name or field value to edit a single value. To edit multiple values you can select all the values and use the right-click menu option **Edit Values**. But when you want to map one or more values to specific values, use the **Group and Replace** option in the drop-down menu.
> For more information about editing and grouping values, see **Cleaning (fixing) variations of the same value** on page 96.

2. Select the **State** field. Click the drop-down arrow and select **Group and Replace** > **Manual Selection**.



A two column card opens. This is the **Group and Replace editor**. The column on the left shows the current field values and the column on the right shows the fields that are available to map to the fields on the left.

You want to map your state abbreviations to the spelled out version of the state name, but you don't have those values in the **Orders_West** data set. You wonder if you can just edit the name directly and maybe add it there, so you give that a try.

3. In the **Group and Replace editor** in the left pane, double-click **AZ** to highlight the value and type **Arizona**. Then press **Enter** to add your change.

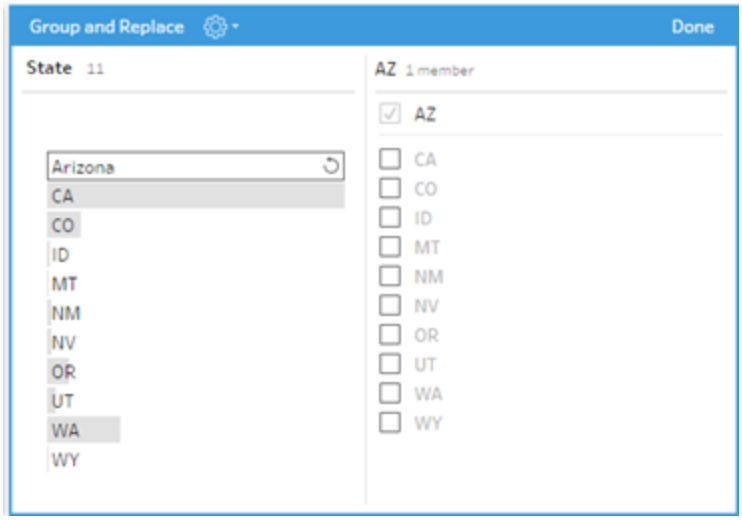


Tableau Prep created a mapped value for your new value **Arizona** and automatically mapped the old value, **AZ** to it. Having a mapped relationship set up for these values will save you time if you get more data from this region entered like this.

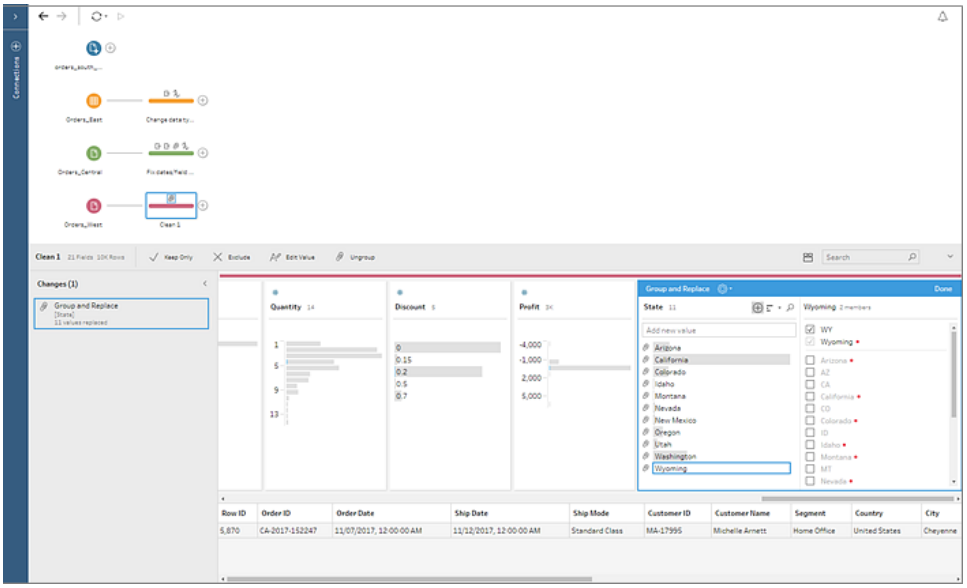
> **Tip**: You can add field values that aren't in your data sample to set up mapping relationships to organize your data. If you refresh your data source and new data is added, you can add the new data to the mapping instead of manually fixing each value.
>
> When you manually add a value that isn't in your data sample, the value is marked with a red dot to help you easily identify it.

4. Repeat these steps to map each state to the spelled out version of its name.

Abbreviation	State Name
AZ	Arizona
CA	California
CO	Colorado
ID	Idaho
MT	Montana

NM	New Mexico
NV	Nevada
OR	Oregon
UT	Utah
WA	Washington
WY	Wyoming

Then click **Done** to close the **Group and Replace** editor.



After all the states are mapped, you look at the **Changes** pane and see there is only one entry there instead of 11.

It looks like Tableau Prep groups similar actions for a field together. You like that because it will make it easier to find changes you made to your data set later.

Fixing the **State** field values was the only change you needed to make here.

5. Name your cleaning step to keep track of your work. For example **Rename states**.

You've done a lot of clean up in your files, and you can't believe how quick and easy it was. You might make it home for dinner after all! To make sure that you don't lose all of your work so far, save your flow.

Click **File** > **Save** or **File** > **Save As**. Save your file as a flow file (.tfl) and give it a name. For example, **My Superstore**.

> **Tip**: When you save your flow files, you can either save them as a flow file (.tfl) or you can save them as a packaged file (.tflx) and package your local data files with them to share the flow and files with someone else. For more information about saving and sharing your flows, see **Save and Share Your Work** on page 136.

4. Combine your data

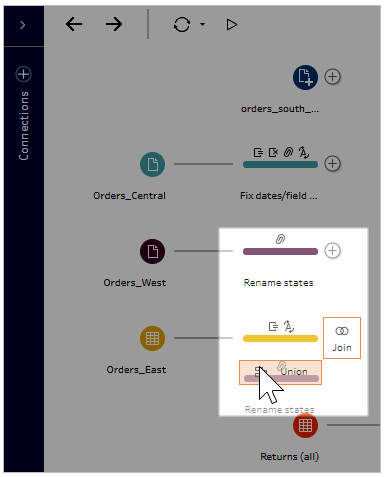
Now that all the files are cleaned up, you are finally ready to combine them all.

Because all the files have similar fields, you want to union the files together to add the rows from each file into a single table.

You remember that there was a step option called **Add Union**, but you wonder if you can simply drag and drop the steps to union them. You decide to try it and see.

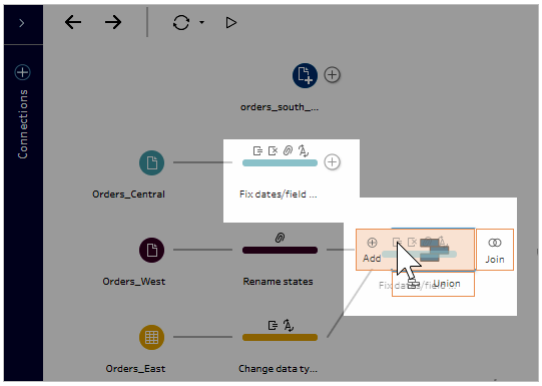
Union your data

1. In the **Flow** pane, drag the cleaning step **Rename states** to the **Changed data type** step and drop it on the **Union** option.



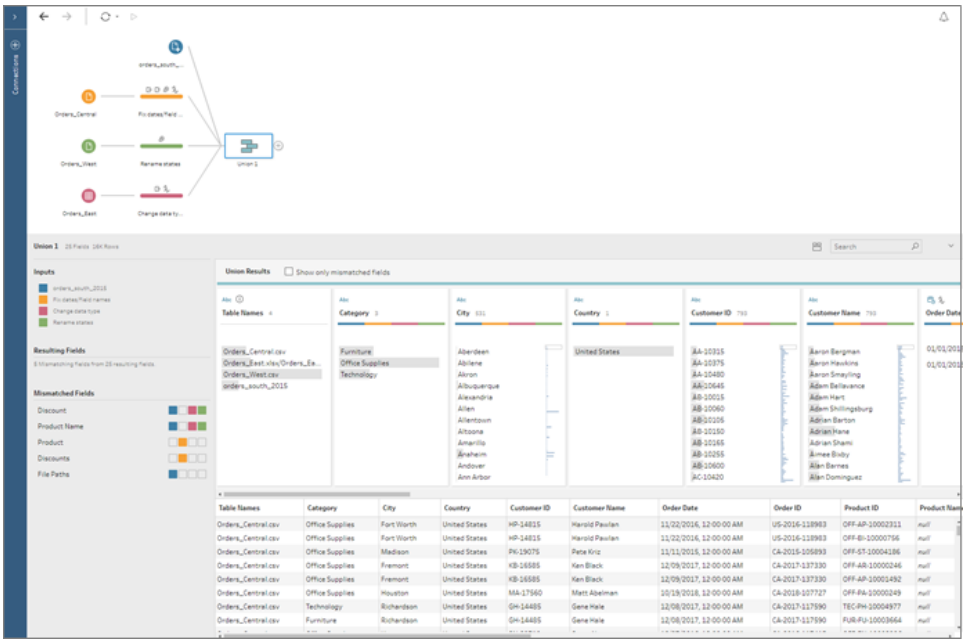
   You see that Tableau Prep added a new **Union** step to your flow. Great! Now you want to add the other files to this union too.

2. Drag the **Fix dates/field names** step to the new **Union** step. Drop it on **Add** to add it to the existing union.



3. Drag the **orders_south_2015** step to the new **Union** step. Drop it on **Add** to add it to the existing union.

   Now all of your files are combined into a single table. In the **Flow** pane, select the **Union** step to see your results.



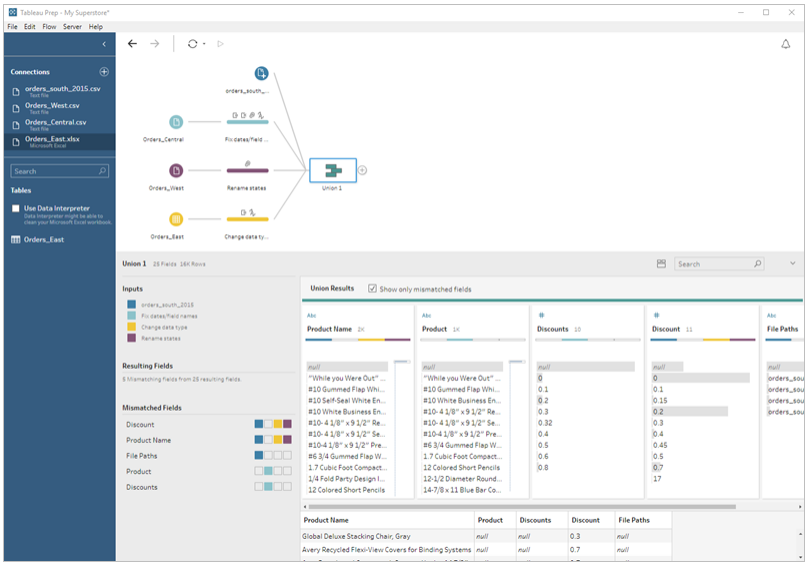
   You notice that Tableau Prep automatically matched up the fields that had the same names and types.

You also see that the **colors** assigned to the steps in the flow are used in the union profiles to indicate where the field came from and also appear in the **colored band** across the top of each field to show you if that field exists in that table.

You notice that a new field called **Table Names** was added that lists the tables where all the rows in the union come from.

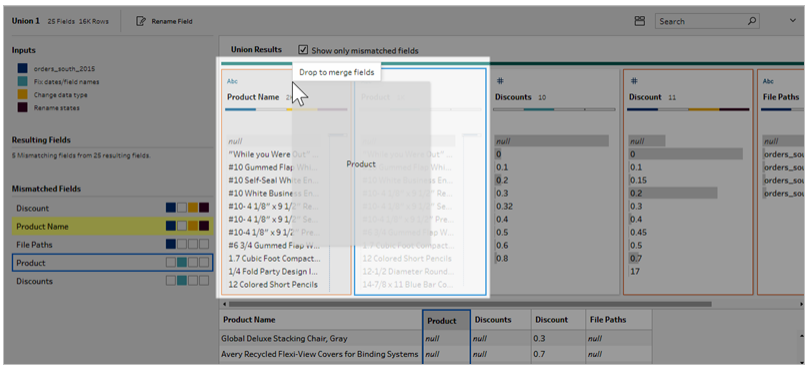
A list of mismatched fields also shows in the summary pane and you can see right away that the fields **Product** and **Discounts** only appear in the **Orders_Central** file.

4. To take a closer look at these fields, in the **Union Results** pane, select the **Show only mismatched fields** check box.



Looking at the field data, you quickly see that the data is the same, but the field name is different. You could simply rename the field, but you wonder if you could just drag and drop these fields to merge them. You decide to try that and see.

5. Select the **Product** field and drag and drop it onto the **Product Name** field to merge the fields. After the fields are merged, they no longer appear in the pane.



6. Repeat this step to merge the **Discounts** field with the **Discount** field.

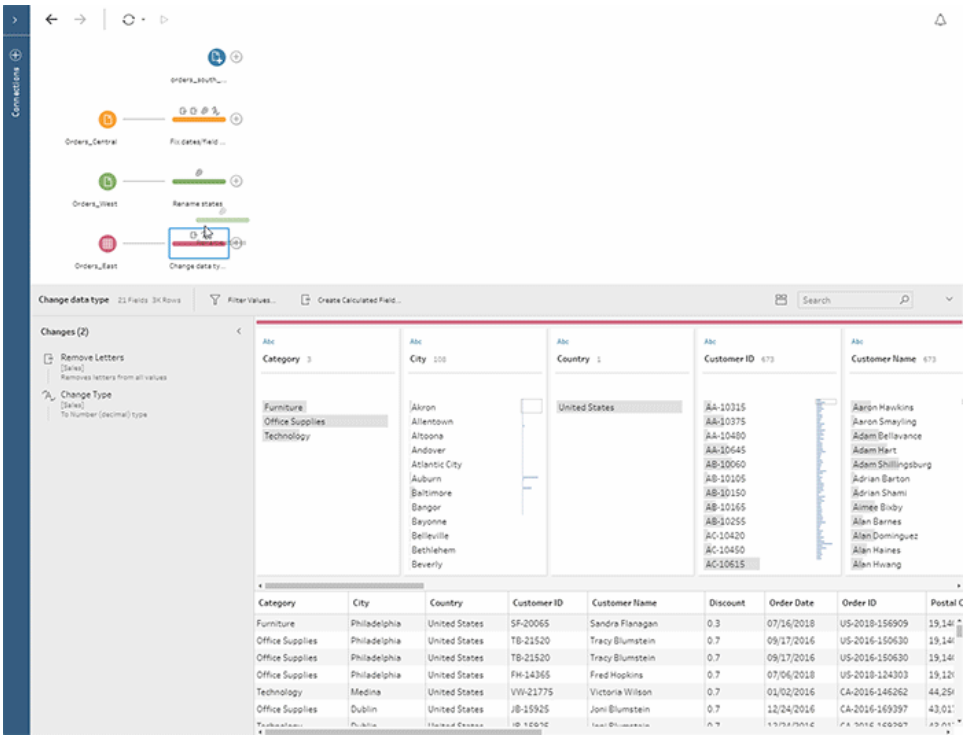
   The only field that doesn't have a match now is the **File Paths** field. This field shows the file paths for the wildcard union that you did for your sales orders from the South. You decide to leave this field there as it has good information.

   > **Tip**: You have several options when fixing mismatched fields after a union.
   >
   > Depending on the Tableau Prep version you are using, you can select a field in the **Mismatched Fields** list (version 2018.2.1 and later) and if Tableau Prep detects a possible match, it will highlight it in yellow. To merge the fields hover over the highlighted field and click the plus ⊕ button that appears.
   >
   > For more ways to merge fields in a union, see **Fix fields that don't match** on page 132.

7. Clear the **Show only mismatched fields** check box to show all the fields included in the union.
8. Name your Union step to represent what this union includes work. For example, **All orders**.

Check your work: Watch "Union your data" in action.

Click the image to replay it



As you are admiring the results of your cleaning prowess, your boss calls. He forgot to mention that he also wants you to include any product returns in your analysis. He hopes that won't be too much trouble. With Tableau Prep in your toolkit, it's not a problem.

Clean the product returns data

You look over the Excel file that your boss sent you for product returns and it looks a little messy. You add the new file **return_reasons new** to your flow to take a closer look.

1. In the **Connections** pane, click **Add connection**. Select **Microsoft Excel**and navigate to the sample Superstore data files (see **Wrap up and resources** on page 59 for the file location).

2. Select **return reasons_new.xlsx**, and then click **Open** to add the file to the flow pane.

   There are only 4 fields that you want to include from this file in your flow: **Order ID**, **Product ID**, **Return Reason** and **Notes**.

3. In the **Input** pane for **returns_new** clear the check box at the top of the field grid to clear all the check boxes. Then select the check box for the **Order ID**, **Product ID**, **Return Reason** and **Notes** fields.

4. Rename the Input step to better reflect the data that is included. In the **Flow** pane,

double-click the Input step name **Returns_new** and type in **Returns (all)**.

Looking at the sample field values, you notice that the **Notes** field seems to have a lot of different data combined together.

You have some cleaning to do in this file before you can do any further work with the data, so you add a cleaning step to check it out.

5. In the **Flow** pane, select the Input step **Returns (all)**, click the plus ⊕ icon, and then select **Add Step**.

   In the **Profile** pane, click and drag the outer right edge of the field to the right to re-size the **Notes** field so you can see the entries better.

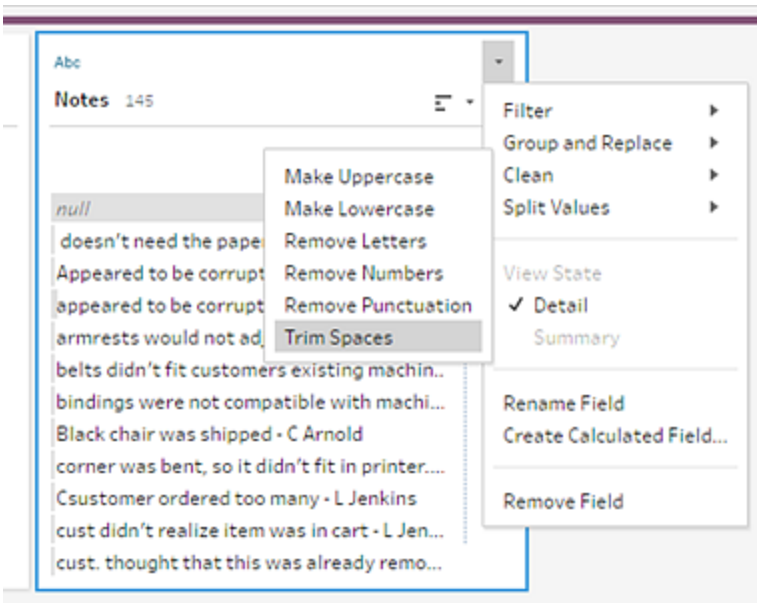
6. In the **Notes** field, use the visual scroll bar to the right of the field values to scan the values.

   You notice a few things that are problematic:

   - Some of the entries have an extra space in the entry. This can result in the field being read as a null value.
   - It looks like the name of the approver is included in the return notes entry. To better work with this data you'll want that information in a separate field.

   To tackle the extra spaces, you remember that there was a cleaning option to remove trailing spaces, so you decide to try that to see if it can fix that problem.

7. Select the **Notes** field. Click the drop-down arrow and select **Clean** > **Trim Spaces**.



Yes! It did exactly what you wanted it to do. The extra spaces are gone.

Next you want to create s separate field for the approver name. You see a **Split Values** option in the menu, so you decide to try that.

8. Select the **Notes** field. Click the drop-down arrow and select **Split Values** > **Automatic Split**.

   This option did exactly what you were hoping it would do. It automatically split the return notes and the approver name into separate fields.

   Just like Tableau Desktop, Tableau Prep automatically assigned a name to those fields. So you'll need to rename the new fields to something meaningful.

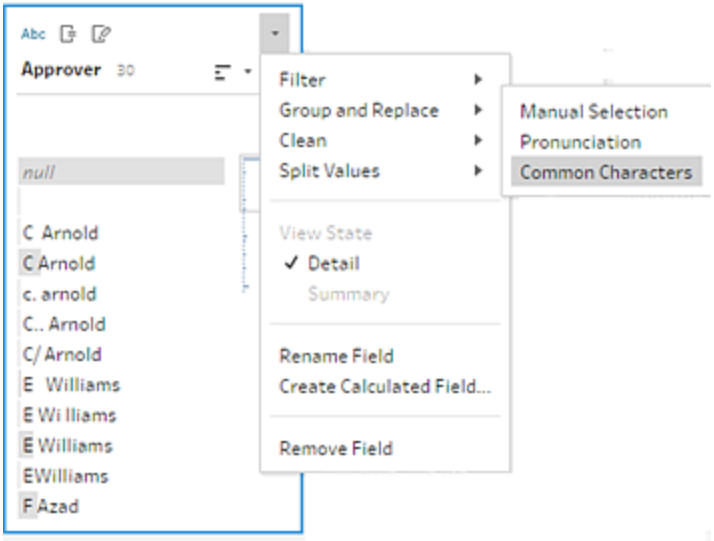
9. Select the field **Notes-Split 1**. Double-click in the field name and type **Return Notes**.
10. Repeat this step for the second field and rename it to **Approver**.
11. Finally remove the original **Notes** field, as you no longer need it. Select the **Notes** field, click the drop-down arrow, and select **Remove Field** from the menu.

    Looking at the new **Approver** field, you notice that the field values lists the same names but they are entered differently. You want to group them to eliminate multiple variations of the same value.

    Maybe the **Group and Replace** option can help with that?

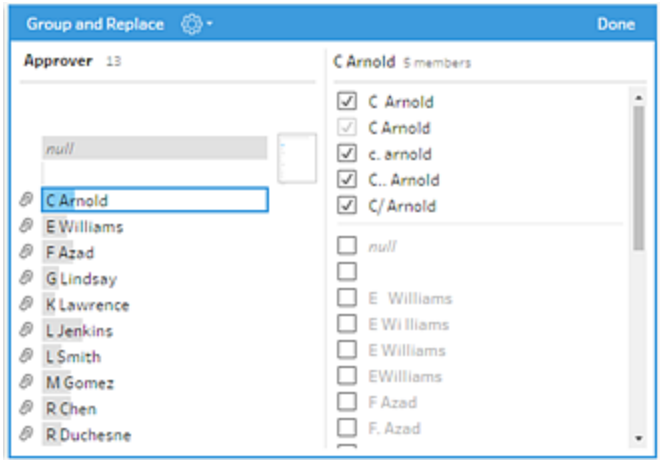
You remember there was an option for **Common Characters**. Since these values share the same letters, you decide to try that.

12. Select the **Approver** field. Click the drop-down arrow and select **Group and Replace** > **Common Characters**.



This option grouped all of the variations of each name together for you. That's exactly what you wanted to do.

After checking the other names to make sure they are grouped properly, you click **Done** to close the **Group and Replace editor**.



This file is looking pretty good.

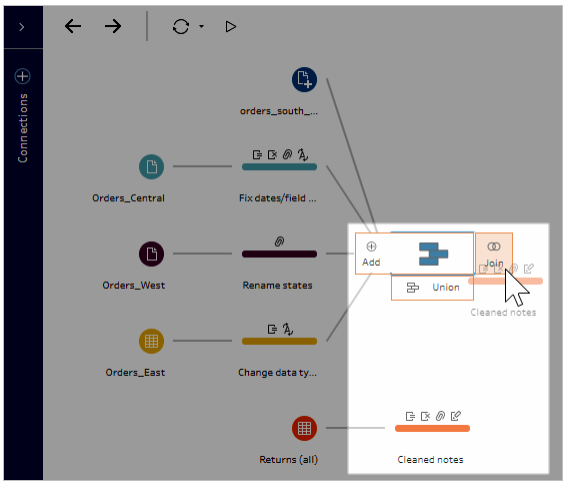
13. Name your cleaning step to keep track of your work. For example **Cleaned notes**.

Now that the product return data is all cleaned up, you want to add this data to the orders data in your unioned files. But many of these fields don't exist in the unioned files. To add these fields (columns of data) to your unioned data set, you need to use a join.

Join your data

When you join data, the files must have at least one field in common. Your files share the **Order ID** and **Product ID** fields, so you can join on those fields to see all the rows that have those fields in common. You remember an option to create a join when you created your union using drag and drop, so you give that a try.

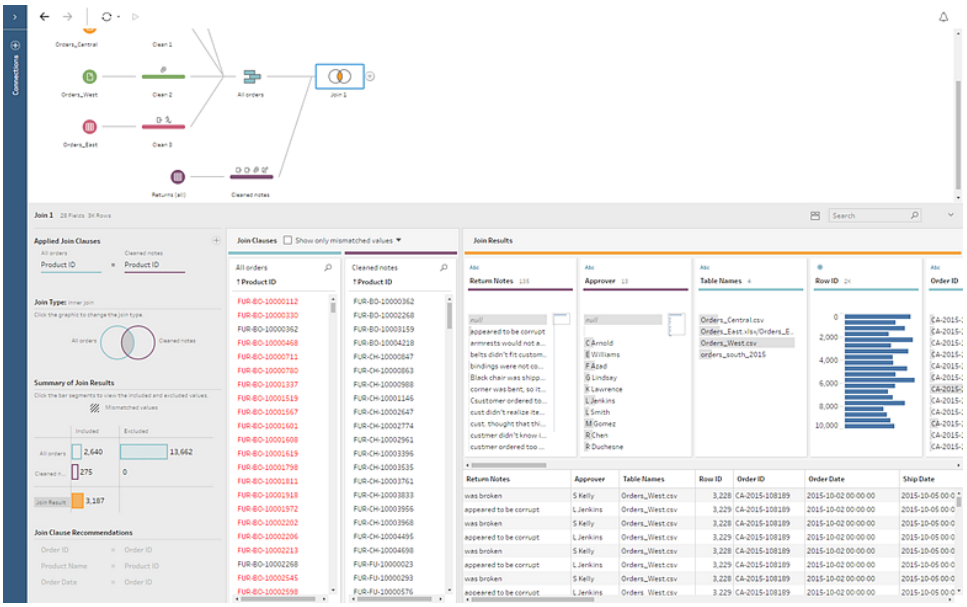
1. In the **Flow** pane, drag the **Cleaned notes** step on to the **All orders** Union step and drop it on **Join**.



When you join files, Tableau Prep shows you the results of your join in the **Join Profile**.

Working with joins can be tricky. You often want to have a clear view of the factors that are included in the join, such as the fields used to join the files, the number of rows included in the results and any fields that aren't included or are null values.

As you review the results of the join in Tableau Prep, you are delighted to see so much information and interactivity at your fingertips.



Tip: The far left pane of the join profile is where you can explore and interact with your join. You can also fix values directly in the **Join Clauses** panes.

Choose the fields that you want to join on in the **Applied Join Clauses** section or add suggested join clauses from the **Join Clause Recommendations** section.

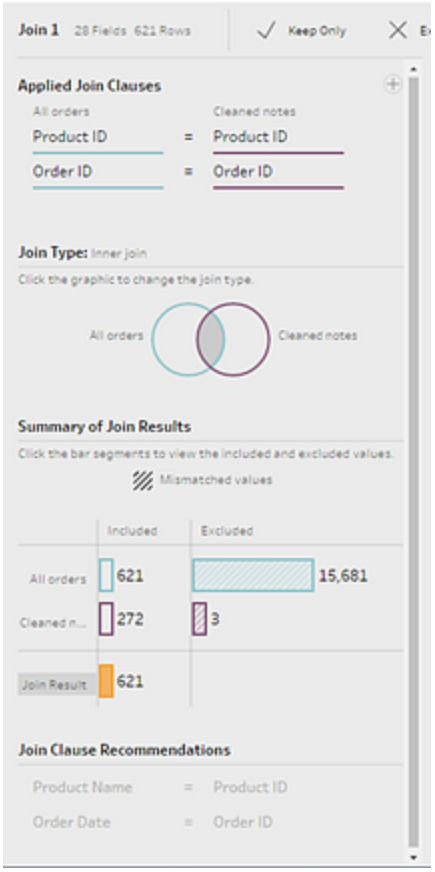
Click in the **Join Type** diagram to try different join configurations and see the number of rows included or excluded in your join for each table in the **Summary of Join Results** section.

For more information about working with joins, see **Join or Union Data** on page 124.

You see that you have over 13,ooo rows excluded from your **All Orders** files. When you created your join, Tableau Prep automatically joined on the **Product ID** field, but you wanted to also join on the **Order ID** field.

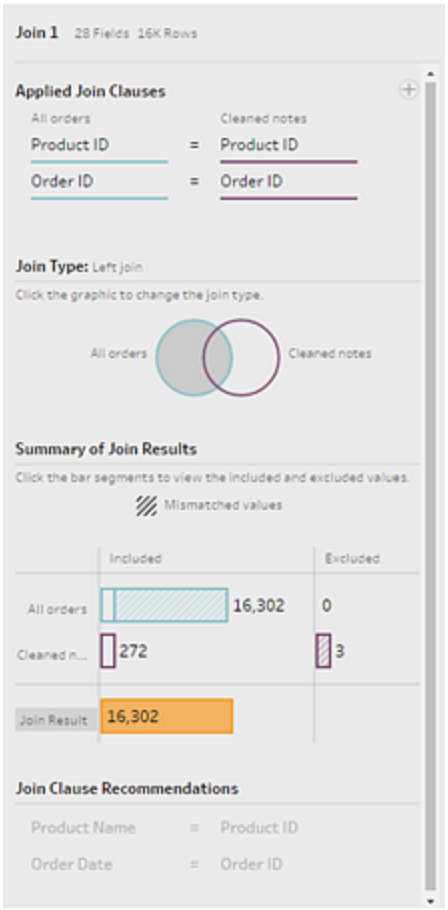
As you scan the left pane of the join profile, you see that **Order ID** is in the list of recommended join clauses, so you quickly add it from there.

2. In the left pane of the **Join** profile, in the **Join Clause Recommendations** section, select **Order ID = Order ID** and click the plus button to add the join clause.



Because the **Join Type** is set to an inner join (the default setting for Tableau Prep), the join is only including values that exist in both files. But you want all of the data from your **Orders** files as well as the return data for those files. So you'll need to change the join type.

3. In the **Join Type** section, click the left side of the diagram to change the join type to a **Left** join.



Now you have all of the data from the sales order files and any return data that apply to those orders. You review the **Join Clauses** pane and see the distinct values that don't exist in the other file.

For example there are many order rows (shown in red) that have no corresponding return data. You love being able to explore this level of detail about your join.

You're anxious to start analyzing this data in Tableau Desktop, but you notice a few results from the join that you want to clean up before you do that. Good thing you know what to do!

> **Tip**: Wonder if your data is clean enough? You can preview your data in Tableau Desktop from any step in your flow to check it out.
>
> Just right-click on the step in the **Flow** pane and select **Preview in Tableau**

Desktop from the menu.

You can experiment with your data and any changes that you make in Tableau Desktop won't write back to your data source in Tableau Prep. For more information see **View your data sample in Tableau** on page 137.

4. Before you start cleaning your join results, name your **Join** step **Orders+Returns** and save your flow.

Clean your join results

To clean up the fields in your join, you'll need to add a cleaning step.

1. In the **Flow** pane, select **Orders+Returns**, click the plus icon, and then select **Add Step**.

   When you joined the two steps, the common fields **Order ID** and **Product ID** were added for both tables.

   You want to keep the **Product ID** field from all of your orders and the **Order ID** field from the returns file and remove the duplicate fields that came from those files. You also don't need the **File Paths** and **Table Names** fields in your output file, so you want to remove those fields as well.

   Tip: When you join tables using fields that exist in both files, Tableau Prep will bring in both fields and rename the duplicate field from the second file by adding a "-1" or a "-2" to the field name. For example **Order ID** and **Order ID-1**.

2. In the **Profile** pane, select and remove the following fields:
   - **Table Names**
   - **Order ID**
   - **File Paths**
   - **Product ID-1**
3. Rename the field **Order ID-1** to **Order ID**.

You have quite a few null values where the product was returned but there was no return note or approver indicated. To make this data easier to analyze, you want to add a field with a value of **Yes** and **No** to indicate whether the product was returned.

You don't have this field, so you can add it by creating a calculated field.

4. In the toolbar, click **Create Calculated Field**.
5. Name the field **Returned?** and then enter the following calculation and click **Save**.

   ```
   If ISNULL([Return Reason])=FALSE THEN "Yes" ELSE "No" END
   ```

   For your analysis you would also like to know the number of days it takes to ship an order, but you don't have that field either.

   You have all the information that you need to create it though, so you add another calculated field to create it.

6. In the toolbar, click **Create Calculated Field**.
7. Name the field **Days to Ship** and then enter the following calculation and click **Save**.

   ```
   DATEDIFF('day',[Order Date],[Ship Date])
   ```

8. Name your step **Clean Orders +Returns**.
9. Save your flow.

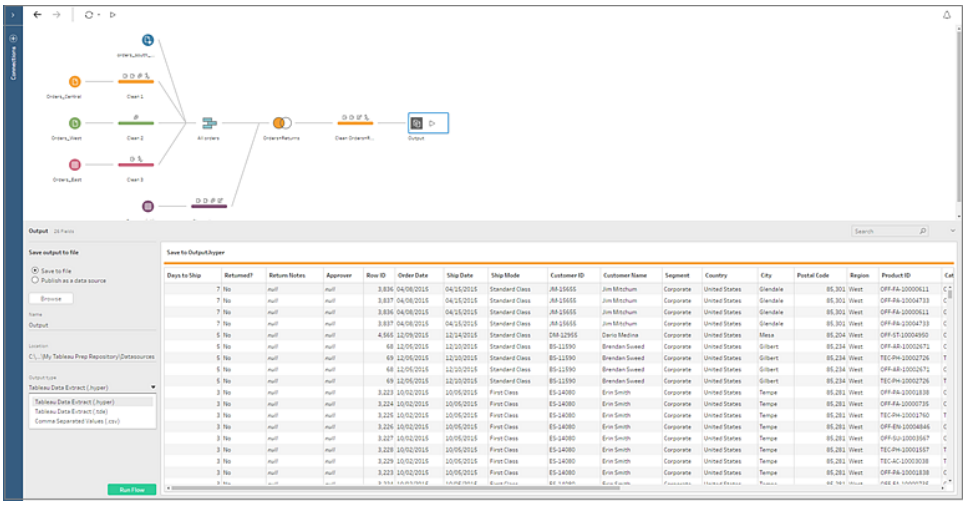
5. Run your flow and generate output

Your data is looking good and you're ready to generate your output file to start analyzing it in Tableau Desktop. All you need to do is run your flow and generate your extract file. To do this you need to add an **Output** step.

1. In the **Flow pane**, select **Clean Orders+Returns**, click the plus ⊕ icon and select **Add Output**,

   When you add an Output step, the **Output** pane opens and shows you a snapshot of your data. Here you can select the type of output that you want to generate, and specify the name and where you want to save the file.

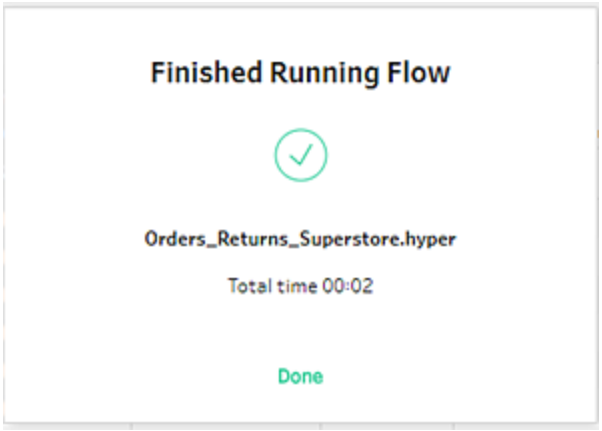
   The default location is in the **My Tableau Prep** repository in your data sources folder.



2. In the left pane select **Save to file**.
3. Click the **Browse** button, then in the **Save Extract As** dialog, enter a name for the file, for example **Orders_Returns_Superstore**, and click **Accept**.
4. In the **Output type** field, select the output type. Depending on the version of Tableau Desktop you use you can choose from the following options:
    - Tableau Data Extract (.hyper) for Tableau Desktop version 10.5 and later.
    - Tableau Data Extract (.tde) for Tableau Desktop version 10.0 through 10.4.
    - Comma Separated Values (.csv) if you want to share the extract with a third party.

    Tip: You have choices when generating output from your flow. You can generate an extract file, or you can publish your data as a data source to Tableau Server or Tableau Online. For more information about generating output files, see **Create and publish data extracts and data sources** on page 137.

5. Click the **Run Flow** button ▷ to generate your output.
6. When the flow is finished running, a status dialog shows whether the flow ran successfully and the time it took to run. Click **Done** to close the dialog.



Wrap up and resources

You are a data prep rock star! You took dirty data and transformed it with ease! In no time, you cleaned and prepped your data from multiple data sets and turned it into a sleek, clean data set that you can now work with in Tableau Desktop to do your analysis.

Want more practice? Try replicating the rest of the sample flow for Superstore using the other data files found here:

- (Windows) `C:\Program Files\Tableau\Tableau Prep <version>\help\Samples\en_US\Superstore Files`
- (Mac) `/Applications/Tableau Prep <version>.app/Contents/help/Samples/en_US/Superstore Files`

Want more training? Check out the new training videos for Tableau Prep or take an in-person training course.

Want more information about the topics we covered? Check out the other topics in the Tableau Prep online help.

About Tableau Prep

Tableau Prep is a new tool in the Tableau product suite designed to make preparing your data easy and intuitive. Use Tableau Prep to combine, shape, and clean your data for analysis in Tableau.

In this article

Using Tableau Prep

Start by connecting to your data from a variety of files, servers, or Tableau extracts. Connect to and combine data from multiple data sources. Drag and drop or double-click to bring your tables into the flow pane, and then use familiar operations such as filter, split, rename, pivot, join, and union to clean and shape your data.

Each step in the process is represented visually in a flow chart that you create and control. Tableau Prep tracks each operation so that you can check your work and make changes at any point in the flow.

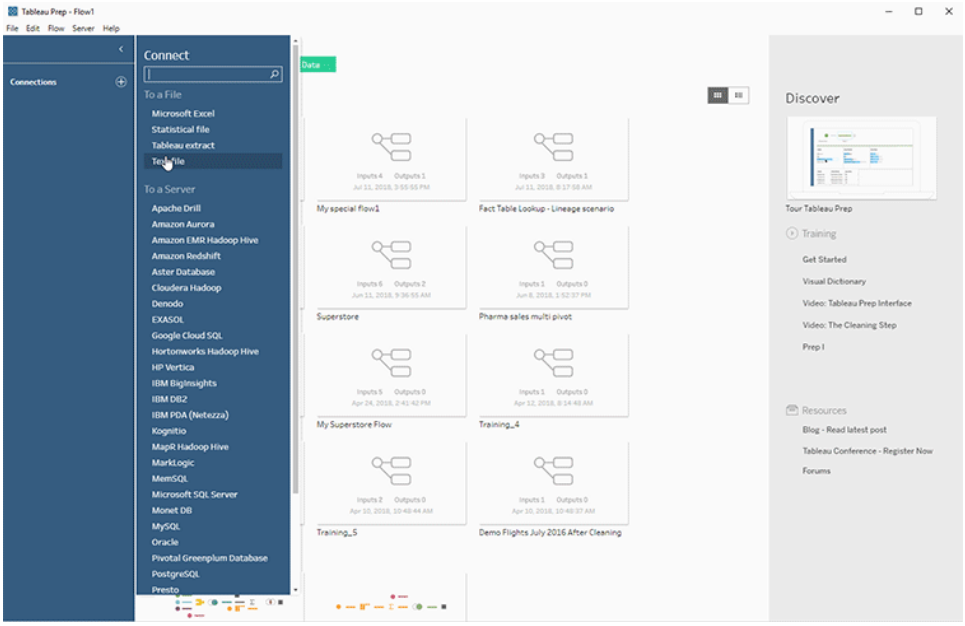
When you are finished with your flow, run it to apply the operations to the entire data set.

Tableau Prep works seamlessly with other Tableau products. At any point in your flow, you can create an extract of your data, publish your data source to Tableau Server or Tableau Online, or even open Tableau Desktop directly from within Tableau Prep to preview your data.

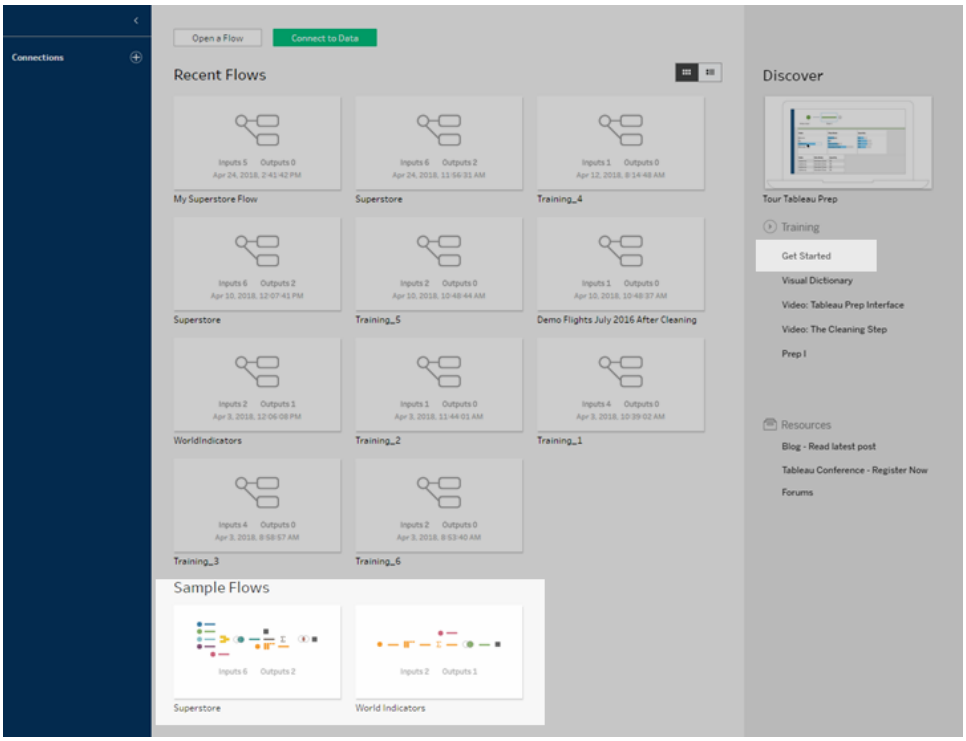
For information about installing Tableau Prep, see Install Tableau Prep in the Tableau Desktop and Tableau Prep Deployment Guide.

See Tableau Prep in action

Click the image to replay it.



Ready to try it out? From the **Start** page, click on one of the sample flows to explore and experiment with the steps, try the **Get Started with Tableau Prep** on page 25 hands-on tutorial to learn how to create a flow or try stepping through one of our **Day in the Life Scenarios** on page 149 using Tableau Prep.



Note: You can find the sample data files used in the flows in these locations:

- (Windows) `C:\Program Files\Tableau\Tableau Prep <version>\help\Samples\en_US`
- (Mac) `/Applications/Tableau Prep <version>.app/Contents/help/Samples/en_US`

To learn more about Tableau Prep and the different features and functions it offers, review the topics in this guide.

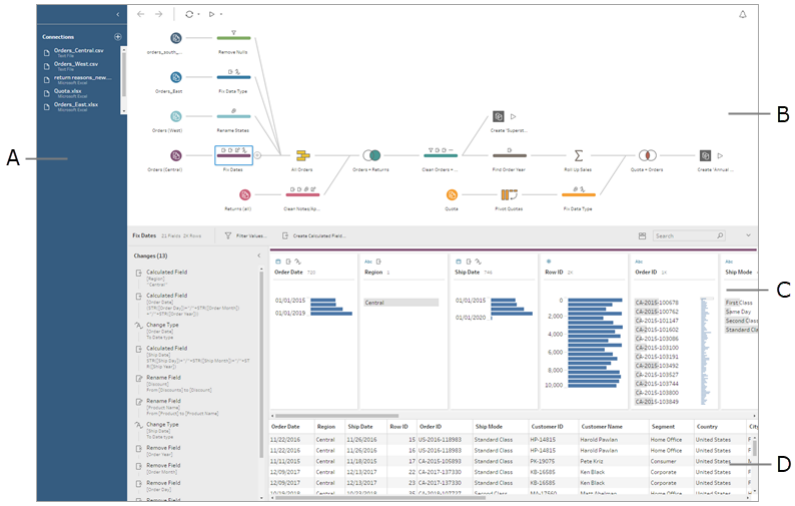
A tour of the Tableau Prep workspace

The Tableau Prep workspace consists of the **Connections pane** (A) and three coordinated areas that help you interact with and explore your data:

- **Flow pane** (B): A visual representation of your operation steps as you prepare your data.
- **Profile pane** (C): A summary of each field in your data sample. See the shape of your

data and quickly find outliers and nulls.

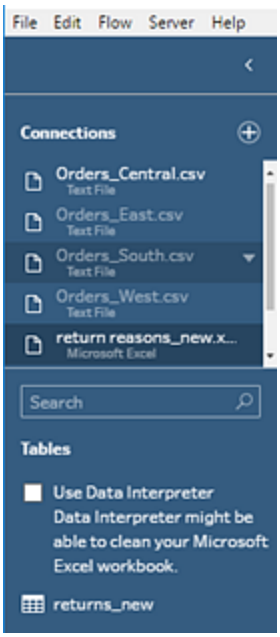
- **Data grid** (D): The row level detail for your data.



After you connect to your data and begin building your flow, you add steps in the **Flow** pane. These steps function as a lens into the structure of your data, as well as a summary of operations that is applied to your data. Each step represents a different category of operations that you define, all as part of your flow.

Connections pane

On the left side of the workspace is the **Connections** pane, which shows the databases and files you are connected to. Add connections to one or more databases and then drag the tables you want to work with into the **Flow** pane. For more information see **Connect to Data** on page 71.

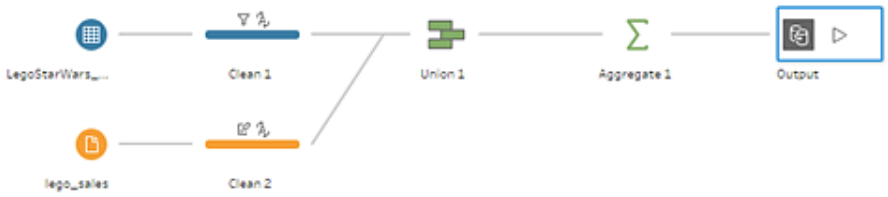


You can minimize the **Connections** pane if you need more room in your workspace.

Flow pane

At the top of the workspace is the **Flow** pane. This is your canvas. As you connect to, clean, shape, and combine your data, steps appear in the **Flow** pane and align from left to right along the top. These steps tell you what kind of operation is being applied, in what order, and how your data is affected by it. For example, the Join step shows you which join type you've applied, the join clauses, recommended join clauses, and the fields of the tables that are included in the join.

You start your flow by dragging tables into the **Flow** pane. Here you can add additional data sets, pivot your data, union or join data, create aggregations, and generate output files in the form of .tde and .tds files or Hyper extract (.hyper) files that you can use in Tableau. For more information about generating output files, see **Save and Share Your Work** on page 136.



Note: If you make changes to the data while in Tableau Desktop, for example renaming fields, changing data types, and so on, these changes aren't written back to Tableau Prep.

Profile pane

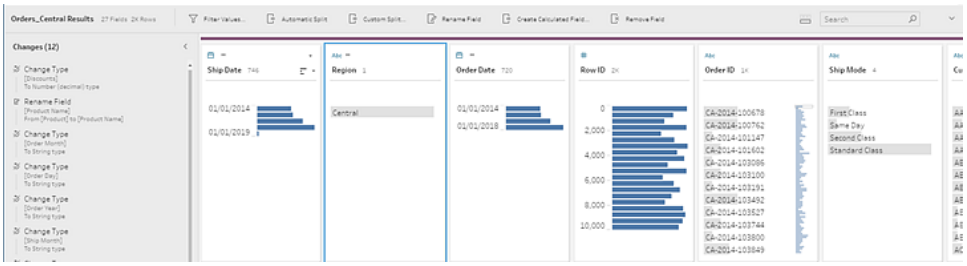
In the center of the workspace is the **Profile** pane. The **Profile** pane shows you the structure of your data at any point in the flow. The structure of your data can be represented in different ways depending on the operation you want to perform on your data or the step that you select in the **Flow** pane.

At the top of the **Profile** pane is a toolbar that shows you the cleaning operations that you can perform for each step in your flow. A drop-down menu also appears on each card in the **Profile** pane where you can select the different operations that you can perform on the data.

For example:

- Search, sort, and split fields
- Filter, include, or exclude values
- Find and fix null values
- Rename fields
- Clean up data entry errors using group and replace or quick cleaning operations
- Use automatic data parse to change data types
- Rearrange the order of your field columns by dragging and dropping them where you want them

Tableau Prep keeps track of any changes you make, in the order you make them, so you can always go back and review or edit those changes if needed. Use drag and drop to re-order those operations to experiment and apply changes in a different order.

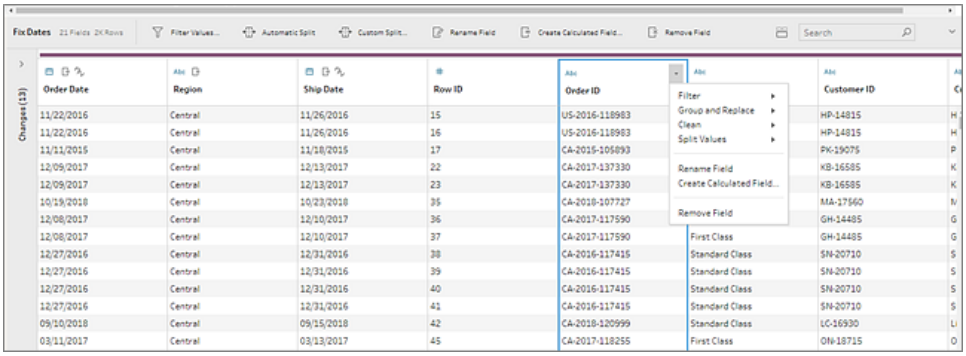


Click the arrow on the upper right of the pane to expand and collapse the **Changes** pane for more room to work with the data in the **Profile** pane.

Data grid

At the bottom of the workspace is the **Data** grid, which shows you the row level detail in your data. The values displayed in the **Data** grid reflect the operations defined in the **Profile** pane. You can perform the same cleaning operations here as you can in the Profile pane if you prefer to work at a more detailed level.

Click the **Collapse Profiles** icon on the toolbar to collapse (and expand) the **Profile** pane to see your options.



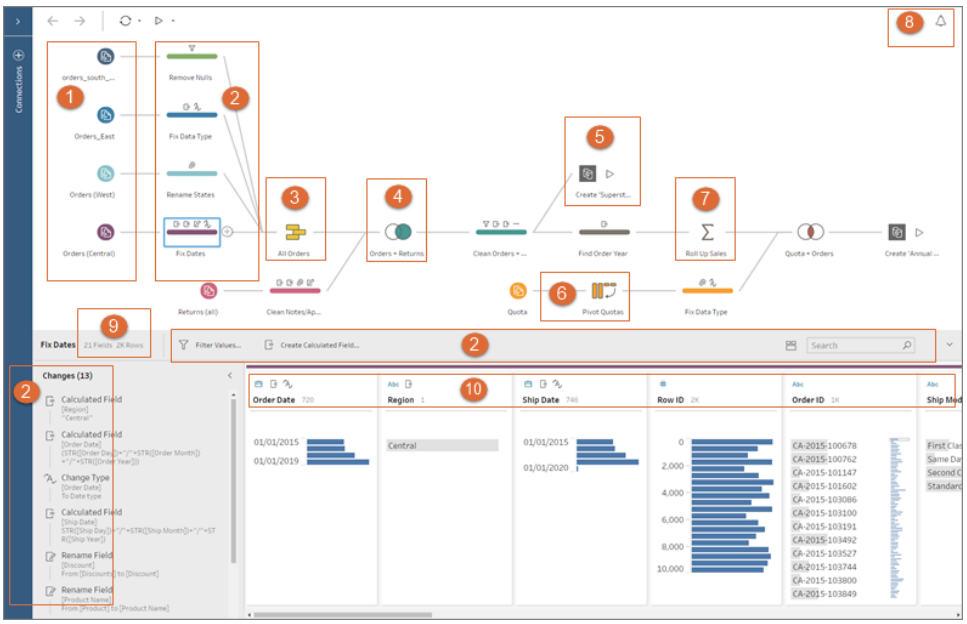
How Tableau Prep stores your data

When you connect Tableau Prep to your data and create a flow, it stores the frequently used data in a .hyper file. For large data sets, this might be a sample of the data. Any stored data is saved under a secure temporary file directory in a file named PrepXXXXX, where XXXXX represents a universally unique identifier (UUID). After you save the flow, the file is deleted.

Tableau Prep also saves data in the Tableau flow (.tfl) file to support the following operations (which can capture entered data values):

- Custom SQL used in Input steps
- Filtering (on data entry)
- Group and replace (on data entry)
- Calculations

Tableau Prep Visual Dictionary



1	**Input Step** Start your flow by dragging data to the Flow pane to create an Input step. The icon shows you the type of data source. Data Source Data Source with Wildcard Union Excel Excel with Wildcard Union CSV CSV with Wildcard Union Tableau Extract

2

Clean Step, Changes Pane, and Toolbar

Easily keep track of the changes you make to your data. Find these icons on the Clean steps in the Flow pane, in the Changes pane, and on the toolbar.

- Calculated Field
- Change Data Type
- Edit Value
- Exclude Values
- Filter Values
- Group Values
- Keep Only
- Hide Profile Pane
- Show Profile Pane
- Merge Fields
- Remove Field
- Rename Field
- Search
- Split Fields

3

Union Step

Combine up to ten sources of data with similar fields in a single Union step.

- Union Data

4

Join Step

Connect two steps to join your data on one or more common fields. Select one of these options to choose the join type.

- Full Anti Join
- Inner Join
- Left Inner Join
- Left Outer Join
- Full Outer Join
- Right Inner Join
- Right Outer Join

5	**Output Step** Add an Output step and run your flow to apply the changes to the complete data set and generate output files.  CSV Published Data Source Tableau Data Extract Run Flow
6	**Pivot Step** Add a Pivot step to change columns to rows.  Pivot Data
7	**Aggregate Step** Add an Aggregate step to group and aggregate your data, which changes the level of detail of your data.  Aggregate Data

8

Notification

If there's a problem with your flow or something you need to know, check notifications. Errors include a **Go to Error** link to help you quickly find the problem.

- No Notifications
- Notification Alert
- Error in the Step

9

Profile Pane

See the exact row count of your data and know when your data is sampled.

Sampled — Only shows when data is sampled

Hover over row summary to see exact row count

10

Profile Card

Identify the data type and see the options available to apply to your data when you select a field in the Profile pane.

- Calculated Field
- Change Data Type
- Edit Value
- Exclude Values
- Filter Values
- Group Values
- Keep Only
- Merge Fields
- Remove Field
- Rename Field
- Search
- Split Fields
- Boolean Data Type
- Date Data Type
- Date Time Data Type
- Numeric Data Type
- Text Data Type

Connect to Data

To use Tableau Prep to clean and prepare your data, start a new flow by connecting to your data, just like in Tableau Desktop. You can also open an existing flow and pick up where you left off.

You can see and access your most recent flows right on the Start page, so it's easy to find your work in progress. After you connect to your data, use the different options in the Input step to identify the data that you want to work with in your flow.

In this article

Start or open a flow

Tableau Prep supports connections to popular types of data as well as Tableau data extracts (.tde or .hyper). New connectors are added with each new version, so check the **Connections** pane to see if your connector is available.

> **Note**: If you open a flow in a version where the connector isn't supported, the flow may open but might have errors or won't run unless the data connections are removed.

You can also use custom SQL queries to connect to data just like you can in Tableau Desktop today. For more information, see Connect to a Custom SQL Query in the Tableau Desktop and Web Authoring Help.

To check whether you can connect to your data, open Tableau Prep and click the **Add connection** button to see if a connector for your data is listed in the left pane under **Connect**.

Note: Some connectors might require you to download and install a driver before you can connect to your data. See the Driver Download page on the Tableau website to get driver download links and installation instructions.

Working with Tableau data extracts

When you connect to a Tableau data extract, Tableau Prep unpackages the extract and hyper expands, using a lot of temp space as it applies your flow operations to the resulting raw data.

This means you may need more RAM and disk space to accommodate a file that size. For example, an extract file with 18 columns and 1.2 million rows that is 360MB (8.5 GB uncompressed) may need up to 32GB RAM, 16-core, and 500GB of disk space available to support the file when it is unzipped.

Start a new flow

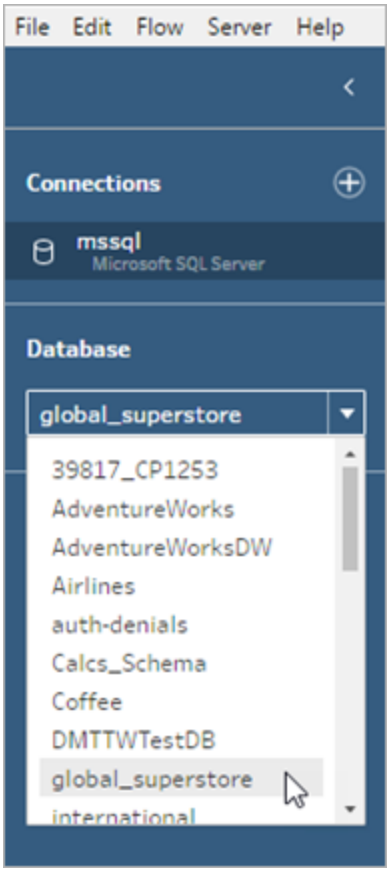
1. Open Tableau Prep and click the **Add connection** button.
2. From the list of connectors, select the file type or server that hosts your data. If prompted, enter the information needed to sign in and access your data.

3. From the **Connections** pane, do one of the following:

    - If you connected to a file, double-click or drag a table to the **Flow** pane to start your flow.

For single tables, Tableau Prep automatically creates an Input step for you in the **Flow** pane when you add data to your flow.

- If you connected to a database, select a database or schema, and then double-click or drag a table to the **Flow** pane to start your flow.



Open an existing flow

To open an existing flow, on the **Start** page do one of the following :

- Under **Recent Flows**, select a flow.
- Click **Open a Flow** to navigate to your flow file and open it.

> **Note:** To go back to the Start page, select **File** > **New**.

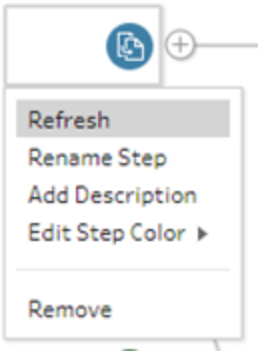
Refresh data in the Input step

If data changes in your input files or tables after you begin working with your flow, you can refresh the Input step to bring in the new data.

Use one of the following options:

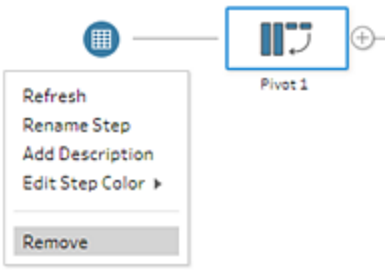
- In the flow pane on the top menu, click the **Refresh** button to refresh all Input steps. To refresh a single Input step, click the drop-down arrow next to the refresh button and select the Input step from the list.

- In the flow pane, right-click the Input step you want to refresh and select **Refresh** from the menu.



- Remove and re-add the Input step to the flow.

    1. In the flow pane, right-click the Input step you want to refresh and select **Remove** from the menu.



    This will temporarily put your flow in an error state.

2. Connect to the updated file again.
3. Drag the table to the flow pane on top of the second step in the flow where you want to add the Input step. Drop it on the **Add** option to reconnect it to the flow.

Use Data Interpreter to clean your files

When working with Microsoft Excel files, you can use Data Interpreter to detect sub-tables in your data as well as remove extraneous information to help prepare your data for analysis. When you turn on Data Interpreter, it detects these sub-tables and lists them as new tables in the **Tables** section of the **Connections** pane.

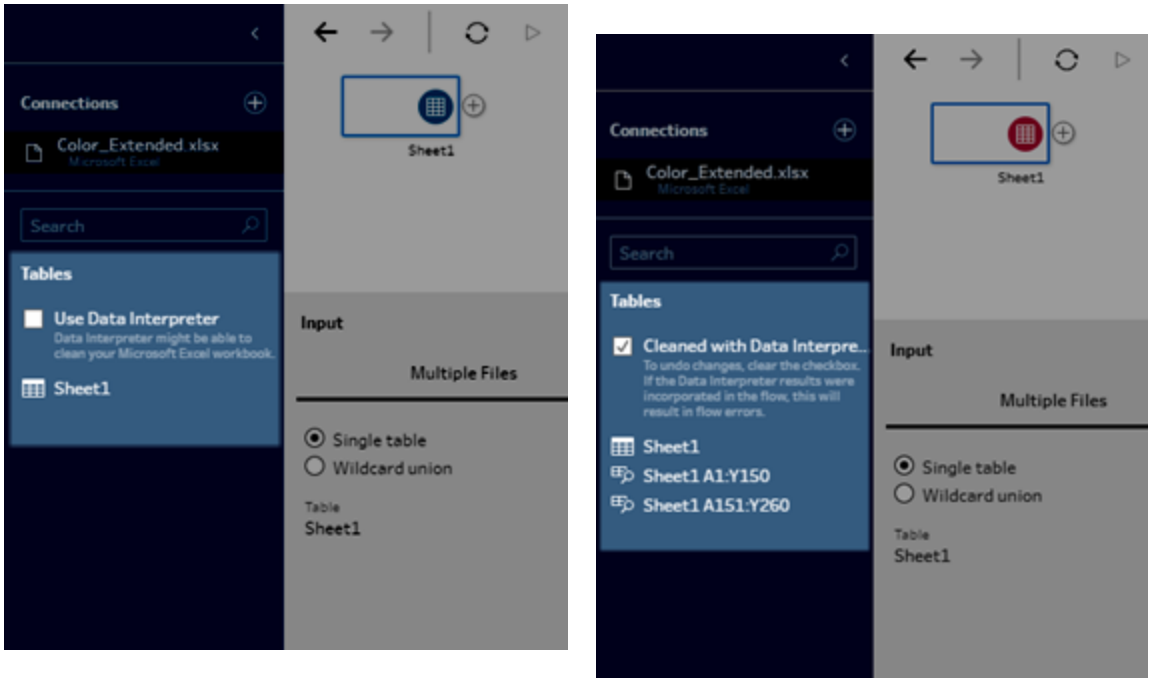
You can then drag them into the **Flow** pane. If you are using Tableau Prep version 2018.1.2 or later, you can select the **Wildcard union** option in the **Multiple Files** tab to include all found sub-tables in your flow. For more information about using Wildcard union in the Input step see **Union files in the Input step** on the next page.

If you turn Data Interpreter off, these tables are removed from the **Connections** pane. If these tables are already used in the flow, this will result in flow errors from the missing data.

> **Note**: Currently, Data Interpreter only detects sub-tables in your Excel spreadsheets.

The example below shows the results of using Data Interpreter on an Excel spreadsheet in the **Connections** pane. Data Interpreter detected two additional sub-tables.

Before Data Interpreter	After Data Interpreter



To use Data Interpreter, complete the following steps:

1. Select **Connect to Data** then select **Microsoft Excel**.
2. Select your file and click **Open**.
3. Select the **Use Data Interpreter** check box.
4. Drag the new table to the **Flow** pane to include it in your flow. To remove the old table, right-click the Input step for the old table and select **Remove**.

Union files in the Input step

When working with multiple files from a single data source, you can search for files using a wildcard search and then union the data to include all of this data in the Input step. To use this type of union, the files must be in the same parent or child directory. If you add or remove files after you create the union you can refresh the Input step to update your flow with the new or changed data.

If you use Data Interpreter to clean your Excel file and are using Tableau Prep version 2018.1.2 or later, you can use the wildcard search to union and add any sub-tables that Data Interpreter found.

Note: Editing the Input connection for a flow in Tableau Prep 2018.1.1 that includes this type of input union can result in errors in the flow.

If you need to union data from different data sources, you can do that using a Union step. For more information about creating Union steps, see **Join or Union Data** on page 124.

Note: Currently, this feature applies only to Excel and .csv (text) files.

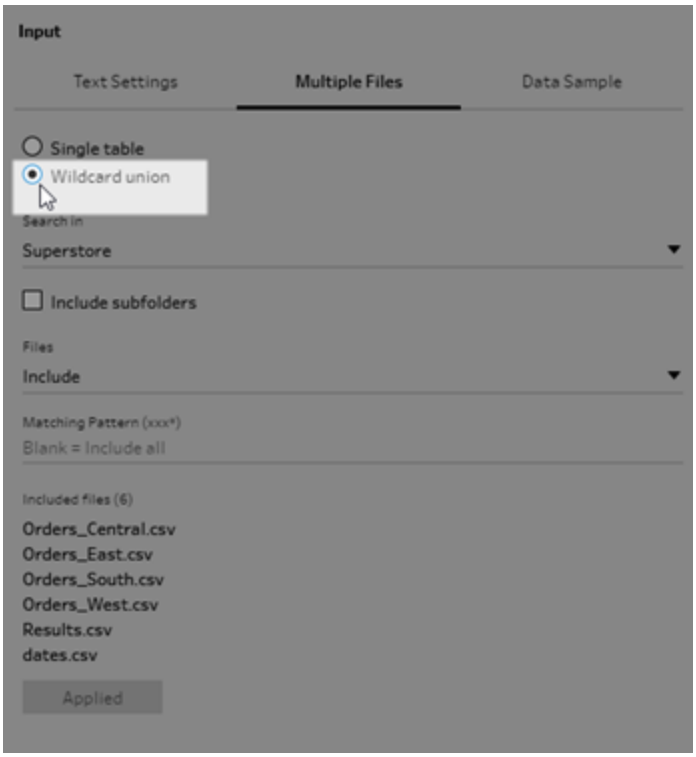
By default, Tableau Prep unions all .csv files in the same directory as the .csv file you connected to or all the sheets in the Excel file you connected to.

If you want to change the default union, use the following criteria to find the files or sheets you want to include in the union:

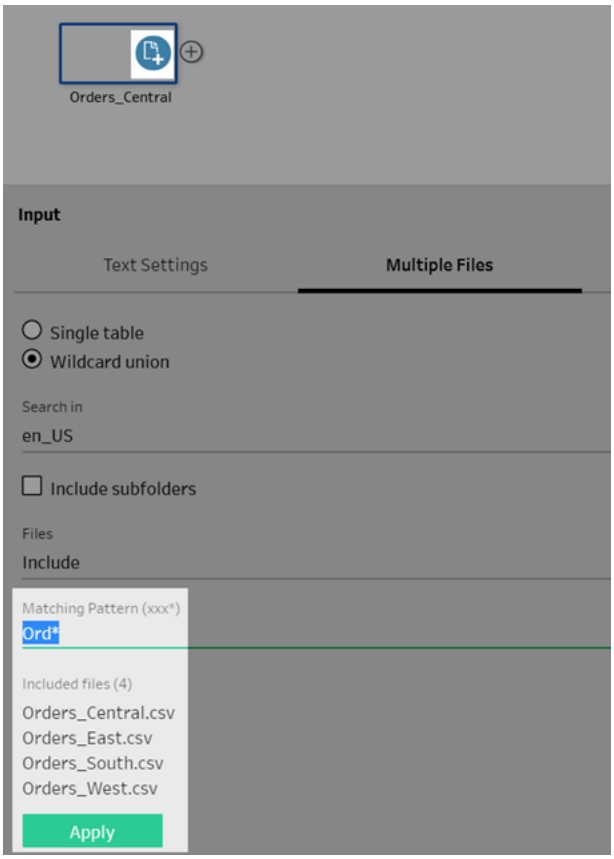
- **Search in**: Select the directory to use to search for files. Select the **Include subfolders** check box to include files in the sub-directory of the parent folder.
- **Files**: Select whether to include or exclude the files that match the wildcard search criteria.
- **Matching Pattern (xxx*)**: Enter a wildcard search pattern to find files that have those characters in the file name. For example, if you enter ord* all files that include the file name are returned. Leave this field blank to include all of the files in the specified directory.

To use wildcard search to union files:

1. Click the **Add connection** button and under **Connect**, click **Text File** for .csv files or **Microsoft Excel** for Excel files, and then select a file to open.
2. In the **Input** pane, select the **Multiple Files** tab, and then select **Wildcard union**.

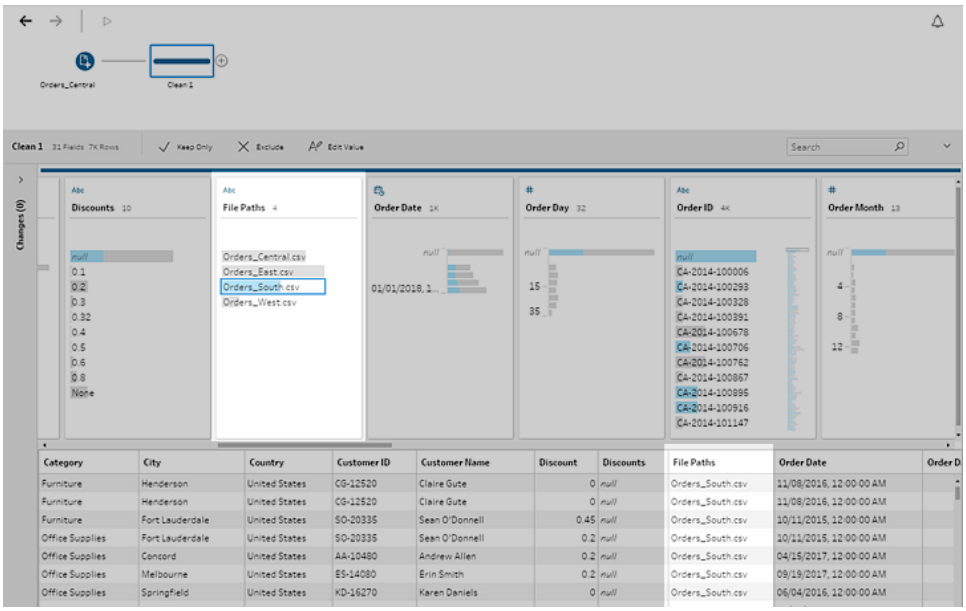


The example below shows a wildcard union using a matching pattern. The plus sign on the file icon on the Orders_Central Input step in the **Flow** pane indicates that this step includes a wildcard union. The files in the union are listed under **Included files**.



3. Use the search, file and matching pattern options to find the files that you want to union.
4. Click **Apply** to union the files.

When you add a new step to the flow, you can see all the files added to the data set in the **File Paths** field in the Profile pane, that shows which file the data came from. This field is added automatically.



Merge fields after the Input step

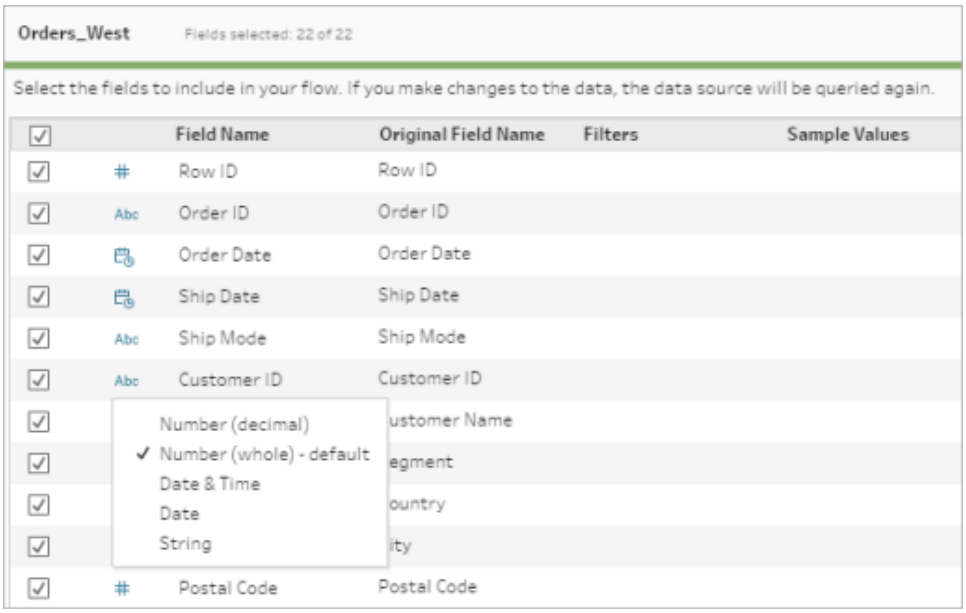
After you create a wildcard union, you might want to merge fields. You can do this in any subsequent step, except for the following steps: Join, Aggregate, Input, or Output. For more information, see **Additional merge field options** on page 135.

Configure your data set

To determine how much of your data set to include in the flow, you can configure your data set. When you connect to your data or drag tables into the **Flow** pane, an Input step is automatically added to the flow. This is always the first step in your flow. You can right-click the Input step to rename or remove it. If you're connected to an Excel or text file, you can also refresh the data from the Input step.

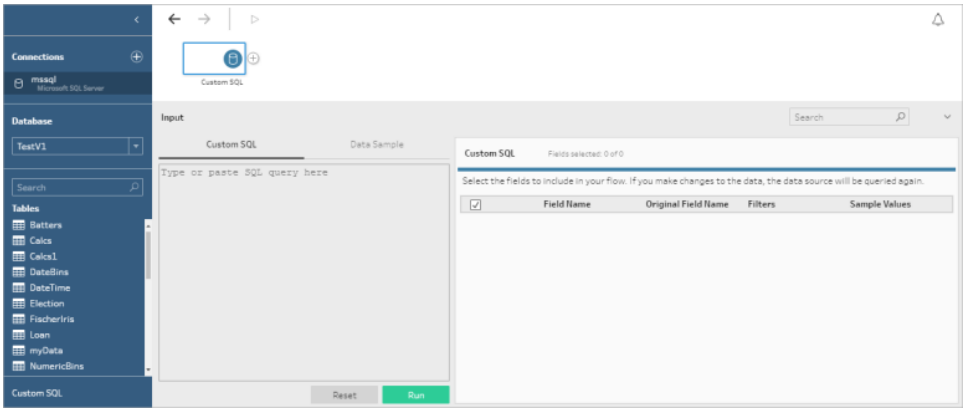
In the Input step, you can see the metadata profile for your data set. Here you can search for fields, see sample values, and perform actions to reduce the size of your data set, such as selecting the fields to include, selecting the data sample to work with, or applying filters to selected fields or rows.

You can also configure the field properties by changing the data type for fields or renaming the field names. You can configure the text settings for text files.



Connect to a custom SQL query

If your database supports using custom SQL, you will see **Custom SQL** displayed near the bottom of the **Connections** pane. Double-click **Custom SQL** to open the **Custom SQL** tab where you can enter queries to preselect data and use source-specific operations. After the query retrieves the data set, you can select the fields to include, apply filters, or change the data type before adding the data to your flow.



For more information about using custom SQL, see Connect to a Custom SQL Query in the Tableau Desktop and Web Authoring Help.

Select fields to include in the flow

The **Input** pane shows you a list of fields in your data set. You can use the **Search** field to find fields in the list, and then use the check boxes to select the fields to include. To include or exclude all fields from the flow, toggle the check box at the top left of the grid.

Configure field properties

When you work with text files, you see a **Text Settings** tab where you can edit your connection and configure text properties, such as the field separator for text files. You can also edit the file connection in the Connections pane.

When you work with text or Excel files, you can correct data types that have been inferred incorrectly before you even start your flow. Data types can always be changed in subsequent steps in the **Profile** pane after you start your flow.

Configure text settings in text files

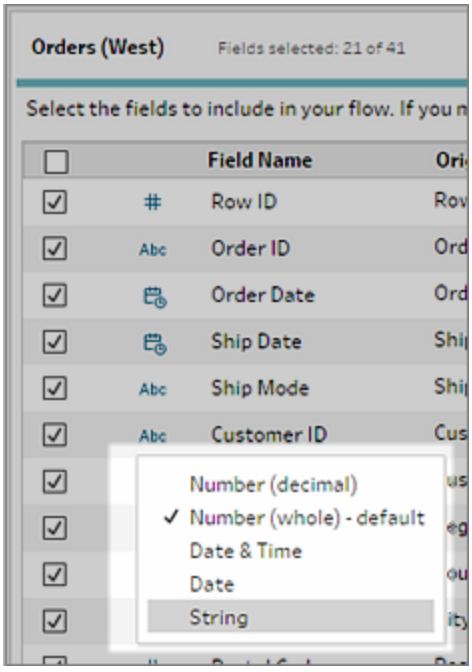
To change the settings used to parse text files, select from the following options:

- **First line contains header** (default): Select this option to use the first row as the field labels.
- **Generate field names automatically**: Select this option if you want Tableau Prep to auto-generate the field headers. The field naming convention follows the same model as Tableau Desktop. For example **F1**, **F2**, and so on.
- **Field Separator**: Select a character from the list to use to separate the columns. Select **Other** to enter a custom character.
- **Text Qualifier**: Select the character that encloses the values in the file.
- **Character Set**: Select the character set that describes the text file encoding.
- **Locale**: Select the locale to use to parse the file. This setting indicates which decimal and thousand separator to use.

Change data types

To change the data type for a field, do the following:

1. Click the data type for the field.
2. Select the new data type from the context menu.



Change field names

To change the name of a field, in the **Field Name** column, select the name, and then type the new name in the field.

Set your data sample size

By default, Tableau Prep limits the data included in the flow to a representative sample of your data set to maintain peak performance. The data sample is determined by calculating the optimal number of rows based on the total number of fields in the data set and the data types for those fields. Tableau Prep then retrieves the top number of rows for the calculated amount as quickly as possible.

The resulting data sample may include all the rows you need, or it may not, depending on how the sample was calculated and returned. If you don't see the data that you expect, you can change the data sample settings to run the query again.

Note: If your data is sampled, a **Sampled** Sampled indicator shows in the **Profile** pane and persists for every step you add. Any changes you make apply to the sample you are working with in the flow. All changes apply to your entire data set when you run the flow.

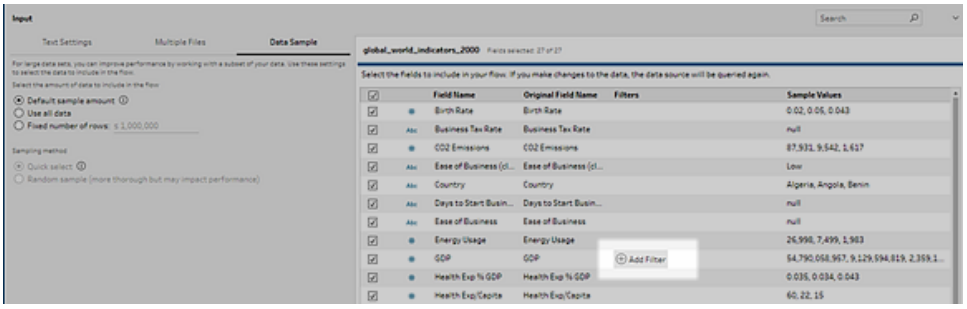
To change your data sample settings, select an Input step, then on the **Data Sample** tab select from the following options:

- **Default sample amount** (default): Tableau Prep calculates the total number of rows to return.
- **Use all data**: Retrieve all rows in your data set regardless of size. This can impact performance or cause Tableau Prep to time out.
- **Fixed number of rows**: Select the number of rows to return from the data set. The recommended number of rows is 1 million or less. Setting the number of rows to more than 1 million can impact performance.
- **Quick select** (default): The database returns the number of rows requested as quickly as possible. This might be the first N number of rows or the rows that the database had cached in memory from a previous query.
- **Random sample**: The database returns the number of rows requested but looks at every row in the data set and returns a representative sample from all of the rows. This option may impact performance when the data is first retrieved.

Apply filters to fields in the Input step

To filter a field, do the following:

1. In the **Filters** column, next to the **Field Name** that you want to filter, click **Add Filter**.



2. Select **Calculation**, then enter your filter criteria in the calculation editor.

Additional filtering capabilities are available in other steps in the flow. For more information, see **Filter values** on page 118.

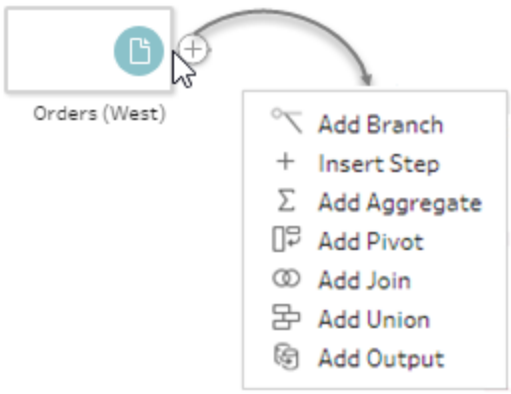
Clean and Shape Data

Clean and shape data to make it easier to combine with other data or to make it simpler for other people to understand.

In this article

Build your flow

After you connect to the data that you want to include in your flow, begin cleaning and shaping your data by adding new steps to the flow. Hover over a step until the plus ⊕ icon appears, and then click the icon to display options.



Use these options to build your flow:

- **Add Branch**: Split your flow into different branches.
- **Insert Step**: Add a step to perform cleaning operations.
- **Add Aggregate**: Select the step that includes the data you want to aggregate or group.
- **Add Pivot**: Select the step that includes the data you want to pivot.
- **Add Join**: Select the step that you want to join data with. As an alternative, you can drag-and-drop to join files. In the following example, we're dragging the Orders_Central Input step and dropping it on **Join**:

For more information about creating a join, see **Join or Union Data** on page 124.

- **Add Union**: Select the step that you want to union data with. As an alternative, you can drag-and-drop to union files. For more information about creating a union, see **Join or Union Data** on page 124.
- **Add Output**: Select this option to save the output to an extract file (.tde or .hyper) or a .csv file, or to publish the output as a data source to a server.

Change the color scheme

Tableau Prep assigns each step in your flow a color by default. This color scheme is applied throughout the flow to help you keep track of your data throughout the flow as you apply cleaning steps, join, union or aggregate the data so you know which files are impacted by your operations.

To select a different color scheme for your steps do the following:

1. Select one or more steps.
2. Right-click on a selected step and select **Edit Step Color**.
3. Click on a color in the color palette to apply it.

To reset the step color back to the default color, do one the following:

- Click **Undo** from the top menu.
- Cntrl+Z or Command-Shift-Z (Mac).
- Select the steps you changed, right-click on a selected step and select **Edit Step Color**, then select **Reset Step Color** from the bottom of the color palette.

Add a description

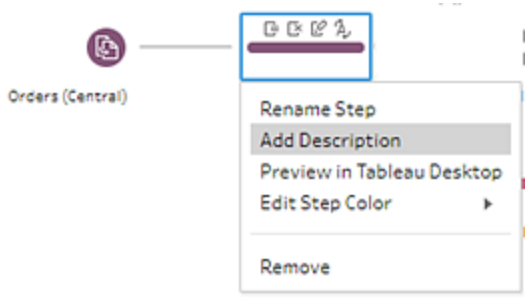
As you build your flow and perform various cleaning operations, you might want to add a description to help others who might later look at or work with your flow better understand your steps.

You can add a description to any individual step in your flow directly on the Flow pane. The description can be up to 200 characters long.

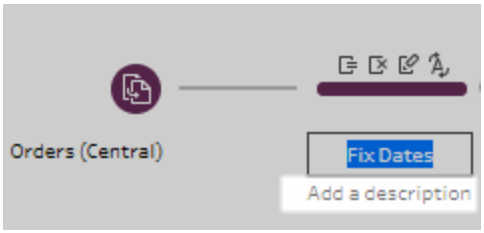
When you add a description, a message icon is added underneath the step. Click the icon to show or hide the description text in the Flow pane.

To add a description to a step do the following:

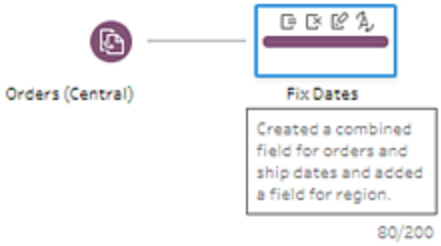
1. In the Flow pane, select a step.
2. Do one of the following:
    - Right-click the step and select **Add Description** from the menu.



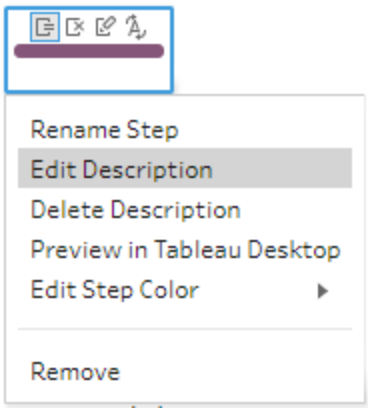
- Double-click in the name field for the step, then click on **Add a description**.



3. Type your description in the text box.



4. Click outside the text box or press Enter to apply your changes. By default, the description displays underneath the step. To hide the description click the message icon.
5. To edit or delete the description, right-click on the step or description and select **Edit Description** or **Delete Description** from the menu.



Remove steps from the flow

At any point in the flow, you can remove steps or the flow lines between steps.

- To remove a step or flow line, select the step or line you want to remove, right-click the element, and then select **Remove**.
- To remove multiple steps or flow lines, do one of the following:
    - Use your mouse to drag and select a whole section of the flow. Then right-click on one of the selected steps and select **Remove**
    - Press **Ctrl+A** (**Cmd+A** on Mac) to select all elements in the flow, or press **Ctrl+click** (**Cmd+Click** on Mac) to select specific elements, and then press the **Delete** key.

Pivot your data

Sometimes analyzing data from a spreadsheet or crosstab format can be difficult in Tableau. Tableau prefers data to be "tall" instead of "wide", which means that you often have to pivot your data from columns to rows so that Tableau can evaluate it properly.

With Tableau Prep, you can pivot on one or more groups of fields to get the results you want from your data. Simply select the fields that you want to work with, pivot the data from columns to rows, and then interact directly with the results to get your data looking just the way you want it.

> **Note**: Pivoting on multiple groups of fields is not supported in Tableau Prep 2018.1.1. Editing a pivot that includes pivoted columns on multiple field groups in Tableau Prep version 2018.1.1 can result in errors or unexpected results.

You can also use Tableau Prep's smart default naming feature to automatically rename your pivoted fields and values.

To pivot your data:

1. Connect to your data source.
2. Drag the table that you want to pivot to the **Flow** pane.
3. Click the plus ⊕ icon, and select **Add Pivot** from the context menu.

4. (Optional) Search for fields to pivot.
5. Select one or more fields from the left pane, and drag them to the **Pivot1 Values** column in the **Pivoted Fields** pane.
6. (Optional) In the **Pivoted Fields** pane, click the plus ⊕ icon to add more columns to pivot on, then repeat the previous step to select more fields to pivot.

   > **Note**: You must select the same number of fields that you selected in Step 5. For example if you selected 3 fields to initially pivot on, then each subsequent column that you pivot on must also contain 3 fields.

7. (Optional) Select the **Automatically rename pivoted fields and values** check box to enable Tableau Prep to rename the new pivoted fields using common values in the data. If no common values are found, the default name is used.

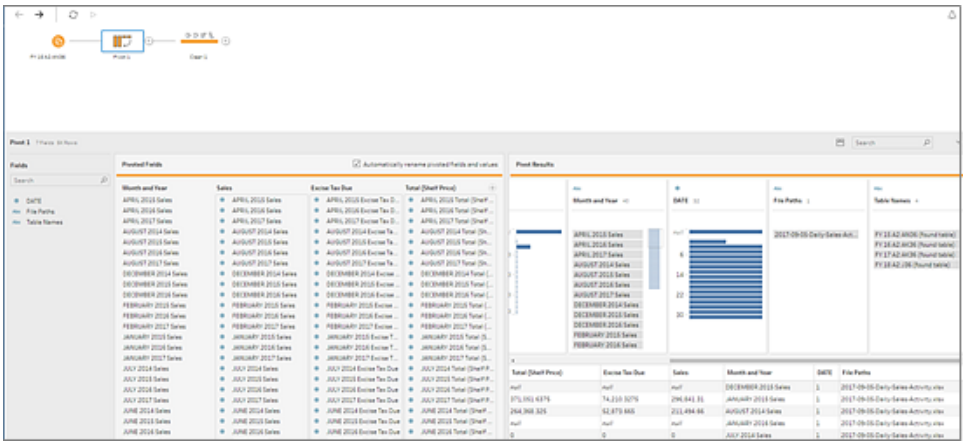
   Your results appear immediately in both the **Pivot Results** pane and the data grid.
8. If you didn't enable the default naming option in the previous step, edit the names of the fields. You can also edit the names of the original fields in this pane to best describe the data.
9. Rename the new Pivot step to keep track of your changes. For example "Pivot months".

Example: Pivoting on multiple fields

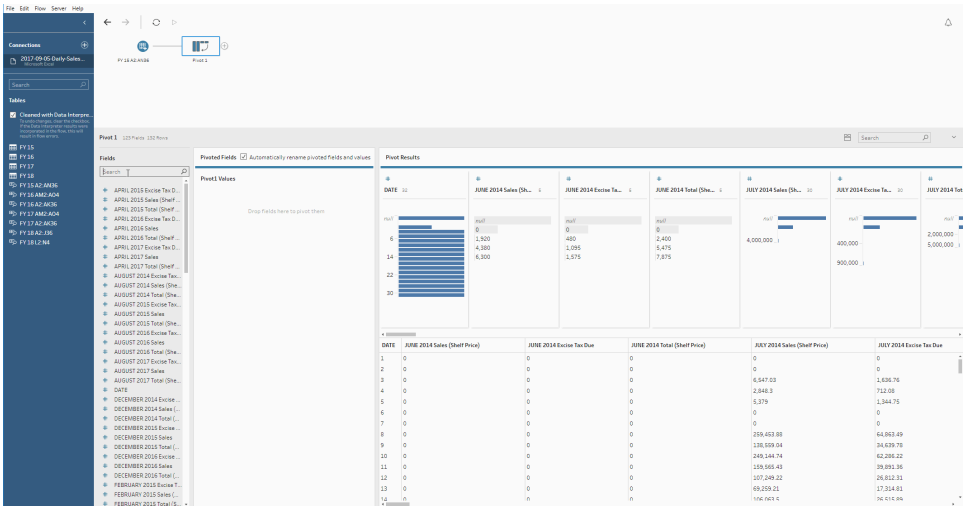
This example shows a spreadsheet for pharmaceutical sales, taxes and totals by month and year.

	A	T	U	V	W	X	Y	Z	AA	AB
2	DATE	DECEMBER 2014			JANUARY 2015			FEBRUARY 2015		
3		Sales (Shelf Price)	Excise Tax Due	Total (Shelf Price	Sales (Shelf Price)	Excise Tax Due	Total (Shelf Price	Sales (Shelf Price)	Excise Tax Due	Total (Shelf Price
4	1	$ 448,111	$ 112,028	$ 560,139	$ 296,841	$ 74,210	$ 371,052	$ 212,919	$ 53,230	$ 266,148
5	2	$ 425,472	$ 106,368	$ 531,840	$ 754,061	$ 188,515	$ 942,577	$ 449,897	$ 112,474	$ 562,371
6	3	$ 435,525	$ 108,881	$ 544,406	$ 482,497	$ 120,624	$ 603,121	$ 627,711	$ 156,928	$ 784,639
7	4	$ 634,765	$ 158,691	$ 793,456	$ 332,228	$ 83,057	$ 415,284	$ 688,263	$ 172,066	$ 860,329
8	5	$ 695,425	$ 173,856	$ 869,282	$ 601,529	$ 150,382	$ 751,912	$ 789,233	$ 197,308	$ 986,541
9	6	$ 436,720	$ 109,180	$ 545,899	$ 527,374	$ 131,843	$ 659,217	$ 867,501	$ 216,875	$ 1,084,377
10	7	$ 238,481	$ 59,620	$ 298,101	$ 560,102	$ 140,026	$ 700,128	$ 554,459	$ 138,615	$ 693,074
11	8	$ 421,422	$ 105,356	$ 526,778	$ 539,974	$ 134,993	$ 674,967	$ 448,846	$ 112,211	$ 561,057
12	9	$ 543,816	$ 135,954	$ 679,770	$ 683,408	$ 170,852	$ 854,260	$ 768,266	$ 192,067	$ 960,333
13	10	$ 616,271	$ 154,068	$ 770,339	$ 442,352	$ 110,588	$ 552,940	$ 719,637	$ 179,909	$ 899,546
14	11	$ 756,542	$ 189,135	$ 945,677	$ 288,605	$ 72,151	$ 360,756	$ 1,154,919	$ 288,730	$ 1,443,649
15	12	$ 726,270	$ 181,567	$ 907,837	$ 674,121	$ 168,530	$ 842,651	$ 1,019,936	$ 254,984	$ 1,274,921
16	13	$ 477,208	$ 119,302	$ 596,510	$ 526,451	$ 131,613	$ 658,064	$ 951,242	$ 237,811	$ 1,189,053
17	14	$ 245,898	$ 61,475	$ 307,373	$ 573,842	$ 143,461	$ 717,303	$ 798,392	$ 199,598	$ 997,991
18	15	$ 456,254	$ 114,064	$ 570,318	$ 658,952	$ 164,738	$ 823,690	$ 453,091	$ 113,273	$ 566,364

By pivoting the data you can create rows for each month and year and individual columns for sales, taxes and totals so that Tableau can more easily interpret this data for analysis.



Watch "pivot on multiple field" in action.



Apply cleaning operations

You clean data by applying cleaning operations such as filtering, adding, renaming, splitting, or removing fields. As you make changes to your data, annotations are added to the corresponding step in the **Flow** pane.

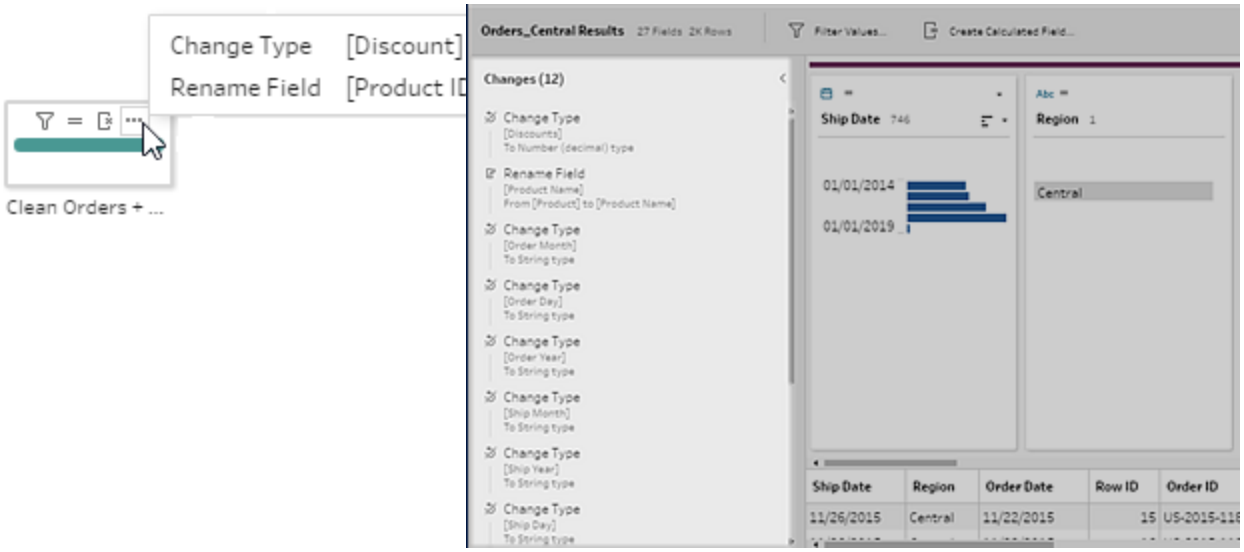
You can apply cleaning operations in either the Profile pane or the data grid. If you want to work in the data grid, you must click the **Hide profile pane** button to collapse the Profile pane to see the cleaning operations.

Note: Not all cleaning operations are available in the data grid. For example if you want to edit a value in-line, you must do this in the Profile pane.

The different types of cleaning operations are represented by icons over cleaning steps. If more than four types of operations are applied to a cleaning step, an ellipsis displays over the step. Hover over these icons to view annotations showing applied operations and the order in which they are performed.

You can also select a step and then expand the **Changes** pane in the **Profile** pane toolbar, where you can view the details for each change, edit or remove your changes, or drag changes up or down to change the order in which they're applied.

Cleaning annotation **Changes pane**



To apply cleaning operations to a field:

1. In the **Profile** pane or data grid, select the field you want to make changes to.
2. From either the drop-down arrow for the field or the toolbar, select from the following options:
    - **Filter** or **Filter Values**: You can also right-click a field value to keep only, exclude, or edit the value.
    - **Group and Replace**: Manually select values or use automatic grouping. You can also multi-select values in the Profile pane and right-click to group or ungroup

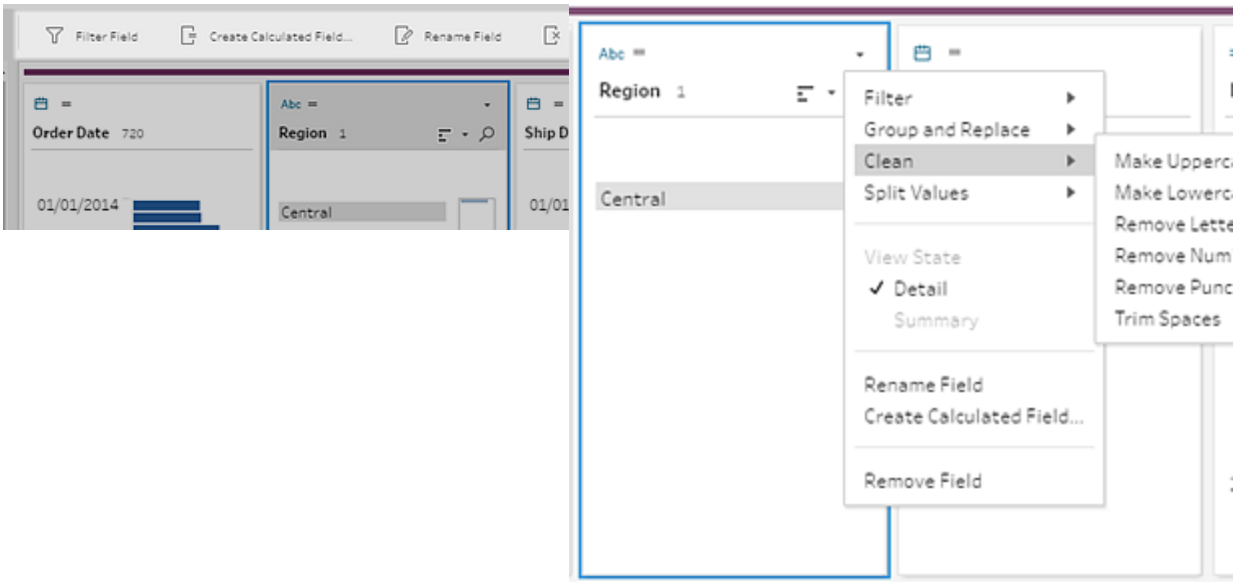
values or edit the group value.

- **Clean**: Select from a list of quick cleaning operations to apply to all values in the field.
- **Split Values**: Select either automatic or custom splits.

  > **Note**: Automatic split and custom split work the same as they do in Tableau Desktop. For more information, see Split a Field into Multiple Fields in the Tableau Desktop and Web Authoring Help.

- **Rename Field**
- **Create Calculated Field**
- **Remove Field**

Profile pane toolbar **Drop-down menu**



3. Review the results of these operations in the **Profile** pane or data grid.

Merge fields

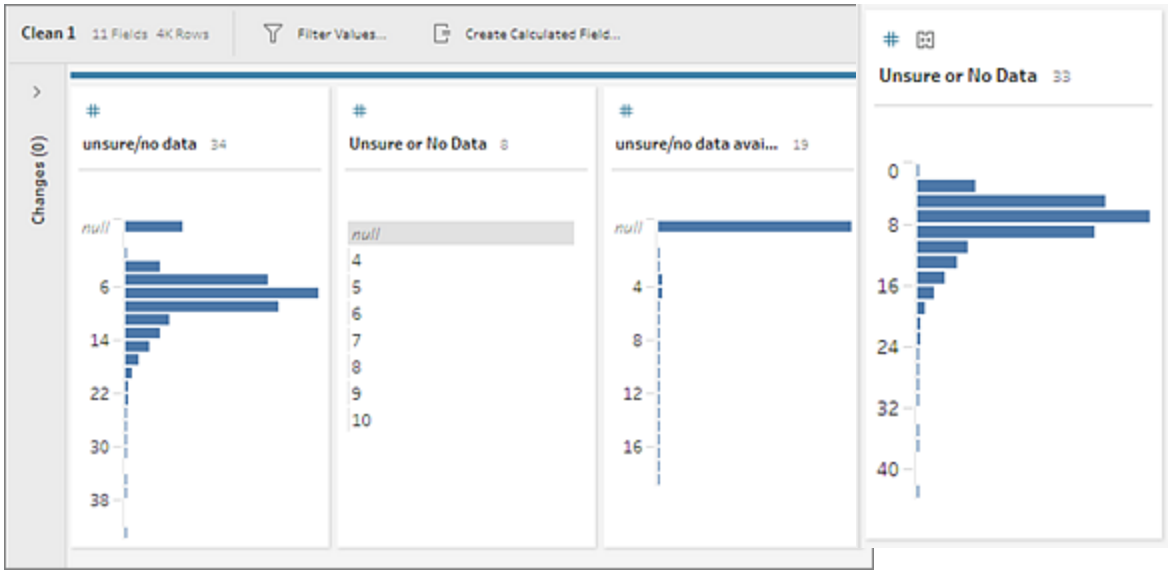
If you have fields that contain the same values but are named differently, you can easily merge them to combine them into one by dragging one field on top of the other. When you merge the

fields, the field name of the target field persists.

Example:

Wildcard union results in 3 fields with the same values

Merge 3 fields into 1



To merge fields, do one of the following:

- Drag and drop one field onto another. A **Drop to merge fields** indicator displays.
- Select multiple fields and right-click within the selection to open the context menu, and then click **Merge Fields**.
- Select multiple fields, and then click **Merge Fields** on the context-sensitive toolbar.

For information about how to fix mismatched fields as a result of a union, see **Fix fields that don't match** on page 132.

Cleaning (fixing) variations of the same value

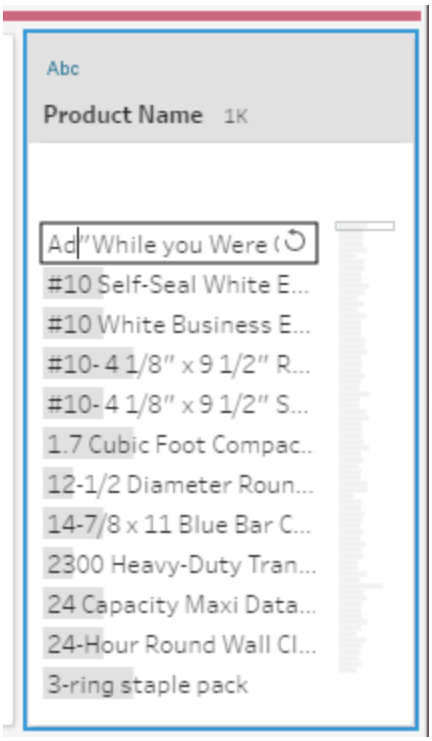
Multiple variations of the same value can prevent you from accurately summarizing your data. You can quickly and easily correct these variations using the following options in Tableau Prep.

Note: Any edits that you make to the values must be compatible with the field data type. In-line editing is only available in the Profile pane.

Edit a single value

1. In the **Profile** pane, click the value you want to edit, and enter the new value. A group icon shows next to the value.

   Alternatively, right-click a value and click **Edit Value**. The change is recorded in the **Changes** pane on the left side of the screen.



2. View the results in the **Profile** pane and data grid.

Edit multiple values

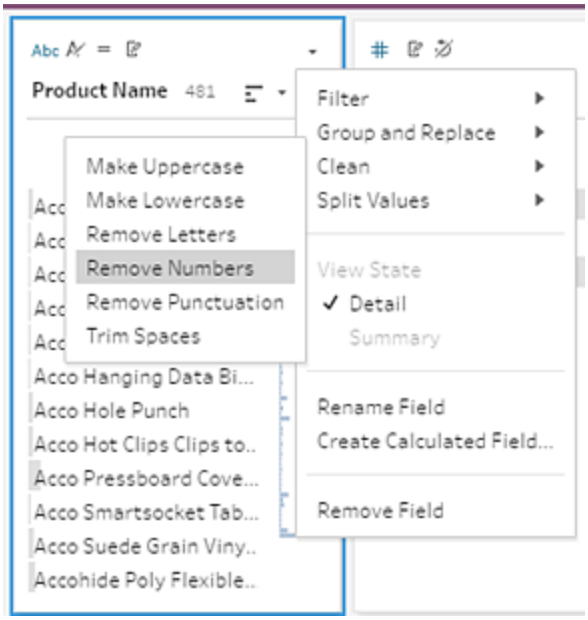
You have several options to edit multiple values at once. For example, use quick cleaning operations to remove punctuation for all values in a field, manually group values using multi-select or automatically group values together using fuzzy-match algorithms that find similar values.

Note: When you map multiple values to a single value, the original field shows a group icon next to the value, showing you which values are grouped together.

Edit multiple values using quick cleaning operations

This option applies only to text fields.

1. In the **Profile** pane or data grid, select the field you want to edit.
2. Click the drop-down arrow, select **Clean**, and then select one of the following options:
    - **Make Uppercase**: Change all values to uppercase text.
    - **Make Lowercase**: Change all values to lowercase text.
    - **Remove Letters**: Remove all letters and leave only other characters.
    - **Remove Numbers**: Remove all numbers and leave letters and other characters.
    - **Remove Punctuation**: Remove all punctuation.
    - **Trim Spaces**: Remove leading and trailing spaces.

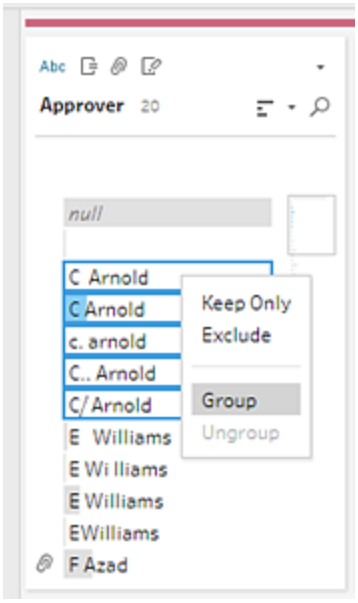


    You can stack operations to apply multiple cleaning operations to the fields. For example first select **Clean** > **Remove Numbers** and then select **Clean** > **Remove Punctuation** to remove all numbers and punctuation from the field values.

3. To undo your changes, click the **Undo** arrow at the top of the **Flow** pane, or remove the change from the change list.

Group and edit multiple values inline

1. In the **Profile** pane, select the field you want to edit.
2. Press Ctrl or Command (on Mac), and select the values that you want to group.
3. Right-click, and select **Group** from the context menu. The value in the selection that you right-click becomes the default name for the new group but you can edit this in-line.



4. To edit the group name, select the grouped field and edit the value or right-click or Ctrl-click (Mac) on the grouped field and select **Edit Value** from the context menu.
5. To ungroup the grouped field values, right-click on the grouped field and select **Ungroup** from the context menu.

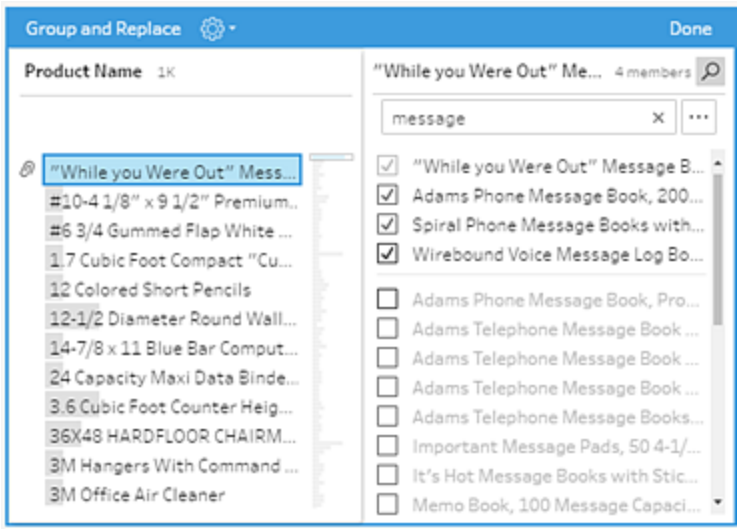
Edit multiple values manually using Group and Replace

Use **Group and Replace** to map the value of a field from one value to another or manually select multiple values to group them. You can even add new values to set up mapping relationships to organize your data.

For example, let's say you have three values in a field: My Company, My Company Incorporated, and My Company Inc. All these values represent the same company, My Company. You can use **Group and Replace** to map the values My Company Incorporated and My Company Inc to My Company, so that all three values appear as My Company in the field.

Map multiple values to a single selected field

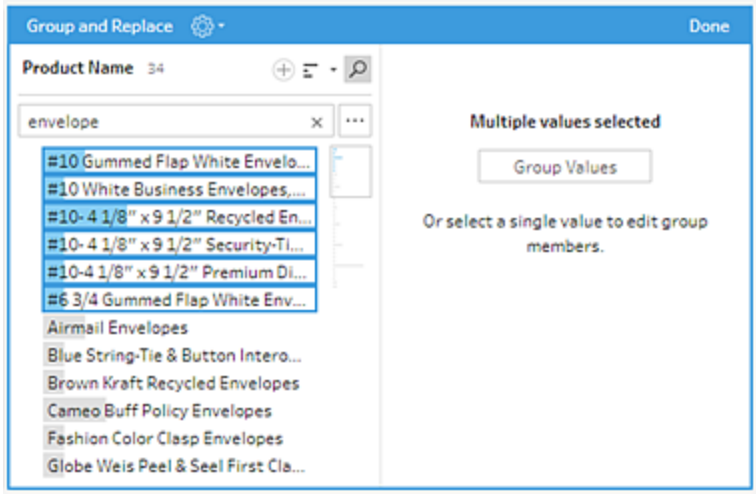
1. In the **Profile** pane, select the field you want to edit.
2. Click the drop-down arrow, and select **Group and Replace** > **Manual Selection** from the context menu.
3. In the left pane of the **Group and Replace** editor, select the field value that you want to use as the grouping value. This value now shows at the top of the right pane.
4. In the lower section of the right pane in the **Group and Replace** editor, select the values you want to add to the group.



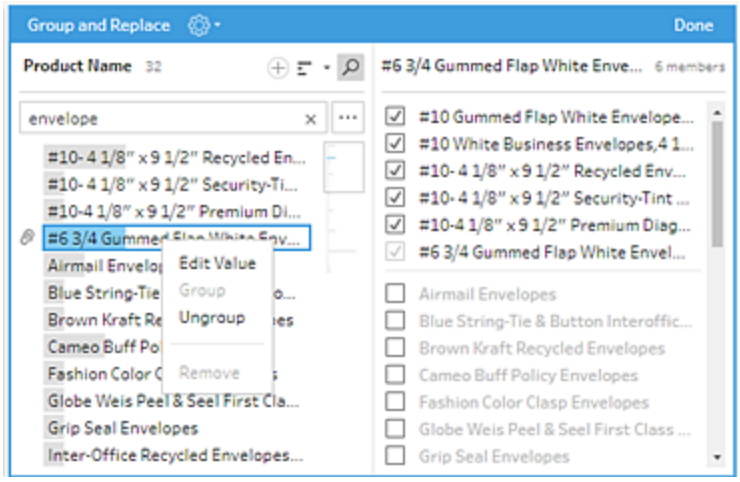
To remove values from the group, in the upper section of the right pane in the **Group and Replace** editor, clear the check box next to the values.

Create a group by selecting multiple values

1. In the **Profile** pane, select the field you want to edit.
2. Click the drop-down arrow, and select **Group and Replace** > **Manual Selection** from the context menu.
3. In the left pane of the **Group and Replace** editor, select multiple values that you want to group.
4. In the right pane of the **Group and Replace** editor, click **Group Values**.



A new group is created using the last selected value as the group name. To edit the group name, select the grouped field and edit the value or right-click or Ctrl-click (Mac) on the grouped field and select **Edit Value** from the context menu.



Edit multiple values using Group and Replace with fuzzy match

To search for and automatically group similar values, use one of the fuzzy match algorithms. Field values are grouped under the value that appears most frequently. Review the grouped values and add or remove values in the group as needed.

Choose one of the following options to group values:

- **Pronunciation**: Find and group values that sound alike. This option uses the Metaphone 3 algorithm that indexes words by their pronunciation and is most suitable for

English words. This type of algorithm is used by many popular spell checkers.

- **Common Characters**: Find and group values that have letters or numbers in common. This option uses the ngram fingerprint algorithm that indexes words by their unique characters after removing punctuation, duplicates, and whitespace. This algorithm works for any supported language.

  For example, this algorithm would match names that are represented as "John Smith" and "Smith, John" because they both generate the key "hijmnost". Since this algorithm doesn't consider pronunciation, the value "Tom Jhinois" would have the same key "hijmnost" and would also be included in the group.

- **Spelling**: Find and group text values that are spelled alike. This option uses the Levenshtein distance algorithm to compute an edit distance between two text values using a fixed default threshold. It then groups them together when the edit distance is less than the threshold value. This algorithm works for any supported language.

To automatically group and replace values using fuzzy match, do the following:

1. In the **Profile** pane, select the field you want to edit.
2. Click the drop-down menu, select **Group and Replace** and select one of these options:
   - **Pronunciation**
   - **Common Characters**
   - **Spelling**

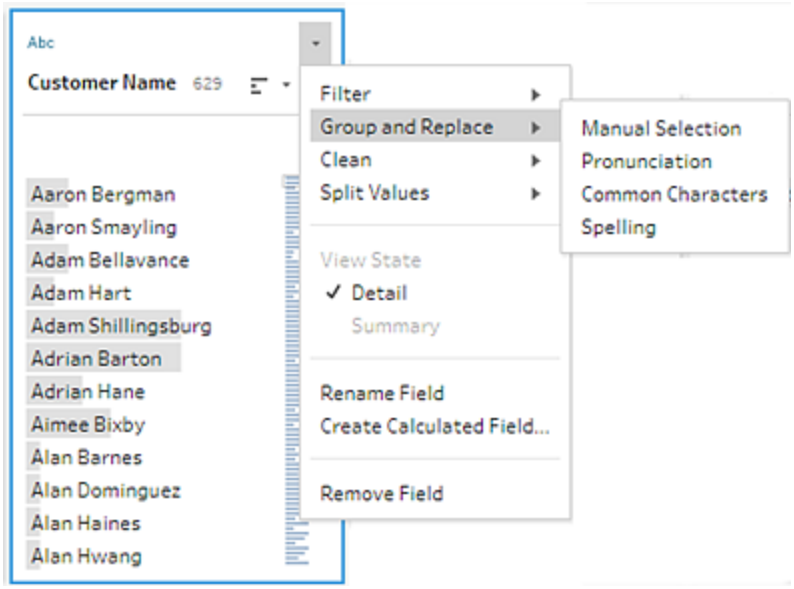
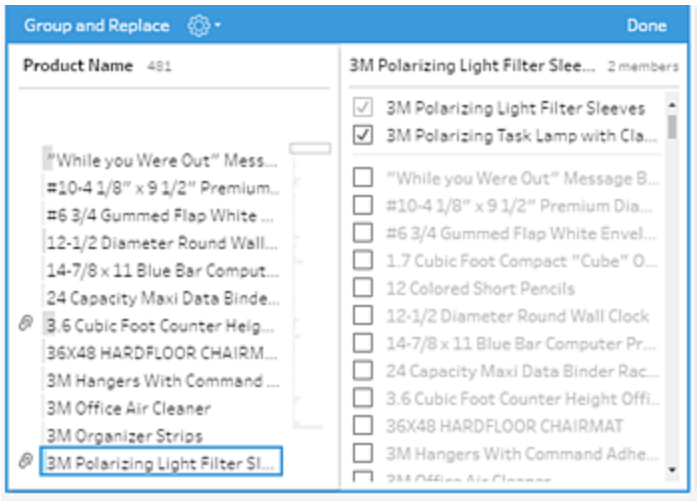


Tableau Prep finds and groups values that match and replaces them with the value that occurs most frequently in the group.

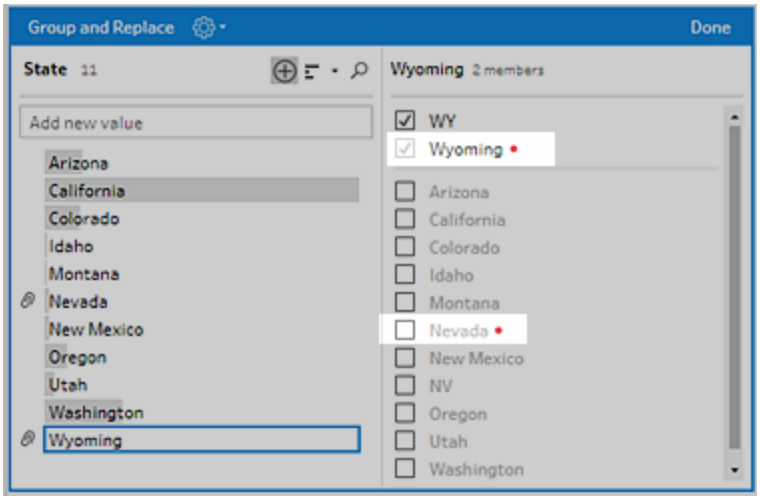
3. Review the groupings and manually add or remove values or edit them as needed. Then click **Done**.



Add and identify values that aren't in the data set

If you want to map values in your data set to a new value that doesn't exist, you can add it using Group and Replace. To easily identify any values that are not in the data set, these values are marked with a red dot next to the value name in the **Group and Replace** editor.

For example in the image below, Wyoming and Nevada aren't in the data set.

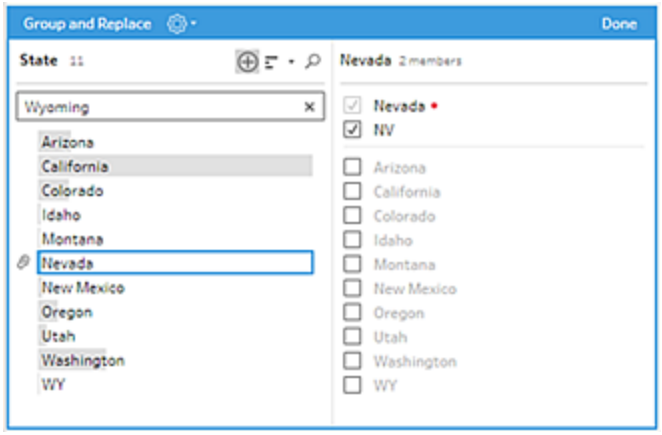


Some reasons why a value might not be in the data set include the following:

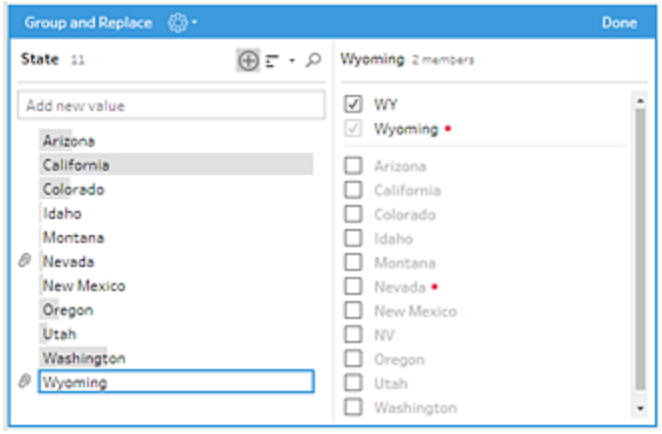
- You just added the new value manually.
- The value is no longer in the data.
- The value is in the data, but isn't in the sampled data set.

To add a new value:

1. In the **Profile** pane, select the field you want to edit.
2. Click the drop-down arrow, and select **Group and Replace** > **Manual Selection** from the context menu.
3. In the left pane of the **Group and Replace** editor, click the plus ⊕ to add a new value.
4. Type a new value in the field and press Enter to add it.



5. In the right pane, select the values that you want to map to the new value.



6. (Optional) To add additional new values to your mapped value, click the plus ⊕ button in the right pane in the **Group and Replace** editor.

Aggregate and group values

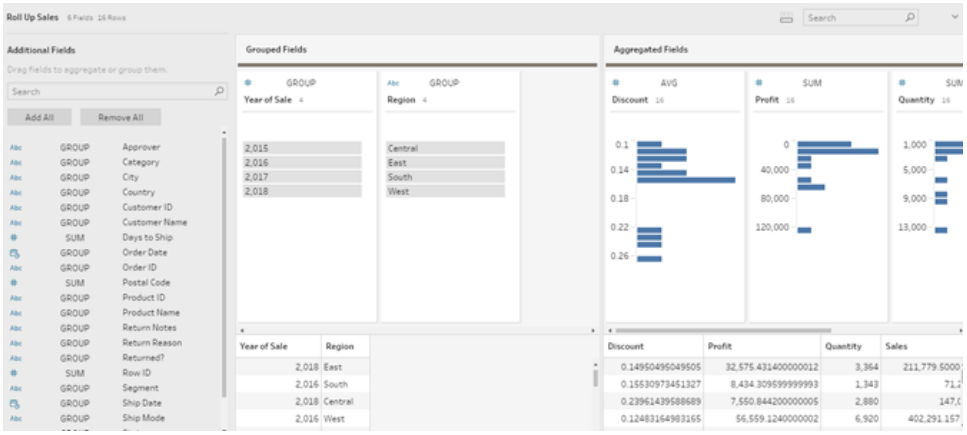
Sometimes you'll need to adjust the granularity of some data, either to reduce the amount of data produced from the flow, or to align data with other data you might want to join or union together. For example, you might want to aggregate sales data by customer before joining a sales table with a customer table.

If you need to adjust the granularity of your data, use the **Add Aggregate** option to create a step to aggregate or group data. Whether data is aggregated or grouped depends on the data type (string, number, or date).

1. In the **Flow** pane, click the plus ⊕ icon, and select **Add Aggregate**. A new aggregation step displays in the **Flow** pane and the **Profile** pane updates to show the aggregate and group profile.
2. To group or aggregate fields, drag them from the left pane to one of the columns in the right pane.

   You can also:

   - Drag and drop fields between the two panes.
   - Search for fields in the list and select only the fields you want to include in your aggregation.
   - Double-click a field to add it to the left or right pane.
   - Change the function of the field to automatically add it to the appropriate pane.
   - Click **Add All** or **Remove All** to bulk apply or remove fields.



Fields are distributed between the **Grouped Fields** and **Aggregated Fields** columns based on their data type. Click the group or aggregation type (for example, AVG or SUM) headings to change the group or aggregation type.

In the data grids below the aggregation and group profile, you can see a sample of the members of the group or aggregation .

Examine and Filter Your Data

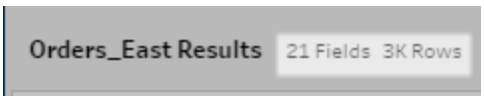
Get a good understanding of the composition of your data to better understand changes you need to make, and the effect of the operations you include in the flow.

In this article

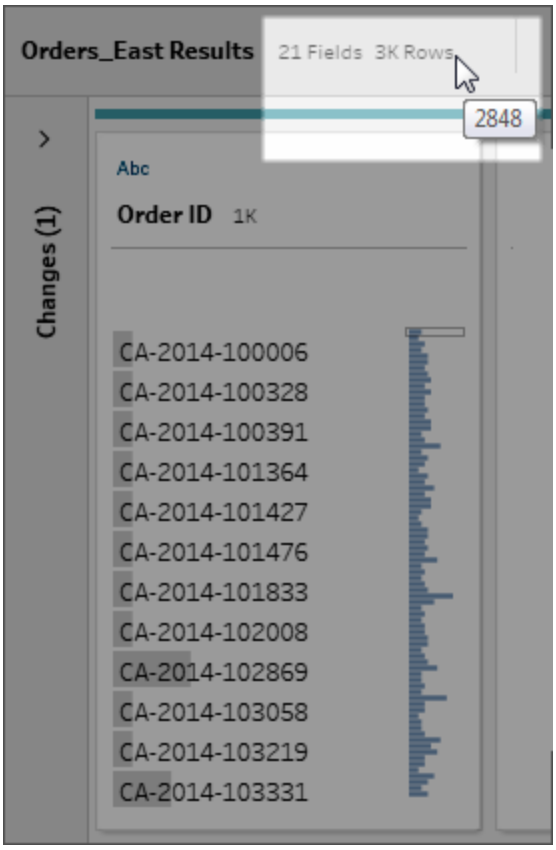
See size details about your data

After you connect to your data, add a table to the flow, and then add a step, you can use the **Profile** pane to see the current state and structure of your data and spot nulls and outliers.

- **Number of fields and rows**: In the upper-left corner of the **Profile** pane you can find information that summarizes the number of fields and rows in the data at a particular point in the flow. Tableau Prep rounds to the nearest thousand. In the example below, there are 21 fields and 3ooo rows in the data set.

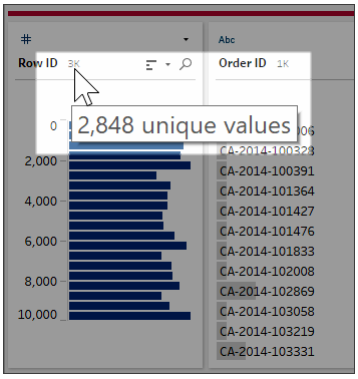
  

  When you hover over the number of fields and rows, you can see the exact number of rows (in this example, 2848):



- **Data set size**: Work with a subset of your data by specifying the number of rows to include in the **Data Sample** tab in the **Input** pane.
- **Sampled**: To enable you to interact directly with your data, Tableau Prep works with a subset of your raw data. The number of rows is determined by the data types and number of fields that are being rendered. String fields take more storage space than integers, so if you have 10 fields of strings in your data set, you might get fewer rows than if you had 10 fields of integers. A **Sampled** Sampled indicator displays next to the size details in the **Profile** pane to indicate that this is a subset of your data set. For more information about data samples, see "Set your data sample size" in **Connect to Data** on page 71.
- **Number of unique values**: The number next to each field header represents the distinct values that are contained within that field. Tableau Prep rounds to the nearest thousand. In the example below, there are 2,000 distinct values that are represented in the Description field, but if you hover over the number, you can see the exact number of

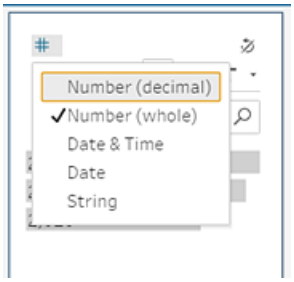
unique values.



Review the data types assigned to your data

Like Tableau Desktop, Tableau Prep interprets the data in your fields when you drag a connection to the **Flow** pane and automatically assigns a data type to it. Because different databases can handle data in different ways, Tableau Prep's interpretation might not always be correct.

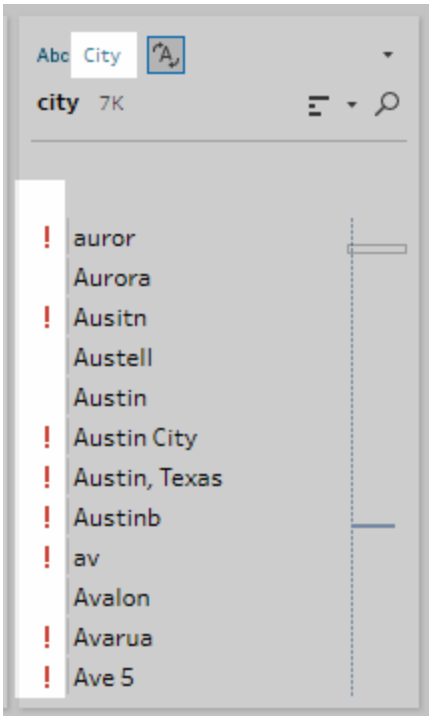
To change a data type, click the data type icon and select the correct data type from the context menu. You can change string or integer data types to Date or Date & Time, and Tableau Prep will trigger Auto DateParse to change these data types. Like Tableau Desktop, if the change is not successful you will see Null values in the fields instead and you can create a calculation to make the change.



For more information about using DateParse, see Convert a Field to a Date Field in the Tableau Desktop and Web Authoring Help.

Assign data roles to your data

To make it easier to identify field values that aren't valid, you can assign a data role to your field the same way you assign a data type. This tells Tableau Prep what your data values represent so it can automatically validate values and highlight ones that aren't valid for that role.



Data roles tell Tableau Prep what the field values mean or represent. For example if you have field values for geographical data, you can assign a data role of **City** and Tableau Prep compares the values in the field to a set of known domain values or patterns to identify values that don't match.

> **Note**: Each field is analyzed independently so a City value of "Portland" in State "Washington" in Country "USA" might not be a valid city and state combination, but it won't be identified that way because it is a valid city name.

Tableau Prep supports the following data roles:

- Email
- URL
- Geographic roles (Based on current geographic data and is the same data used by

Tableau Desktop)

- Airport
- Area code (U.S.)
- CBSA/MSA
- City
- Congressional District (U.S.)
- Country/Region
- County
- NUTS Europe
- State/Province
- Zip code/Postal code

To set a data role for your field, do the following:

1. In the Profile pane, click the data type for the field.
2. Select the data role for the field.

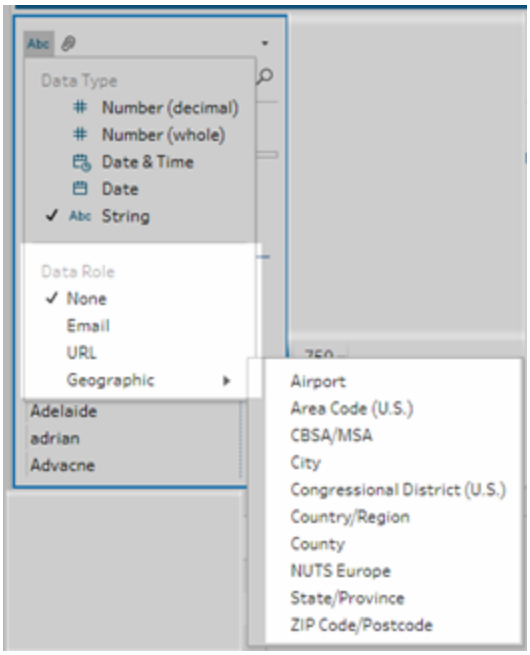
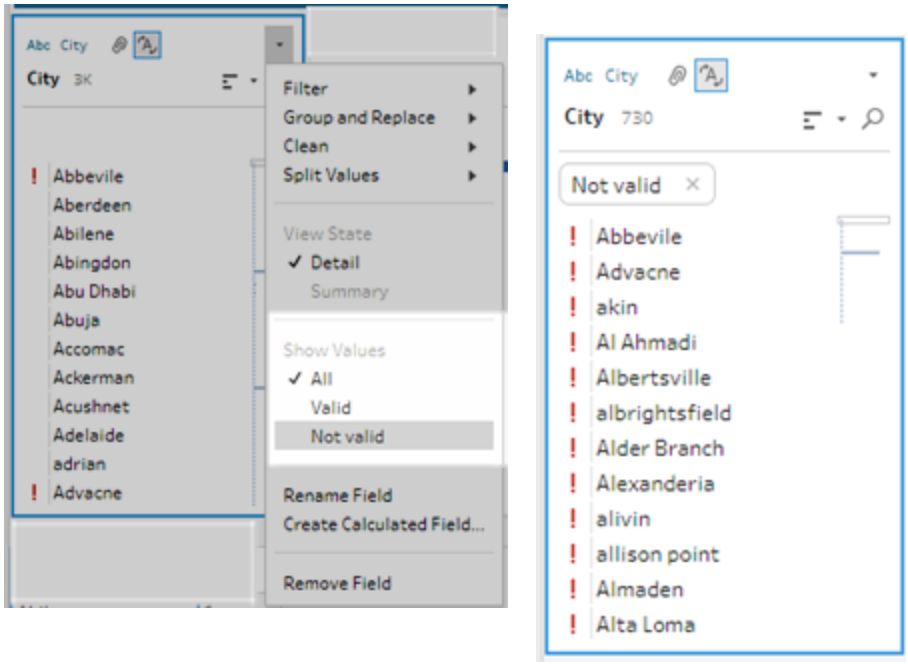


Tableau Prep compares the field's data values to known domain values or patterns (for email or URL) for the data role you select and marks any values that don't match with a red exclamation point.

3. Click the drop-down arrow for the field and from the **Show Values** section select an option to show all values or only values that are valid or not valid for the data role.

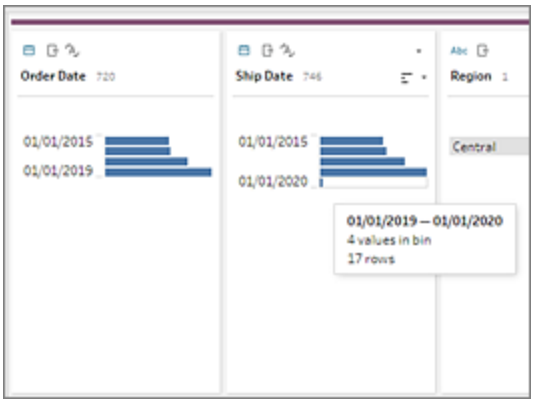


4. Use the cleaning options on the drop-down menu for the field to correct any values that aren't valid. For more information about how to clean your field values see **Apply cleaning operations** on page 93.

See the distribution of values or unique values

By default, Tableau Prep groups numerical, date, and datetime values in a field into buckets. These buckets are also known as bins. The bins ensure that you can see the distribution of values as a whole and quickly identify outliers and null values. The bin size is calculated based on the minimum and maximum values in the field, and null values are always shown at the top of the distribution.

For example, order and ship dates are summarized or "binned" by year. Each bin represents a year from January of the beginning year to January of the following year and labeled accordingly. Because there are sales dates and ship dates that fall in the latter part of 2018 and 2019, a bin is created for the following year for those values.



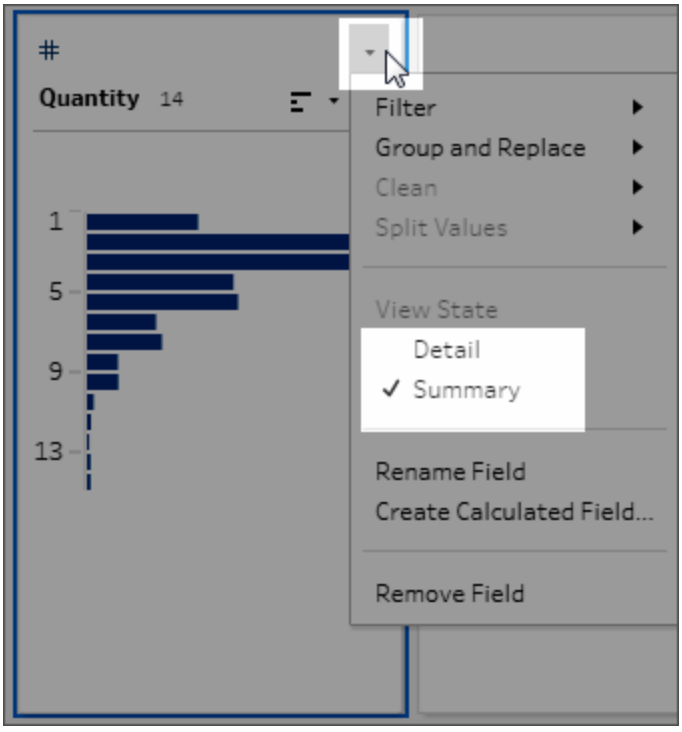
If a discrete (or categorical) data field contains many rows or has a distribution that is large enough that it can't be displayed in the field without scrolling, you can see a summarized distribution to the right of the field. You can click and scroll through the distribution to target specific values.

When your data contains numeric or date fields, you can toggle to display the detailed (discrete) version of the values or a summarized (continuous) version of the values. The summarized view shows you the range of values in a field and the frequency with which certain values appear.

This toggle can help you isolate unique values (like the number of "3" records in a field) or the distribution of values (like the sum of all "3" records in a field)

To toggle your view:

1. In the **Profile** pane, click the drop-down arrow for a numeric or date field.

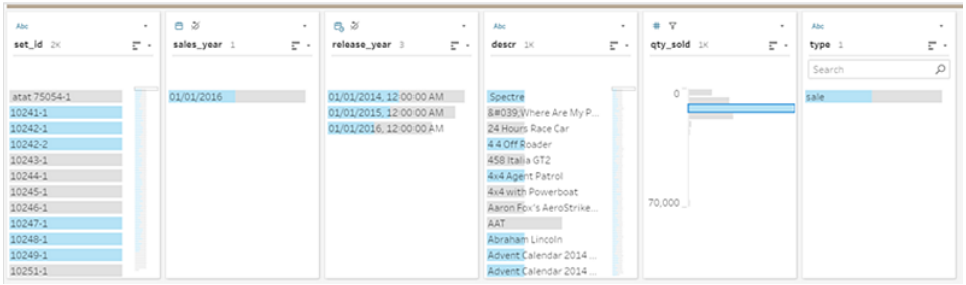


2. In the context menu, select **Detail** to see the detailed version of the values, or **Summary** to see the distributed version of the values.

See related values

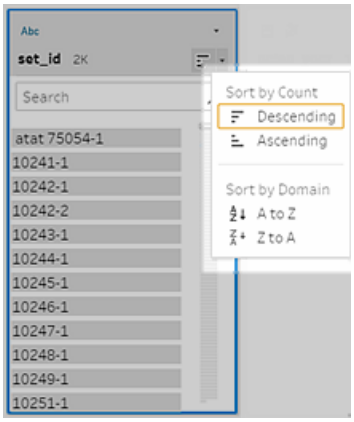
You can use highlighting to find related values across fields. When you click a value in the **Profile** pane, all the related values in the other fields are highlighted in blue. The blue color shows the relationship distribution between the value you selected and the values in the other fields.

To highlight related values, in the **Profile** pane, click a value in a field. The related values in other fields turn blue and the proportion of the bar highlighted in blue represents the degree of association.

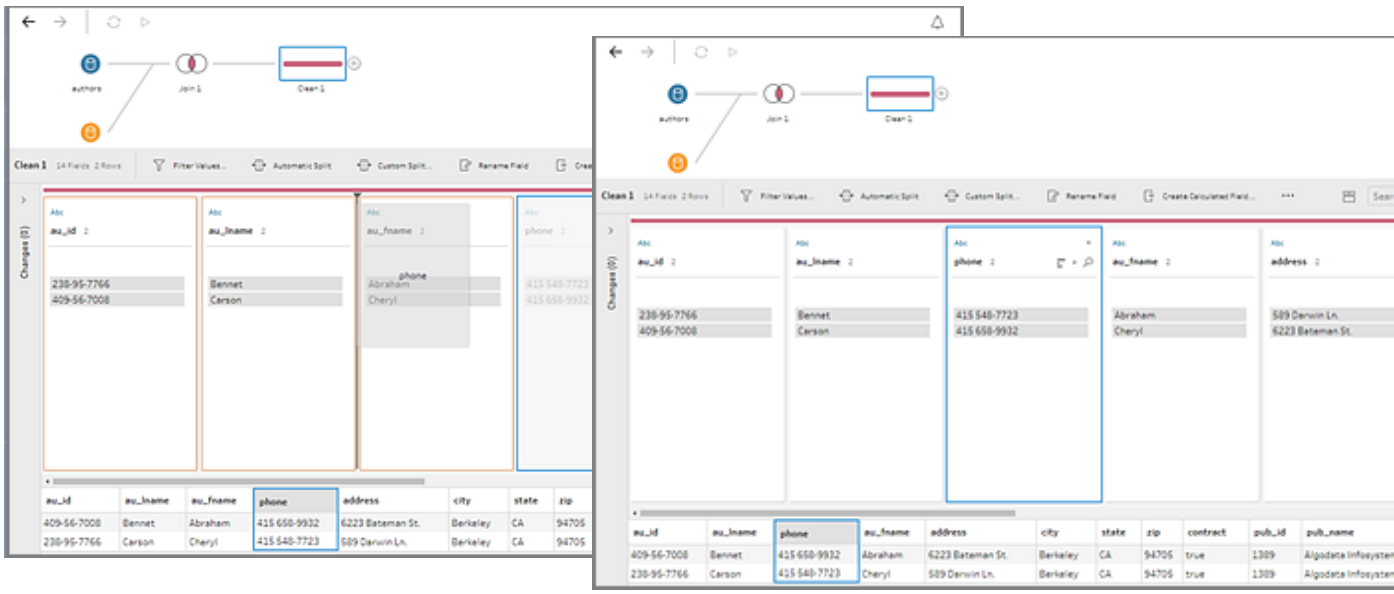


Sort values and fields

In the **Profile** pane, sort options let you sort the bins (the count of values represented by the distribution bars) in ascending or descending order, or the individual field values in alphabetical order.



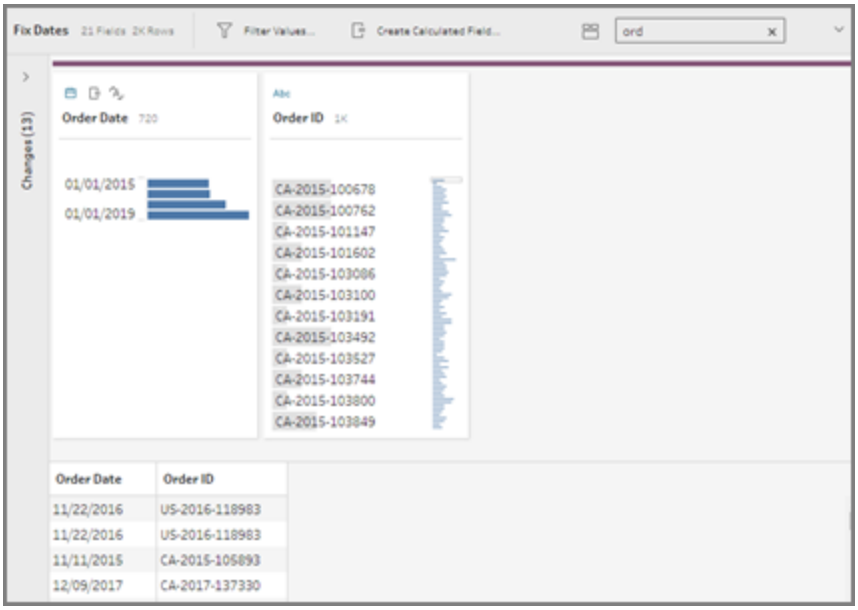
If you want to rearrange the order of your fields, in the **Profile** pane or **Data** grid simply select a profile card or field in the data grid and drag it until you see the black target line appear. Then drop it into place. The Profile pane and data grid are synced so the field will appear in the same order in both places.



Search for fields and values

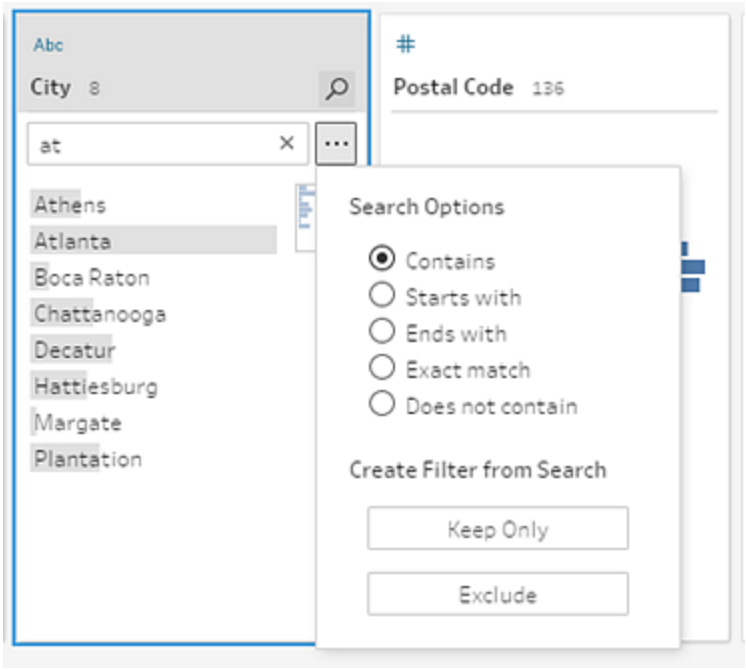
In the **Profile** pane, you can search for fields or values of particular interest to you and use the search results to filter your data.

To search for fields, enter a full or partial search term in the search box on the toolbar in the Profile pane.

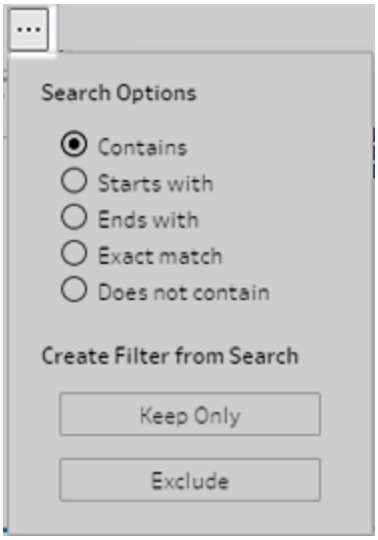


To search for a value in a field:

1. Click the Search icon for a field, and enter a value.



2. To use advanced search options, click the **Search options ...** button.

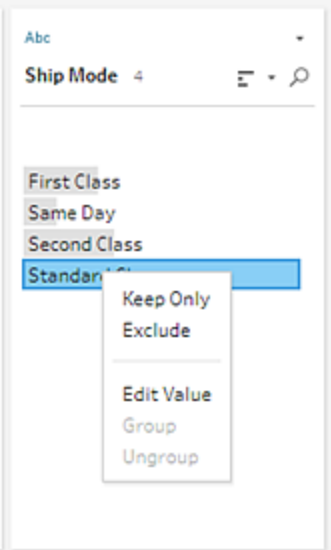


3. To use the search results to filter the data, select **Keep Only** or **Exclude**.

   In the **Changes** pane, a filter icon appears above affected steps.

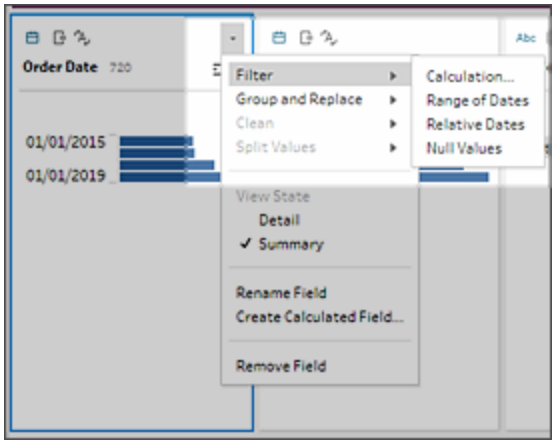
Filter values

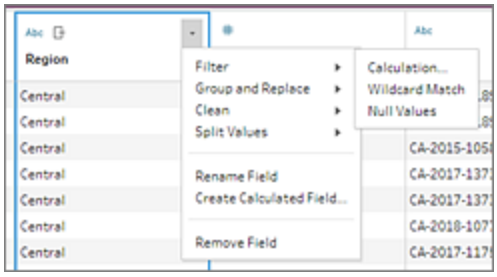
An easy way to filter a value is to select it in either the **Profile** pane or data grid, right-click, and then select **Keep Only** or **Exclude**. In the Profile pane, you can also select **Edit Value** to edit the value in-line.



You can filter data at any step in the flow except the Aggregate, Union, and Pivot steps. To add a filter, in the **Profile** pane or in the data grid, click the drop-down arrow.

Note: To apply a filter in the data grid using the drop-down menu, click the **Hide profile pane** button and then click the drop-down arrow for the field you want to filter.



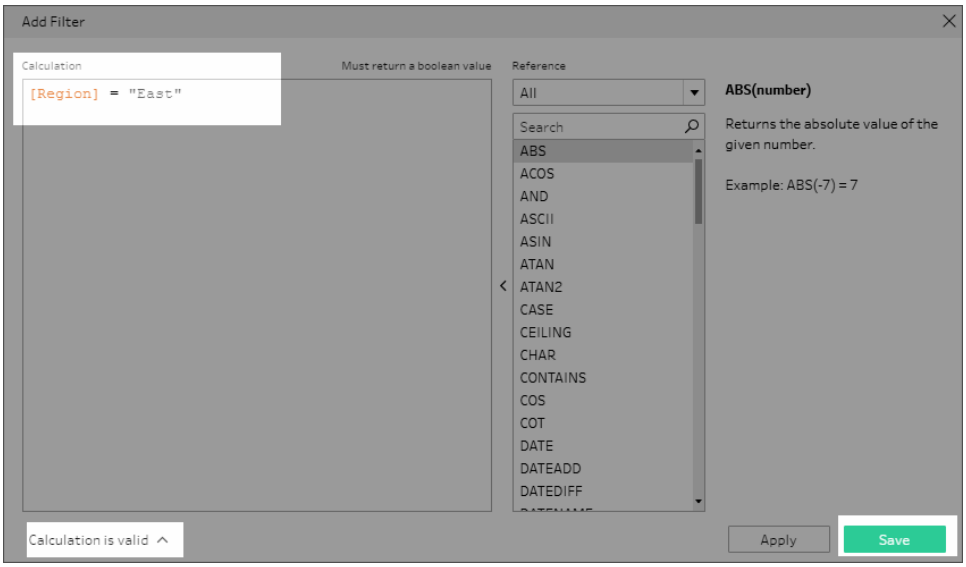


Filters available for each data type

Data type	Available filters
String	Calculation, Wildcard Match, Null Values
Number	Calculation, Range of Values, Null Values
Date & Time	Calculation, Range of Values, Relative Date, Null Values

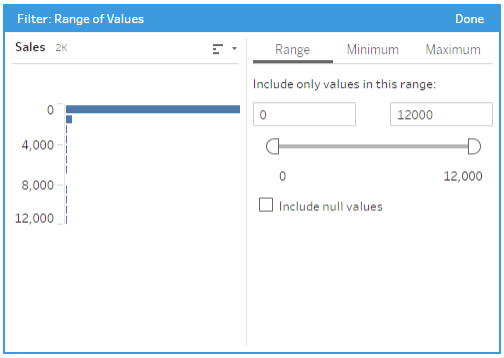
Use a Calculation filter

When you select **Calculation**, the **Add Filter** dialog box opens. Enter the calculation, verify that it's valid, and click **Save**.



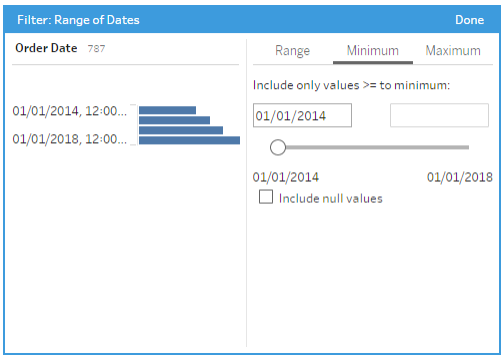
Use a Range of Values filter

When you select **Range of Values**, you can specify a range or set minimum or maximum values.



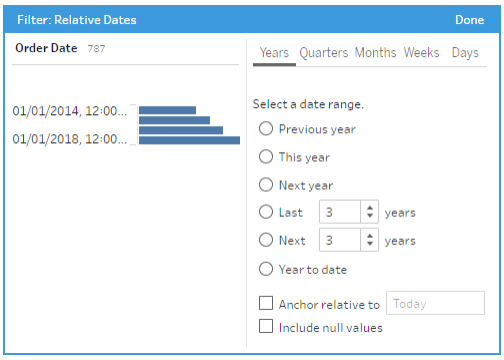
Use Range of Dates filter

When you select **Range of Dates**, you can specify a range of dates or set a minimum or maximum date.



Use a Relative Date filter

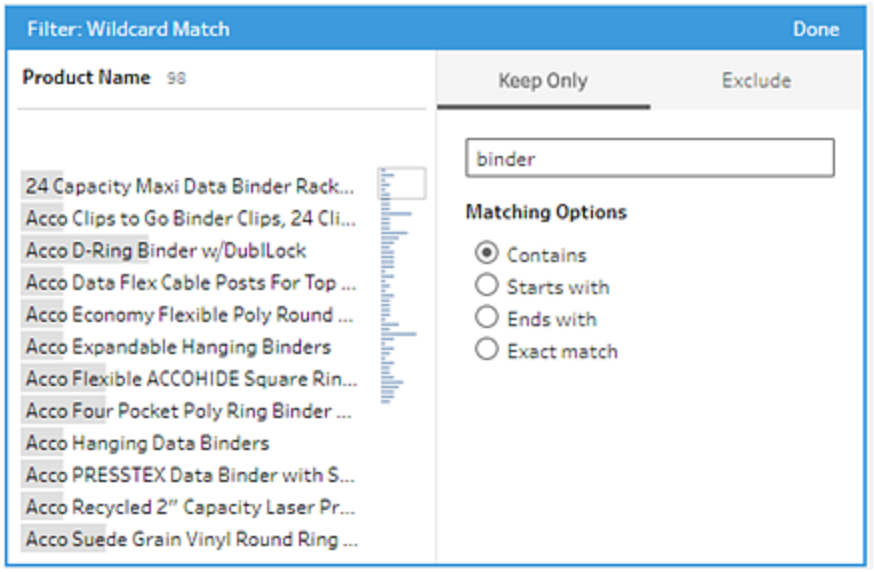
When you select **Relative Dates**, you can filter the date range based on year, quarter, month, week, or day. You can also configure an anchor relative to a specific date, and include null values.



Use a Wildcard Match filter

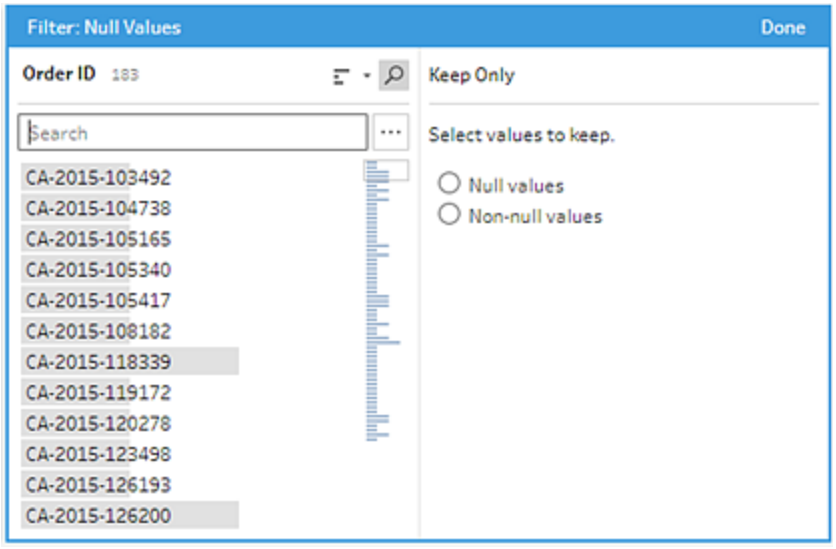
When you select **Wildcard Match**, you can filter the field values to keep or exclude only those values that match your criteria. In the filter editor, select the **Keep Only** or **Exclude** tab, enter a value to match and then set the criteria to return the values you are looking for.

The filtered results display in the left pane of the filter editor so that you can review and experiment with your results. Once you have the results you want, click **Done** to apply your change.



Use a Null Values filter

When you select **Null Values** you can filter the values in the selected field to show only null values or exclude all null values.



Highlight identical values

When you select a value in the data grid, all identical values are highlighted too. These highlights help you identify patterns or irregularities in your data.

Type	Customer	Purchases	Date
Cash	Wei	5	08/18/2016
Cash	Jim	7	07/15/2016
Credit	Arnold	5	06/29/2016
Credit	Lee	1	08/07/2016
Cash	Maria	2	08/30/2016
Cash	Wendy	1	07/21/2016
Credit	Max	2	07/02/2016
Credit	Juan	1	05/10/2016
Cash	Isaac	4	06/28/2016
Credit	Philip	1	08/09/2016
Credit	Lane	5	05/04/2016

Join or Union Data

There are two methods you can use to combine data in Tableau Prep: join and union.

In this article

Join your data

The data that you want to analyze is often made up of a collection of tables that are related by specific fields. Joining is a method for combining the related data on those common fields. The result of combining data using a join is a table that's typically extended horizontally by adding fields of data.

Joining is an operation you can do anywhere in the flow. Joining early in a flow can help you understand your data sets and expose areas that need attention right away.

To create a join, do the following:

1. After you add at least two tables to the flow pane, select and drag the related table to the other table until the **New Join** option displays. You can also click the ⊕ icon and select **Add Join** from the menu. A new join step is added to the flow and the profile pane updates to show the join profile.

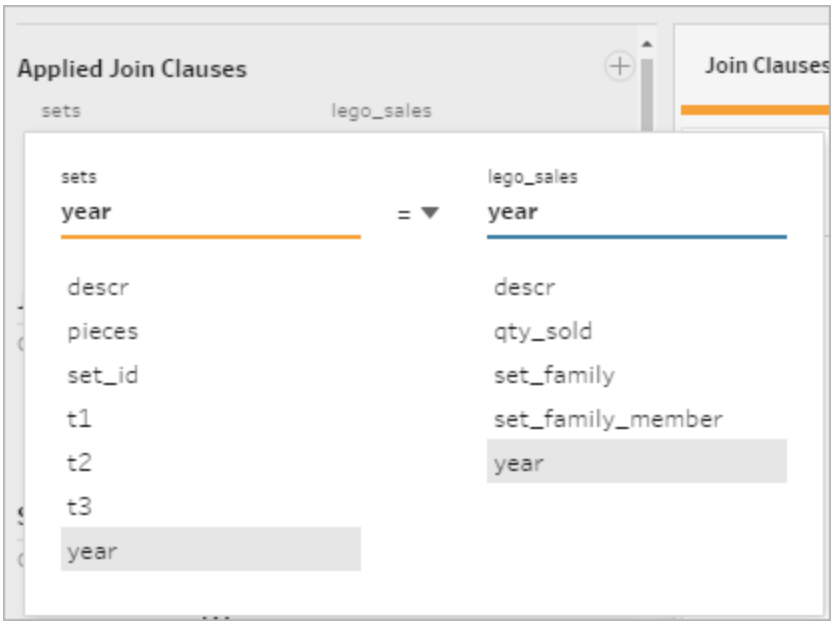
   Note: If you use the ⊕ icon to create a join you will need to manually add the other input to the join and add the join clauses.

2. To review and configure the join, do the following:

   a. Review the **Summary of Join Results** to see the number of fields included and excluded as a result of the join type and join conditions.

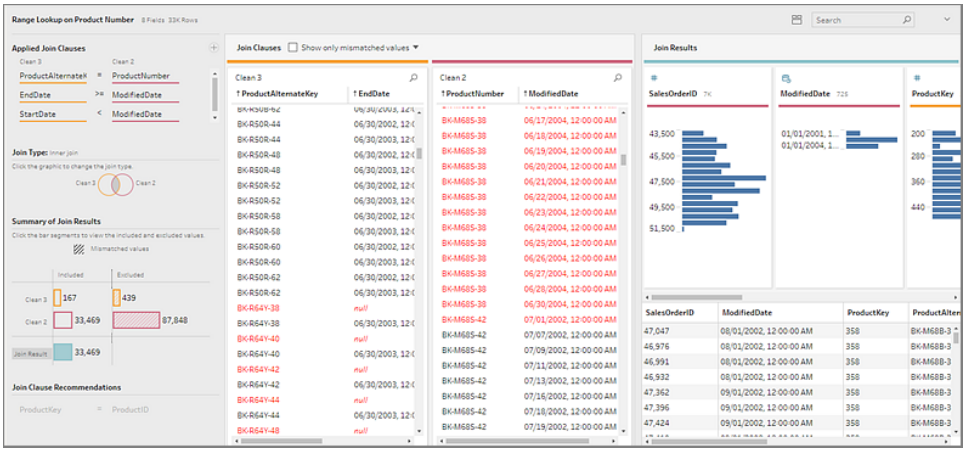
   b. Under **Join Type**, click in the Venn diagram to specify the type of join you want.

   c. Under **Applied Join Clauses**, click the plus ⊕ icon or, on the field chosen for the default join condition, specify or edit the join clause. The fields you selected in the join condition are the common fields between the tables in the join.



   d. Alternatively, you can click the recommended join clauses shown under **Join**

Clause Recommendations to add the clause to the list of applied join clauses.

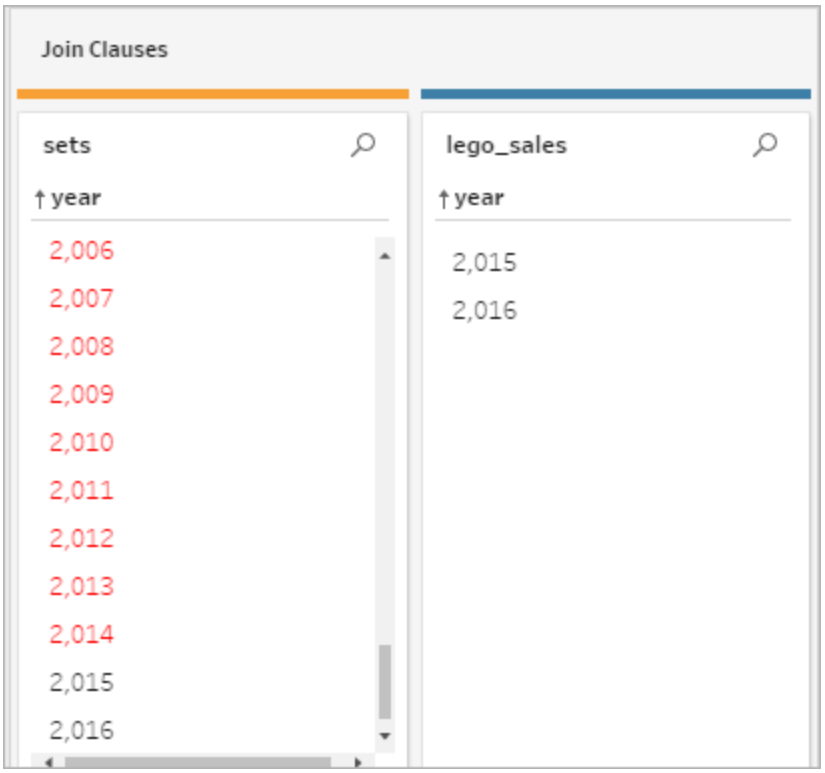


Inspect the results of the join

The summary in the join profile shows metadata about the join to help you validate that the join includes the data you expect.

- **Applied Join Clauses**: By default, Tableau Prep defines the first join clause based on common field names in the tables being joined. Add or remove join clauses as needed.
- **Join Type**: By default, when you create a join, Tableau Prep uses an inner join between the tables. Depending on the data that you connect to, you might be able to use left, inner, right, or outer joins.
- **Summary of Join Results**: The Summary of Join Results shows you the distribution of values that are included and excluded from the tables in the join.
    - Click each **Included** bar to isolate and see the data in the join profile included in the join.
    - Click each **Excluded** bar to isolate and see the data in the join profile that are excluded from the join.
    - Click any combination of the **Included** and **Excluded** bars to see a cumulative perspective of the data.
- **Join Clause Recommendations**: Click the plus ⊕ icon next to the recommended join clause to add it to the **Applied Join Clauses** list.

- **Join Clauses** pane: In the **Join Clauses** pane, you can see the values in each field in the join clause. The values that don't meet the criteria for the join clause are displayed in red text.

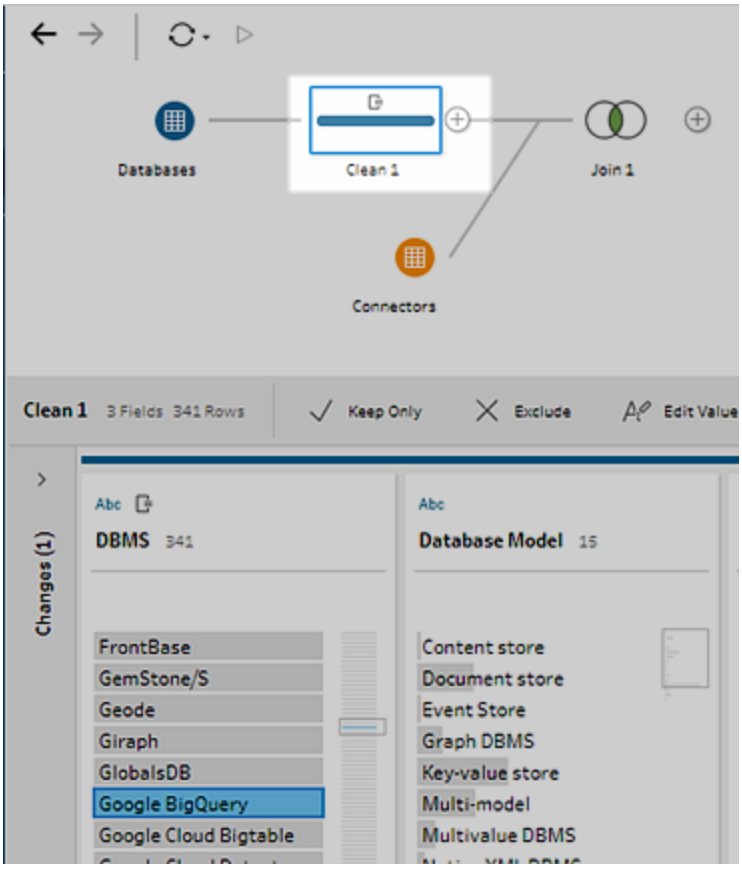


- **Join Results** pane: If you see values in the **Join Results** pane that you want to change, you can edit the values in this pane.

Fix mismatched fields

Fix mismatched fields right in the join clause. Double-click or right-click the value and select **Edit Value** from the context menu on the field that you want to fix and enter a new value.

Your data changes are automatically pushed back to the previous cleaning step for the table where the join data came from. If no cleaning step exists for the table, then one is added automatically.



If you need to perform other types of cleaning operations, manually add a cleaning step to the flow. For more information about cleaning fields, see **Apply cleaning operations** on page 93.

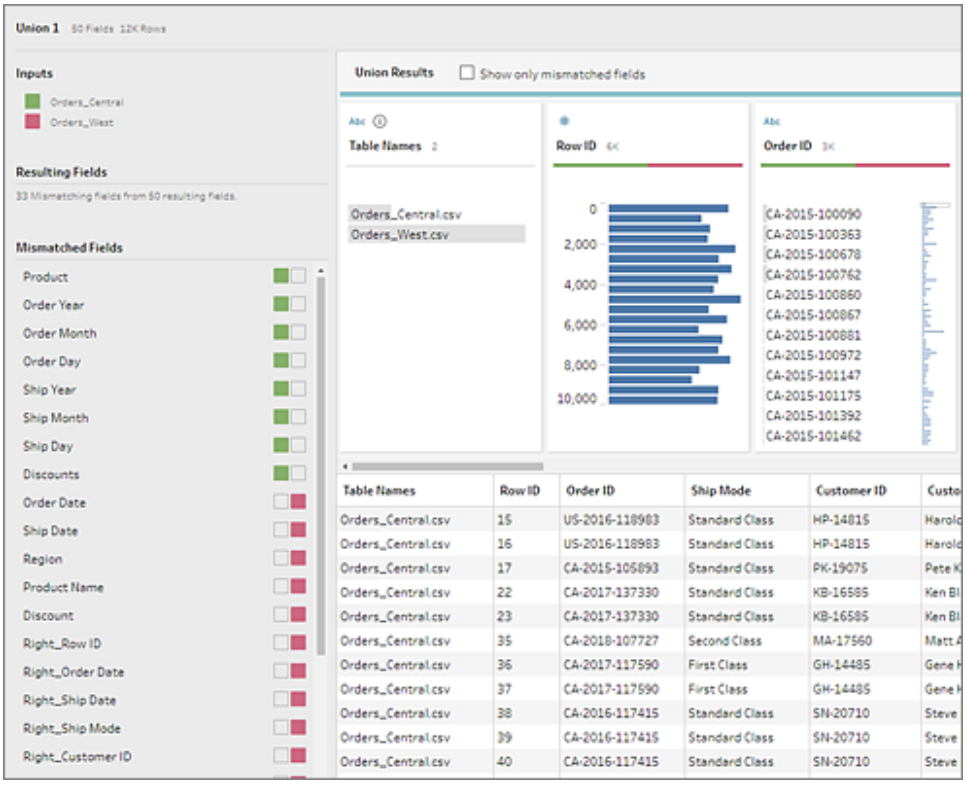
Union your data

Union is a method for combining data by appending rows of one table onto another table. For example, you might want to add new transactions in one table to a list of past transactions in another table. Make sure the tables you union have the same number of fields, the same field names, and the fields are the same data type.

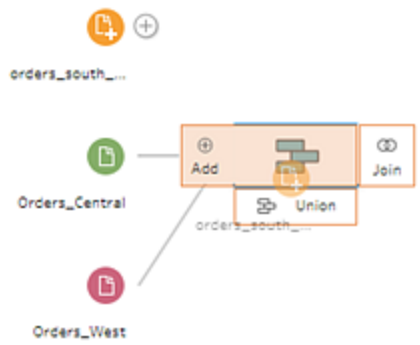
Similar to a join, you can use the union operation anywhere in the flow.

To create a union, do the following:

1. After you add at least two tables to the flow pane, select and drag a related table to the other table until you see the **New Union** option. You can also click the ⊕ icon and select **Add Union** from the menu. A new union step is added in the **Flow** pane, and the **Profile** pane updates to show the union profile.



2. Add additional tables to the union by dragging tables toward the unioned tables until you see the **Add** option.

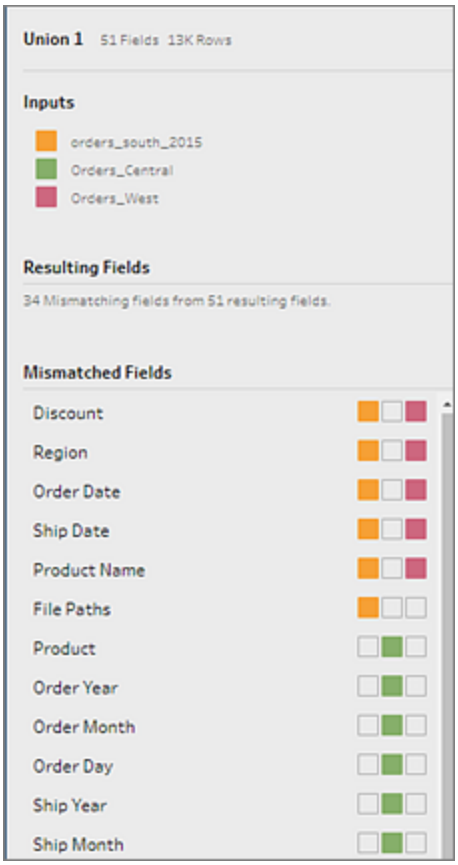


3. In the union profile, review the metadata about the union. You can remove tables from the union as well as see details about any mismatched fields.

Inspect the results of the union

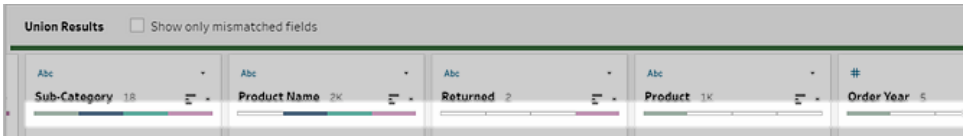
After you create a union, inspect the results of the union to validate that the data in the union is what you expect. There are a number of areas in the union profile that you can check to help you validate the data in the union.

- **Review the union metadata**: The union profile shows some metadata about the union. Here you can see the tables that make up the union, the resulting number of fields and any mismatched fields.

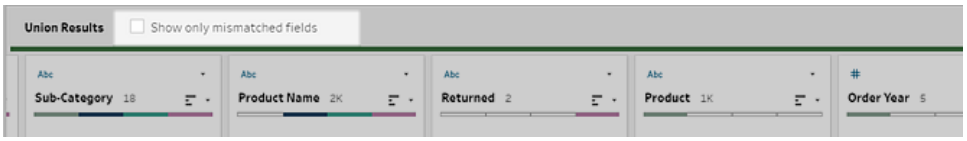


- **Review the colors for each field**: Next to each field listed in the Union summary and above each field in the union profile, is a set of colors. The colors correspond to each table in the union.

  If all table colors show for that field, then the union performed correctly for that field. A missing table color indicates that you have mismatched fields.



Mismatched fields are fields that might have similar data but are different in some way. You can see the list of fields that don't match in the Union summary and the tables where they came from. If you want to take a closer look at the data in the fields, select the **Show only mismatched fields** check box to isolate the mismatched fields in the Union profile.



To fix these field, follow one of the suggestions in the “Fix fields that don’t match” section below.

Fix fields that don’t match

When tables in a union don’t match, the union produces extra fields. The extra fields are valid data being excluded from their appropriate context.

To resolve a field mismatch issue, you must merge the mismatched fields together.

There are a number of reasons why fields might not match.

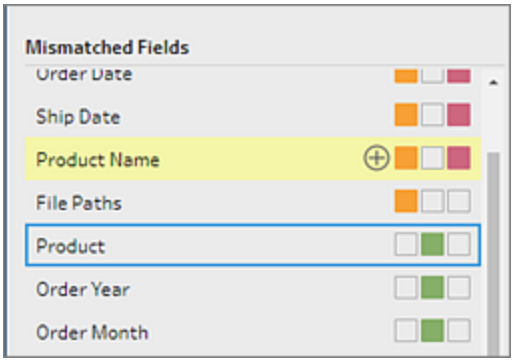
- **Corresponding fields have different names**: If corresponding fields between tables have different names, you can use union recommendations, manually merge fields in the **Mismatched Fields** list, or rename the field in the union profile to merge the mismatched fields together.

  To use union recommendations, do the following:

  1. in the **Mismatched Fields** list, click on a mismatched field. If a suggested match exists, the matching field is highlighted in yellow.

     Suggested matches are based on fields with similar data types and field names.

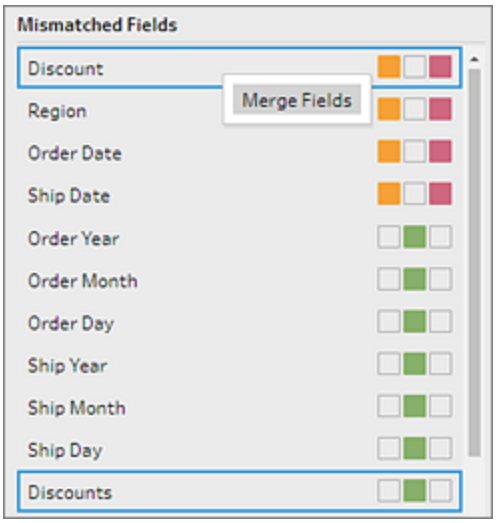
  2. Hover on the highlighted field and click the plus button to merge the fields.



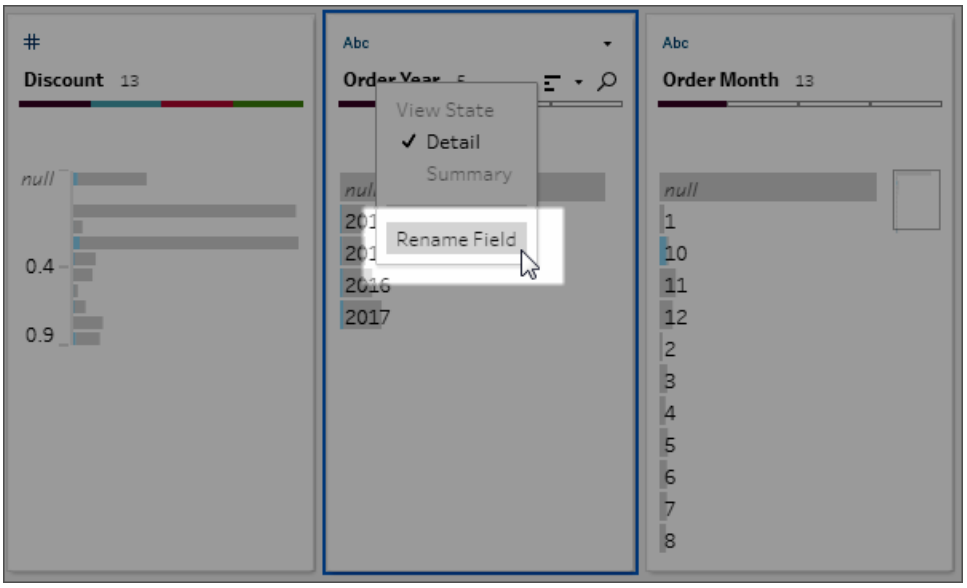
To manually merge fields in the **Mismatched Fields** list, do the following:

1. Select one or more fields in the list.
2. Right click a selected field and if the merge is valid, the **Merge Fields** menu option appears.

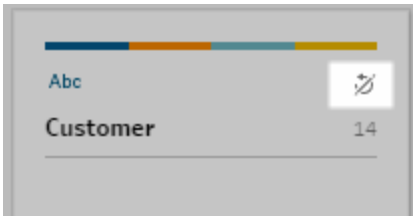
   If you see **No options available** when you right-click the field, this is because the fields are not eligible to merge. For example trying to merge two fields from the same input.
3. Click **Merge Fields** to merge the selected fields.



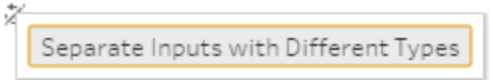
To rename the field in the union profile pane, right-click the field name and click **Rename Field**.



- **Corresponding fields have the same name but are a different type:** By default, when the name of corresponding fields match but the data type of the fields don't, Tableau Prep will change the data type of one of the fields so they are compatible with each other. If Tableau Prep makes this change, it's noted at the top of the merged field by the Change Data Type icon.



In some cases, Tableau Prep might not pick the correct data type. If that happens and you want to undo the merge, right-click the Change Data Type icon and select **Separate Inputs with Different Types**.



You can then merge the fields again by first changing the data type of one of the fields and then using the suggestions in **Additional merge field options** on the next page.

- **Corresponding tables have different number of fields**: To union tables, each table

in the union must contain the same number of fields. If a union results in extra fields, merge the field into an existing field.

Additional merge field options

In addition to the methods described in the above section for merging fields you can also use one of the following methods to merge fields. You can merge fields in any step, except for the following steps: Join, Aggregate, Input, or Output.

For information about how to merge fields in the same file, see **Merge fields** on page 95.

To merge fields, do one of the following:

- Drag and drop one field onto another. A **Drop to merge fields** indicator displays.
- Select multiple fields and right-click within the selection to open the context menu, and then click **Merge Fields**.
- Select multiple fields, and then click **Merge Fields** on the context-sensitive toolbar.

Save and Share Your Work

At any point in the flow, you can save your flow, view a preview of the data in your flow in Tableau Desktop, or create an extract of your data that includes all the operations that you've applied to your flow. You can also package your data with your flow to share it with others or publish your data extract to Tableau Server or Tableau Online as a data source.

In this article

Save a flow

Save your flow to back up your work before performing any additional operations. Your flow is saved in the Tableau Prep flow (.tfl) file format.

You can also package your local files (Excel, Text Files, and Tableau extracts) with your flow to share with others, just like packaging a workbook for sharing in Tableau Desktop. Only local files can be packaged with a flow. Data from database connections, for example, aren't included.

When you save a packaged flow, the flow is saved as a Packaged Tableau Flow File (.tflx).

- To save your flow, from the top menu, select **File** > **Save**.
- To package your data files with your flow, from the top menu, do one of the following:
    - Select **File** > **Export Packaged Flow**
    - Select **File** > **Save As**. Then in the **Save As** dialog, select **Packaged Tableau Flow Files** from the **Save as type** drop down menu.

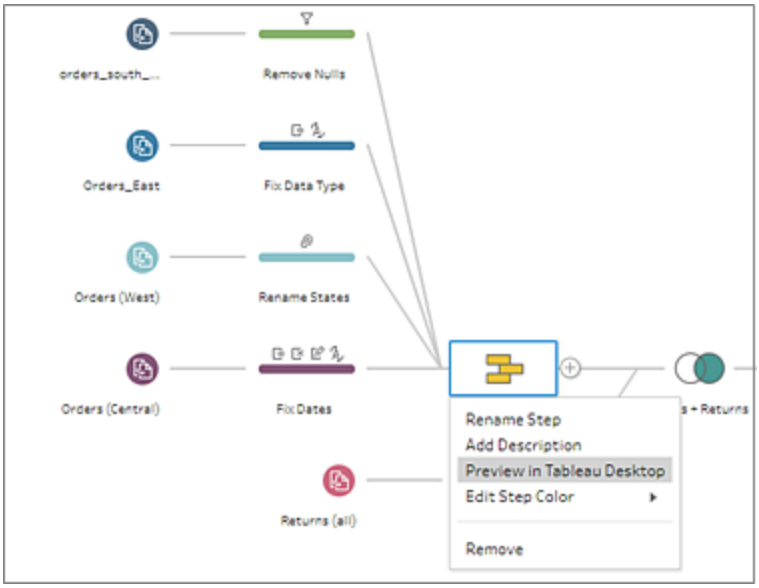
View your data sample in Tableau

Sometimes when you're cleaning your data you might want to check your progress by looking at it in Tableau Desktop. When your flow opens in Tableau Desktop, Tableau Prep creates a permanent Tableau extract (.tde or .hyper depending on your version of Tableau) and a Tableau data source (.tds) file. The files are saved in your Tableau repository in the **Datasources** file so you can experiment with your data at any time.

When you open the flow in Tableau Desktop, you can see the data sample that you are working with in your flow with the operations applied to it, up to the step that you selected.

To view your data sample in Tableau do the following:

1. Right-click the step where you want to view your data, and select **Preview in Tableau Desktop** from the context menu.



2. Tableau Desktop opens on the **Sheet** tab.

Create and publish data extracts and data sources

To create an extract, run your flow. When you run your flow your changes are applied to your entire data set. Running the flow results in a Tableau Data Source (.tds) and a Tableau Data

Extract (.tde or .hyper) file. You can create an extract file from your flow output to use in Tableau Desktop or to share your data with third parties.

> **Note:** You can publish data extracts or data sources to Tableau Server version 10.0 and later as well as to Tableau Online.
>
> Setting up schedules in Tableau Server or Tableau Online to automatically refresh your data outputs is not yet supported but you can refresh your output files from the command line. For more information, see **Refresh output files from the command line** on page 141.

You can create an extract file in the following formats:

- Tableau Data Extract (.tde): The extract is saved as both a Tableau extract (.tde) and a Tableau data source (.tds) file. Use this file type if you use Tableau Desktop or Tableau Server version 10.0 through 10.4.
- Hyper Extract (.hyper): This is the new Tableau extract file type but can only be consumed by Tableau Desktop or Tableau Server version 10.5 and later.
- Comma Separated Value (.csv): Save the extract to a .csv file to share your data with third parties.

You can also publish your output as a data source to Tableau Server or Tableau Online to share your data and provide centralized access to the data you have cleaned, shaped, and combined.

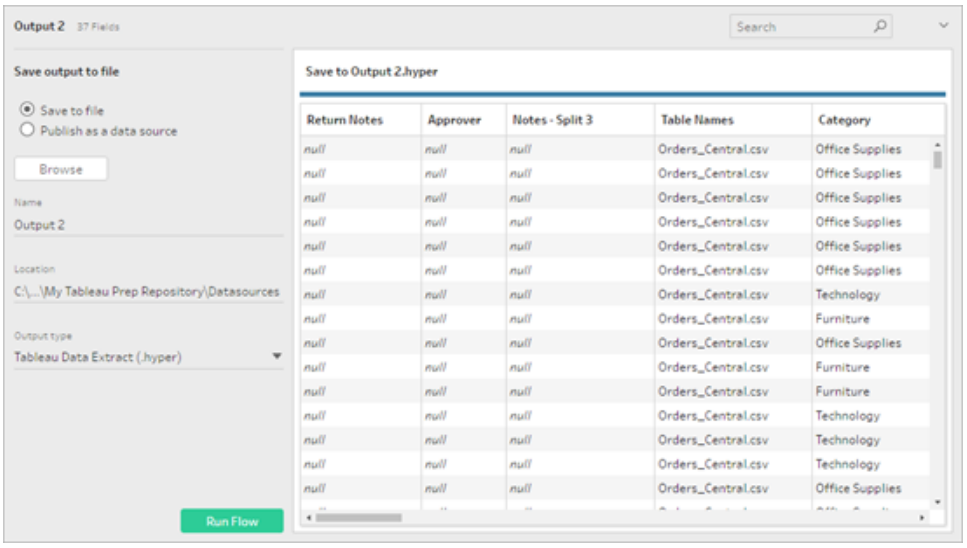
> **Note**: To publish your output to Tableau Server, the Tableau Server REST API must be enabled. For more information see Rest API Requirements in the Tableau Rest API Help. To publish to a server that uses Secure Socket Layer (SSL) encryption certificates, additional configuration steps are needed on the machine running Tableau Prep. For more information, see the System Requirements in the Tableau Desktop and Tableau Prep Deployment Guide.

Create an extract file

1. Click the plus icon ⊕ on a step and select **Add Output**.

   If you have run the flow before, click the run flow ▷ button on the Output step. This overwrites the previous output.

   The **Output** pane opens and shows you a snapshot of your data.



2. In the left pane select **Save to file**.
3. Click the **Browse** button, then in the **Save Extract As** dialog, enter a name for the file and click **Accept**.
4. In the **Output type** field, select the output type. Depending on the version of Tableau Desktop you use you can choose from the following options:
   - Tableau Data Extract (.hyper) for Tableau Desktop version 10.5 and later.
   - Tableau Data Extract (.tde) for Tableau Desktop version 10.0 through 10.4.
   - Comma Separated Values (.csv) if you want to share the extract with a third party.

Tip: You have choices when generating output from your flow. You can generate an extract file, or you can publish your data as a data source to Tableau Server or

Tableau Online. For more information about generating output files, see **Create and publish data extracts and data sources** on page 137.

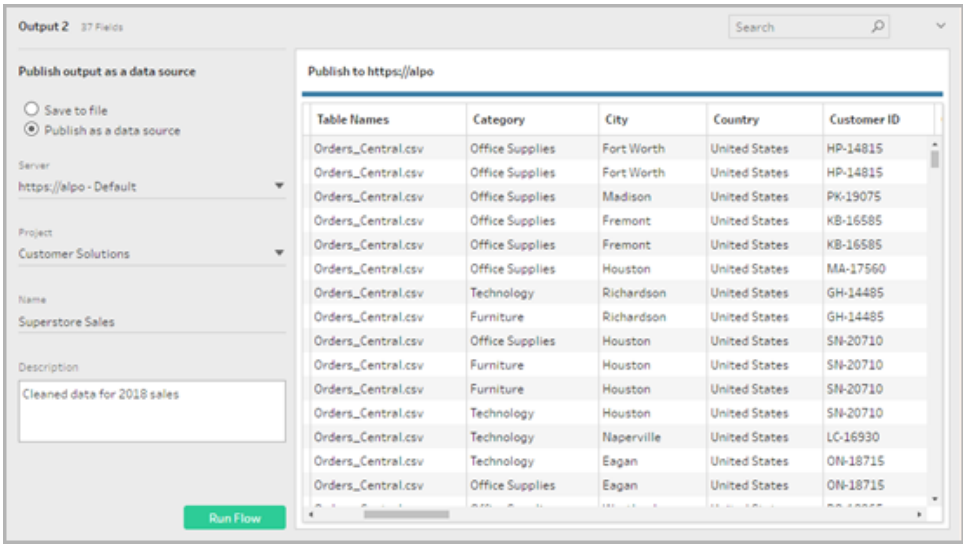
5. Click **Run Flow** to run the flow and generate the extract file.

Publish as a data source

1. Click the plus icon ⊕ on a step and select **Add Output**.

   If you have run the flow before, click the run flow ▷ button on the Output step. This overwrites the previous output.

2. The output pane opens and shows you a snapshot of your data.



3. Select the **Publish as data source** radio button and complete the following fields:

   - **Server**: Select the Tableau server where you want to publish the data source and data extract. If you aren't signed in to a server you will be prompted to sign in.

     On the Mac you may be prompted to provide access to your Mac keychain so Tableau Prep can securely use SSL certificates to connect to your Tableau Server or Tableau Online environment.

   - **Project**: Select the project where you want to load the data source and extract.

- **Name**: Enter file name
- **Description**: Enter a description for the data source.

4. Click **Run Flow** to run the flow and publish the data source.

Refresh output files from the command line

If you want to refresh output files for your flow you can run the flow from the command line instead of opening it in Tableau Prep and running the flow from there.

This option is available on both Windows and Mac machines where Tableau Prep is installed.

For Windows machines, you can also schedule this process using Windows Task Scheduler. For more information about using Windows Task Scheduler, see Task Scheduler in the Microsoft online help.

When you run flows from the command line, Tableau Prep refreshes all outputs for the flow.

> **Note:** The output location for the files is specified in the output step for the flow in Tableau Prep when you run the flow. When you refresh the files from the command line, this process uses that same location and will overwrite any previous output files for the flow with the refreshed version.
>
> For information about how to specify an output location for your flow files, see **Create and publish data extracts and data sources** on page 137.

To run the flow from the command line, you'll need the following:

- Administrator privileges on the machine where you are running the flow.
- The path where Tableau Prep is installed.
- If connecting to databases and publishing output files to a server, a credentials .json file that includes all required credentials.
- The path where the Tableau Flow (.tfl) file is located.

Before running the flow

If you are running a flow that connects to database files or publishes output files to a server, then you'll need to create a .json file that includes the credentials that are required to connect to

these locations.

When you run the process, the hostname, port and username provided in the credentials .json file is used to find the matching connection in the Tableau flow file (tfl) and updated before running the process.

> **Note:** Skip this step if the flow connects to and outputs to local files, files stored on a network share or input files that use Windows Authentication (SSPI). For more information about Windows Authentication, see SSPI Model in the Microsoft online help.

If you plan to reuse the file, place it in a folder where it won't be overwritten by the Tableau Prep install process.

The following table lists the credentials that you need to include in the .json file. Port ID and Site ID are optional if your connections don't require this information.

Input connections	Output location
• User name • Host name (Server name) • Port ID • Password	• Server Url • Content Url (Site ID. This appears after /site/ in the URL when you sign into Tableau Server or Tableau Online) • User name • Password

The following example shows the syntax to use for the credentials .json file.

In this example the flow connects to two databases (Microsoft SQL Server and Oracle) and outputs files to a server that includes a Site ID.

```
{
"inputConnections":[
  {
    "username": "jsmith",
    "hostname":"mssql.example.lan",
    "port":1234,
    "password": "passw0rd"
```

```
  },
  {
    "username": "jsmith",
    "hostname":"Oracle.example.lan",
    "port":5678,
    "password": "passw0rd"
  }
],
"outputConnections":[
  {
    "serverUrl":"http://MyServer",
    "contentUrl":"FinanceTeam",
    "username":"jsmith",
    "password":"passw0rd$"
  }
]
}
```

Tips for creating your credentials file

Tableau Prep uses information from the flow file and from the credentials .json file to run the flow when you have remote connections. For example, the database name for your remote connections and the project name for your output files come from the flow, and the server name and the log in credentials come from the .json file.

To avoid errors when running the flow, make sure your credentials file follows these guidelines:

- Always include the "inputConnections" and "outputConnections" arrays even if the flow doesn't have remote connections for inputs or outputs. Just leave those arrays blank.
    - No remote input connection? Include this syntax at the top of the .json file

      ```
      {
      "inputConnections":[
        ],
      ```

    - No remote output connection? Include this syntax at the bottom of the .json file

```
"outputConnections":[
  ]
}
```

- No port ID for your input connection? Don't include the `"port":xxxx,` reference in the .json file, not even `"port": ""`.
- When referencing the `"serverUrl":` don't include a "/" at the end of the address. For example, use this `"serverUrl": "http://server"` not this `"serverUrl": "http://server/"`.
- If you have multiple input or output connections include the credentials for each one in the file.

Run the flow

1. Open the command prompt or terminal command prompt (Mac) as an Administrator.
2. Run one of the following commands:
    - The flow connects to local files or files stored on a network share and publishes to local files, files stored on a network share or uses Windows authentication:

      > **Note**: If connecting to or outputting to files stored on a network share, use the UNC format for the path: \\server\path\file name. It can't be password protected.

      Windows: `"\[Tableau Prep install location]\Tableau Prep <version>\scripts"\tableau-prep-cli.bat -t "path\to\[your flow file name].tfl"`

      Mac: `/Applications/Tableau\ Prep\[Tableau Prep version].app/Contents/scripts/./tableau-prep-cli -t path/to/[your flow file name].tfl`
    - The flow connects to databases or publishes to a server:

      Windows: `"\[Tableau Prep install location]\Tableau Prep <version>\scripts"\tableau-prep-cli.bat -c "path\to\[your credential file name].json" -t "path\to\[your flow file name].tfl"`

Mac: `/Applications/Tableau\ Prep\[Tableau Prep version].app/Contents/scripts/./tableau-prep-cli -c path/to/[your credential file name].json -t path/to/[your flow file name].tfl`

- The flow file or credentials file is stored on a network share (use the UNC format for the path: \\server\path\file name):

  Windows: `"\[Tableau Prep install location]\Tableau Prep <version>\scripts"\tableau-prep-cli.bat -c "\server\path\[your credential file name].json" -t "\server\path\[your flow file name].tfl"`

  Mac: Map the network share to /Volumes in Finder so that it is persistent, then use `/Volumes/.../[your file]` to specify the path:

  `/Applications/Tableau\ Prep\ [Tableau Prep version].app/Contents/scripts/./tableau-prep-cli -c /Volumes/.../[your credential file name].json -t path/to/[your flow file name].tfl`

For examples of sample commands see **Syntax examples** on the next page.

Command options

If you want to view the help options, include `-h` in the command line.

Command options	Description	Notes
`-c`	The connection path to the credentials file.	Requires the path to where the credentials file is located.

`-d`	Debug the flow process.	Include this option to view more information to help debug a problem with refreshing the flow. Log files are stored in: `My Tableau Prep Repository\Command Line Repository\Logs`

`-h`	View the help for syntax options	The help option or a syntax error shows the following information:
		`usage: tableau-prep-cli [-c <arg>] [-d] [-h] [-t <arg>]`
		`-c,--connections <arg>  path to a file with all connection information`
		`-d,--debug  This option is for debugging`
		`-h,--help  print usage message`
		`-t,--tflFile <arg>  The tableau prep flow file`

`-t`	The .tfl flow file	Requires the path to where the .tfl flow file is located.

Syntax examples

The command lines below show three different examples for running a flow using the following criteria:

- **Tableau Prep version**: 2018.2.2
- **Flow name**: Flow1.tfl
- **Flow location**: C:\Users\jsmith\Documents\My Tableau Prep Repository\Flows
- **Credentials file name**: Flow 1.json
- **Credentials file location**: C:\Users\jsmith\Desktop\Flow credentials
- **Credentials file location stored on a network share**: \tsi.lan\files\Flow credentials

The flow connects to and publishes to local files

Windows: `"\Program Files\Tableau\Tableau Prep 2018.2.2\scripts"\tableau-prep-cli.bat -t "\C:\Users\jsmith\Documents\My Tableau Prep Repository\Flows\Flow1.tfl"`

Mac: `/Applications/Tableau\ Prep\ 2018.2.2.app/Contents/scripts/./tableau-prep-cli -t /Users/jsmith/Documents/My\ Tableau\ Prep\ Repository/Flows.Flow1.tfl`

The flow connects to databases and publishes to a server

Windows: `"\Program Files\Tableau\Tableau Prep 2018.2.2\scripts"\tableau-prep-cli.bat -c "\C:\Users\jsmith\Desktop\Flow credentials\Flow1.json" -t "\C:\Users\jsmith\Documents\My Tableau Prep Repository\Flows\Flow1.tfl"`

Mac: `/Applications/Tableau\ Prep\ 2018.2.2.app/Contents/scripts/./tableau-prep-cli -c /Users/jsmith/Desktop/Flow\ credentials/Flow1.json -t /Users/jsmith/Documents/My\ Tableau Prep\ Repository/Flows.Flow1.tfl`

The flow publishes to a server and the credentials file is stored on a network share

Windows: `"\Program Files\Tableau\Tableau Prep 2018.2.2\scripts"\tableau-prep-cli.bat -c "\\tsi.lan\files\Flow credentials\Flow1.json" -t "\C:\Users\jsmith\Documents\My Tableau Prep Repository\Flows\Flow1.tfl"`

Mac: `/Applications/Tableau\ Prep\ 2018.2.2.app/Contents/scripts/./tableau-prep-cli -c /Volumes/files/Flow\ credentials/Flow1.json -t /Users/jsmith/Documents/My\ Tableau\ Prep Repository/Flows.Flow1.tfl`

(Back to top)

Day in the Life Scenarios

What does it mean to shape data? How does that impact what visualizations can be built and what analysis can be performed? In the tutorials below, we explore scenarios for analysis and visualization, identify the data limitations holding us back, then see how Tableau Prep can help us shape the data to reach our intended outcome.

Download the data sets and follow along with these day in the life scenarios using Tableau Prep and Tableau Desktop. Learn how to apply the features and functions in Tableau Prep to get your data ready for analysis in Tableau Desktop.

> **Give us your feedback**. We are just starting to build this section of the online help. If there are specific scenarios you'd love to see here, please let us know. Use the feedback bar at the top of the page to tell us more.

To complete the tasks in these tutorials, you need Tableau Prep and Tableau Desktop installed, and you'll need to download and save the data to your computer.

For information about how to install Tableau Prep and Tableau Desktop, see Install Tableau Prep and Install Tableau Desktop in the Tableau Desktop and Tableau Prep Deployment guide. Otherwise you can download the Tableau Prep and Tableau Desktop free trials.

Hospital Bed Use with Tableau Prep

Reaching capacity in a hospital is problematic but so is an overabundance of resources. It's important to understand hospital beds from the perspective of the bed as a resource. However, the data is often stored from the perspective of a patient. How can we take data that captures when patients are in beds and determine the bed usage?

> **Note**: To complete the tasks in these tutorials, you need Tableau Prep and optionally Tableau Desktop installed:
>
> To install Tableau Prep and Tableau Desktop see Install Tableau Prep and Install Tableau Desktop in the Tableau Desktop and Tableau Prep Deployment guide. Otherwise you can download the Tableau Prep and Tableau Desktop free trials.
>
> You will also need to download three data files. It is recommended to save them in your

My Tableau Prep Repository > Datasources folder.
- Beds.xlsx
- Hours.xlsx
- Patient Beds.xlsx

In this article

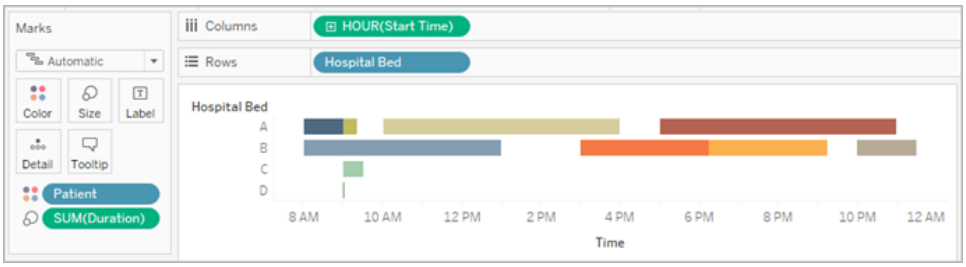
The Data

For our four beds, A, B, C, and D, we track which patient was in the bed and their start and end time there. The data looks like this:

	A	B	C	D
1	**Hospital Bed**	**Patient**	**Start Time**	**End Time**
2	A	Person 1	1/1/2018 8:34	1/1/2018 9:34
3	A	Person 5	1/1/2018 9:55	1/1/2018 10:15
4	A	Person 9	1/1/2018 10:34	1/1/2018 16:34
5	A	Person 8	1/1/2018 17:00	1/1/2018 23:00
6	B	Person 2	1/1/2018 8:45	1/1/2018 13:45
7	B	Person 6	1/1/2018 15:13	1/1/2018 18:27
8	B	Person 7	1/1/2018 18:41	1/1/2018 21:56
9	B	Person 10	1/1/2018 22:13	1/1/2018 23:43
10	C	Person 3	1/1/2018 9:05	1/1/2018 9:35
11	D	Person 4	1/1/2018 9:30	

Preliminary Analysis

If we bring this data into Tableau Desktop, we can create a Gantt chart to show when patients are in beds.



This is a useful visual. We can see that there are only small gaps in use for beds A and B, but bed C is very under-used. Bed D's patient has no end time, but we could address that with some calculations. This gives us a visual overview of how the beds are used.

However, what if we wanted to count the hours when a bed was empty? Or compare open bed time before and after a new policy is put in place? There's no easy way to do that with the data as it's currently structured.

Desired Data Structure

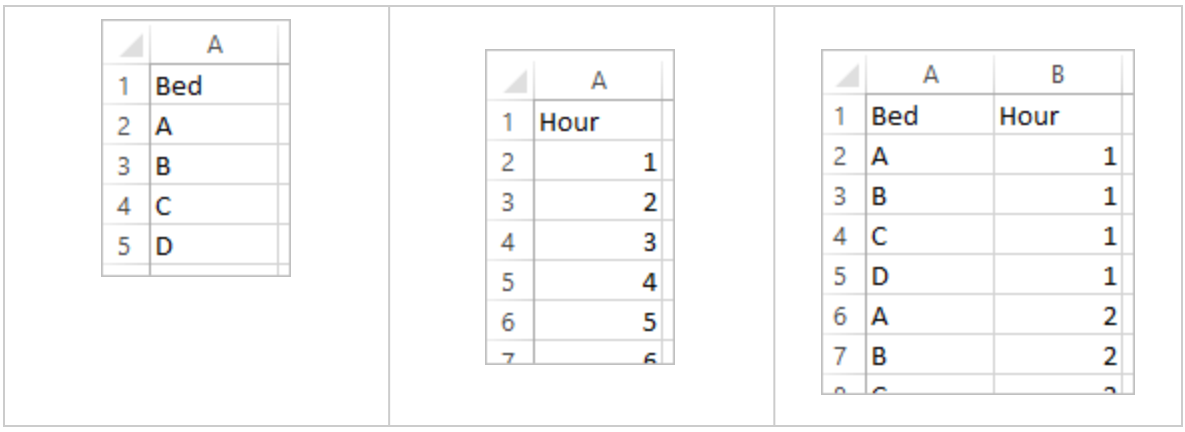
By creating some very basic data sets and combining them in Tableau Prep, we can modify this data set into a form that will allow us to perform deeper analysis and create even more useful visualizations.

Before we jump into Tableau Prep, let's step back and think about what we need to create to answer the question, "How many hours was each bed empty?"

We need to be able to look at each bed for each hour, and know whether or not there was a patient in the bed. Right now, the data is solely when a patient was in the bed; we haven't given Tableau information about the *empty* hours.

To create that full matrix of all beds and all hours, we'll create two new data sets. One is simply a list of beds (A, B, C, D) and the other is a list of hours (1, 2, 3, …, 23, 24). By performing a cross join (joining every row in one data set with every row in the other data set) we'll wind up with every possible combination of beds and hours.

The **Beds.xlsx** data set looks like this:	The **Hours.xlsx** data set looks like this:	And the cross joined results look like this:



	A
1	Bed
2	A
3	B
4	C
5	D

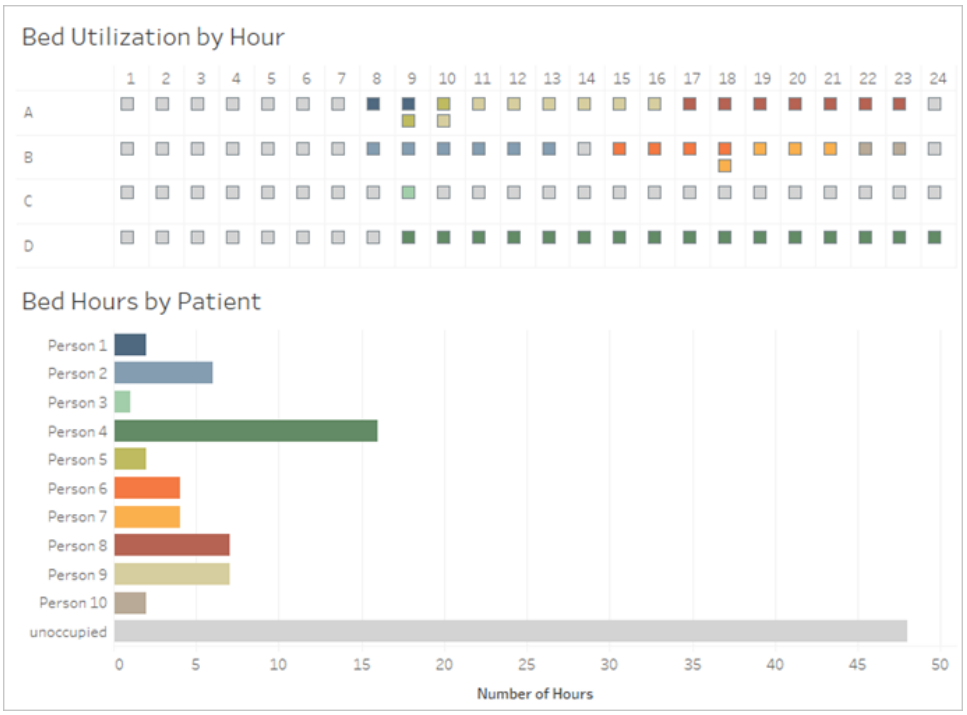
	A
1	Hour
2	1
3	2
4	3
5	4
6	5

	A	B
1	Bed	Hour
2	A	1
3	B	1
4	C	1
5	D	1
6	A	2
7	B	2

Next, we'll bring in the **Patient Beds** information, labeling each bed-hour combination as having a specific patient or not. We wind up with a data set that has a row for each bed-hour, and if a patient was in the bed, their number and start and end times. Null values indicate the bed was unoccupied.

	A	B	C	D	E
1	Bed	Hour	Patient	Start Time	End Time
29	D	7			
30	A	8	Person 1	1/1/2018 8:34	1/1/2018 9:34
31	B	8	Person 2	1/1/2018 8:45	1/1/2018 13:45
32	C	8			
33	D	8			
34	A	9	Person 5	1/1/2018 9:55	1/1/2018 10:15
35	A	9	Person 1	1/1/2018 8:34	1/1/2018 9:34
36	B	9	Person 2	1/1/2018 8:45	1/1/2018 13:45
37	C	9	Person 3	1/1/2018 9:05	1/1/2018 9:35
38	D	9	Person 4	1/1/2018 9:30	
39	A	10	Person 9	1/1/2018 10:34	1/1/2018 16:34
40	A	10	Person 5	1/1/2018 9:55	1/1/2018 10:15
41	B	10	Person 2	1/1/2018 8:45	1/1/2018 13:45
42	C	10			
43	D	10	Person 4	1/1/2018 9:30	
44	A	11	Person 9	1/1/2018 10:34	1/1/2018 16:34

With the data in this structure, we can perform analyses like this, which enables us to investigate unoccupied beds as easily as patient beds.



Restructuring the Data

So how do we get there with Tableau Prep? We'll build out the flow in two parts, first building the Bed Hours matrix, then combining it with the Patient Beds data. Make sure you've downloaded all three Excel files (**Beds.xlsx**, **Hours.xlsx**, and **Patient Beds.xlsx**) to follow along.

Bed Hour Matrix

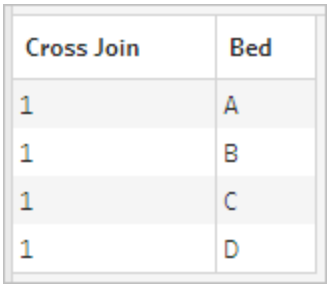
First, we'll connect to the **Beds.xlsx** file.

1. Open Tableau Prep.
2. From the start screen, click **Connect to Data**.
3. On the **Connections** pane, click **Microsoft Excel**. Navigate to where you saved **Beds.xlsx** and click **Open**.
4. The **Beds** sheet should automatically be brought out to the **Flow** pane.

Tip: For more information about connecting to data, see **Connect to Data** on page 71.

Next, we need to create a field we can use to do the cross join with the **Hours** data set. We'll add a calculation that is simply the value **1**.

5. In the **Flow** pane, select **Beds**, click the plus ⊕ icon, and select **Add Step**.
6. With the **Clean** step we just added, the **Profile** pane will come up. Click **Create Calculated Field** in the toolbar.
7. Name the field **Cross Join** and enter the value **1**.
8. The **Data** grid should update show the current state of the data.



Cross Join	Bed
1	A
1	B
1	C
1	D

Now we'll repeat the process with the Hours data set.

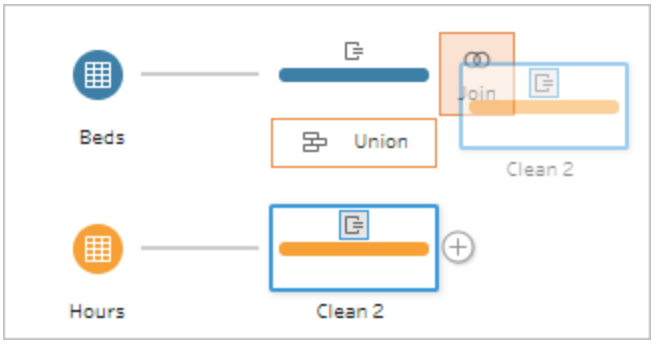
Click for directions

9. On the **Connections** pane, click the **Add connection** ⊕ button to add another data connection.
10. Choose **Microsoft Excel** and then select the **Hours.xlsx** file and click **Open**.
11. In the **Flow** pane, select **Hours**, click the plus ⊕ icon, and select **Add Step**.
12. From the toolbar in the **Profile** pane, create a calculated field named **Cross Join** and enter the value **1**.

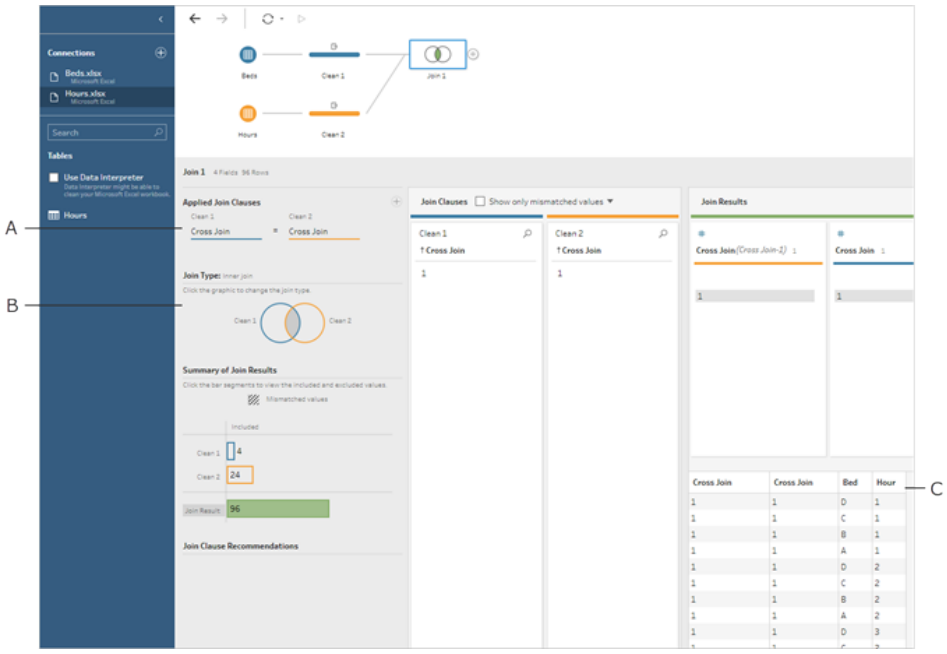
Cross Join	Hour
1	1
1	2
1	3
1	4
1	5

Both data sets now have a shared field, **Cross Join**, and can be joined.

13. Join the two cleaning steps by dragging **Clean 2** onto **Clean 1** and dropping it on the **Join** option.



14. In the **Join Profile** below, the join configurations should populate automatically.
    - Because we named both fields **Cross Join**, Tableau Prep automatically identifies them as the shared field and creates the appropriate **Applied Join Clauses**.
    - The default **Join Type** is inner, which is what we want.
    - This join will match all rows from **Beds** with all rows from **Hours**, as seen in the **Data** grid.



A. Join clause, B. Join type, C. Data grid results

Tip: For more information about joins, see **Join your data** on page 124.

We no longer need the **Cross Join** fields, so we can remove them.

15. In the **Flow** pane, select **Join 1**, click the plus icon, and select **Add Step**.
16. Select the fields **Cross Join-1** and **Cross Join**, and click **Remove Fields**.
17. Double click on the **Clean 3** label and rename that step **Bed Hour Matrix**.

We now have the Bed Hour Matrix data set that contains all beds and all hours and have finished the first part of building our data set.

Patient Bed Use

Part two is bringing in the patient bed usage. To start, we'll connect to the data.

1. On the **Connections** pane, click the **Add connection** button to add another data connection.

2. Choose **Microsoft Excel** and then select the **Patient Beds.xlsx** file, and click **Open**.

3. In the **Flow** pane, select **Patient Beds**, click the plus ⊕ icon and select **Add Step**.

Because the Bed Hour Matrix file is based on *hour* but Patient Beds is based on *actual time*, we need to pull the hour out of the Patient Beds start and end times. Additionally, for the end time, we want to ensure that if a patent is still in the bed at the end of the day (midnight, hour 24) we indicate that the bed is occupied even though there's no end time in the data set. We'll add a calculated field in this new step.

4. In the toolbar, click **Create Calculated Field**.
5. Name the field **Start Hour**. For the calculation, enter `DATEPART('hour',[Start Time])`.

   This takes the hour of the start time and pulls it out. Therefore, "1/1/18 9:35 AM" becomes simply "9".
6. Create another calculated field named **End Hour**. For the calculation, enter `IFNULL (DATEPART('hour',[End Time]), 24)`.

   The `DATEPART` portion takes the hour of the end time. The `IFNULL` portion will assign an end time of 24 (midnight) to any missing end time.

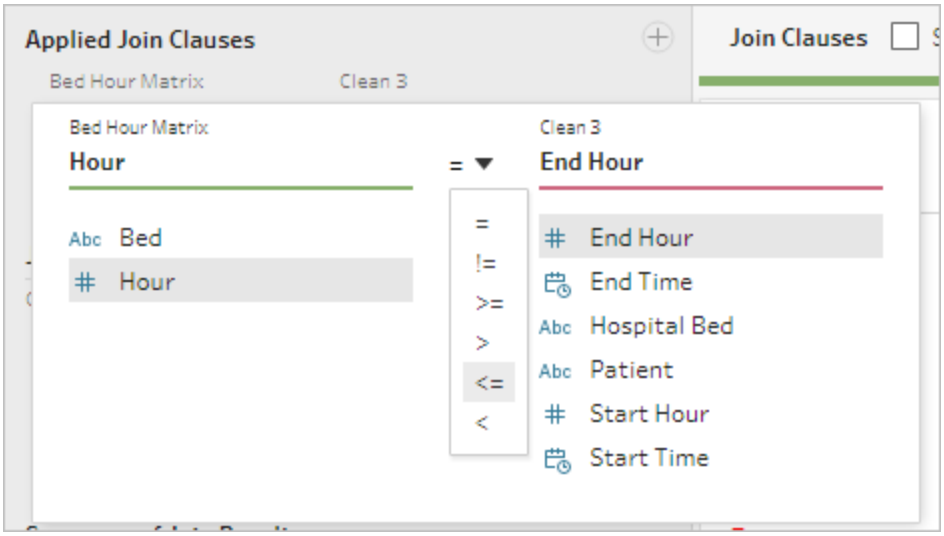
Now we're ready to join patient bed usage to the **Bed Hour Matrix**. This is a bit more complex join than we did previously. An inner join would only return values present in both data sets. Because we want to make sure we keep all the bed-hour slots, regardless of whether or not a patient was in the bed, we'll need to do a left join. This will result in a lot of nulls, but that's appropriate.

We also need to match when a bed-hour slot is taken by a patient (or patients). In addition to matching the bed the patient is in, we need to consider the time. The Bed Hour Matrix data set just has an **Hour** field, and the **Patient Beds** data set has **Start Hour** and **End Hour**. We'll use some basic logic to determine if a patient should be assigned to a given bed-hour slot: *A patient is considered in a bed if their start hour is less than or equal to (<=) the bed-hour slot AND their end hour is greater than or equal to (>=) the bed-hour slot.*

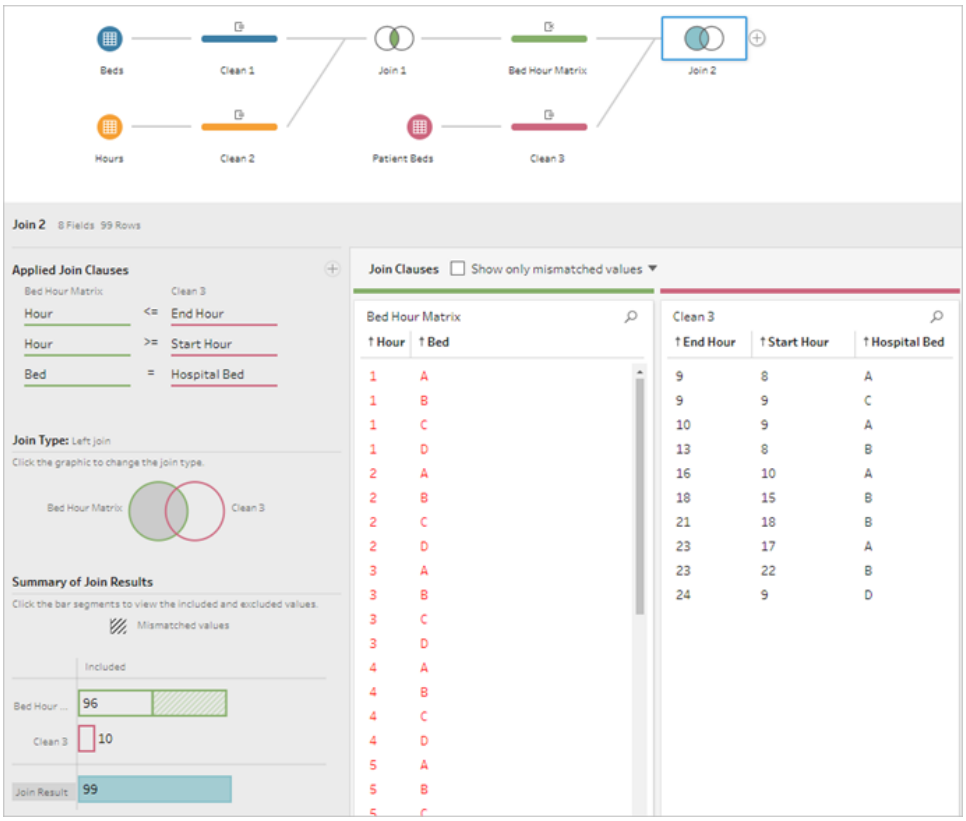
Therefore, three join clauses are needed to appropriately match these two data sets together.

9. Join the **Clean 3** step with the **Bed Hour Matrix** step.
10. In the **Applied Join Clauses** area, the default should be **Hour** = **End Hour**. Click the

join clause to change the operator from "**=**" to "**<=**".



11. Click the plus button in the upper right corner of the **Applied Join Clauses** area to add another join clause. Set it to be **Hour >= Start Hour**
12. Add a third join clause for **Bed = Hospital Bed**.
13. In the **Join Type** section, click the unshaded area of the graphic next to **Bed Hour Matrix** to change the join type to a **Left** join.



> **Note**: If you drag the **Bed Hour Matrix** to **Clean 3** instead of the other way around, the desired results can be obtained by using a right join instead of a left join. The order of dragging the steps matters for the orientation of the join. The join clauses will also be in reverse order—be sure to preserve the correct logic of comparing the hours.

Our data is now joined, but we should clean up some artifacts from the join and make sure the fields are tidy. We no longer need **Start Hour** and **End Hour**. **Hospital Bed** and **Bed** are also redundant. Finally, a value of null in the **Patient** field really means the bed is unoccupied.

14. In the **Flow** pane, add a cleaning step so we can tidy up the joined data.
15. Ctrl+click (Command+click on Mac) to multi select the fields **End Hour**, **Start Hour**, and **Hospital Bed**, then click **Remove Fields** in the toolbar.
16. On the **Patient** field profile card, double click the **null** value and type **Unoccupied**.

We now have a data structure with a row for every bed-hour; if there was a patient in bed during that hour, we have the patient information as well. All that remains to do is add an output step and generate the data set itself.

17. In the **Flow** pane, select **Clean 4**, click the plus ⊕ icon, and select **Add Output**.
18. In the **Output** pane, change the **Output type** to .csv then click **Browse**.
19. Enter **Bed Hour Patient Matrix** for the name and choose the desired location before clicking **Accept** to save.
20. Click the**Run Flow** ▷ button at the bottom of the pane to generate your output. Click **Done** in the status dialog to close the dialog.

Tip: For more information about outputs and running a flow, see **Save and Share Your Work** on page 136.

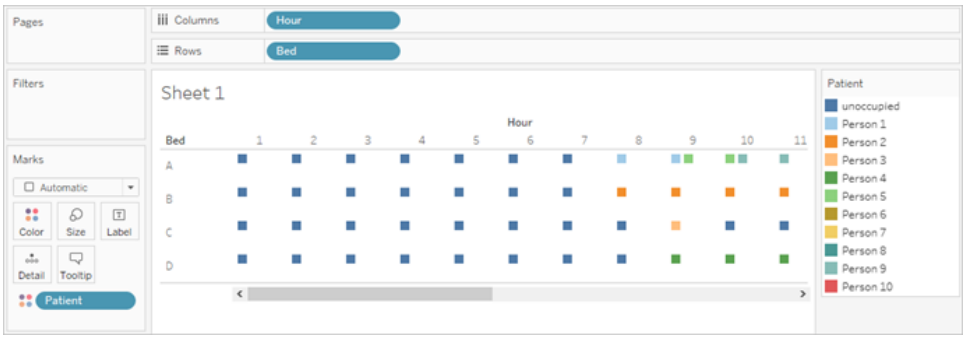
The final flow should look like this:

Analysis in Tableau Desktop

To install Tableau Desktop before continuing with this tutorial, you can download the free trial.

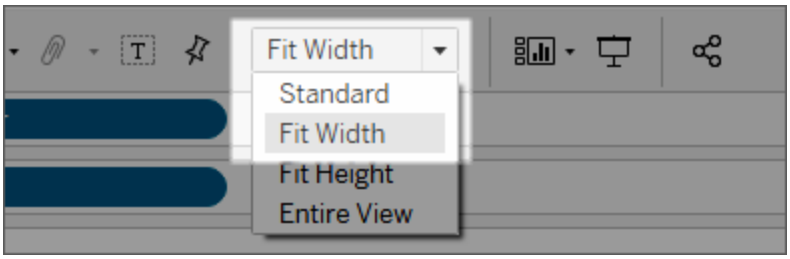
Now that we have the data set in the desired structure, we can perform deeper analysis than with the original data.

1. Open Tableau Desktop. In the **Connect** pane, select **Text file**, navigate to the **Bed Hour Patient Matrix.csv** file, and click **Open**.
2. On the **Data source** tab, the data should appear in the canvas by default. Click to **Sheet 1**.
3. In the **Data** pane, drag **Hour** from **Measures** to **Dimensions** to make it a discrete dimension.
4. Drag **Bed** to the **Rows** shelf and **Hour** to the **Columns** shelf.
5. Drag **Patient** to the **Color** shelf.

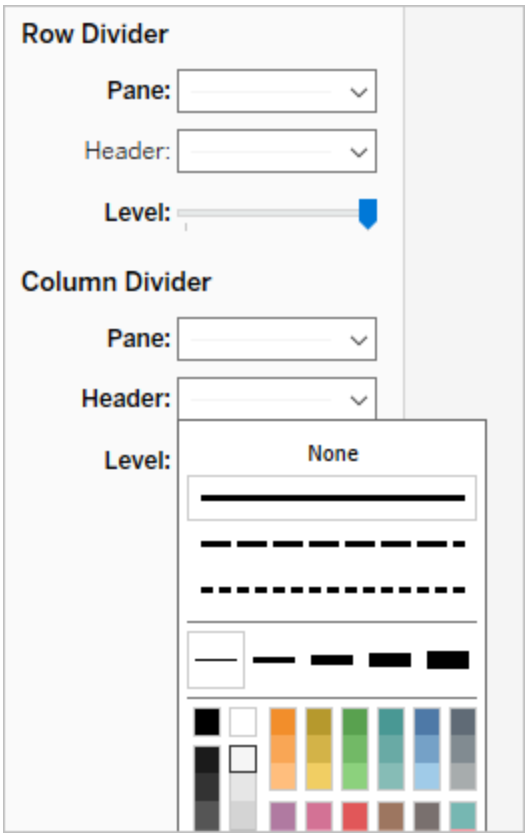


Formatting is optional, but may help make the visual more readable.

6. Click on the **Color** shelf and select **Edit Colors**.
7. In the area to the left, select **Unoccupied**. From the drop down on the right, choose the **Seattle Grays** color palette.
8. Select the fourth, lightest gray, and click **OK**.
9. Click the **Color** shelf again, then click the **Border** dropdown. Choose the second gray option at the far right.
10. In the toolbar, from the Size dropdown, change from **Standard** to **Fit Width**.

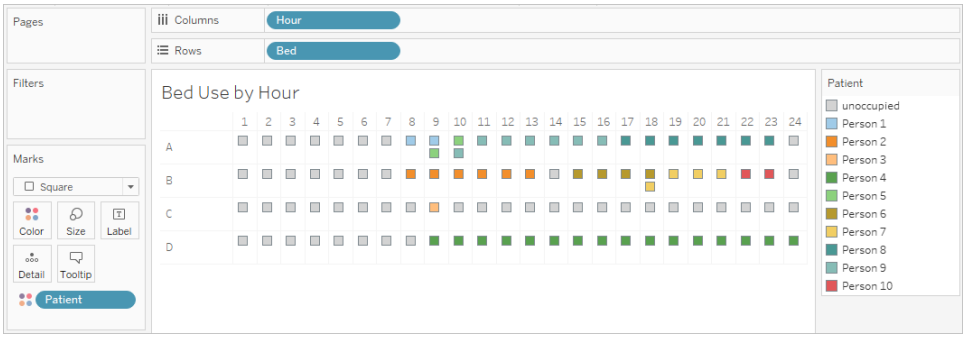


11. Click the **Format** menu and then **Borders**.
12. For **Row Divider**, click the Pane dropdown and choose a very light gray.
13. Adjust the **Level** slider to the second tick mark.
14. Repeat with the **Column Divider**. Set the **Pane** color to be light gray and the **Level** to the second tick mark.



15. Double click the sheet tab at the bottom and rename it **Bed Use by Hour**.

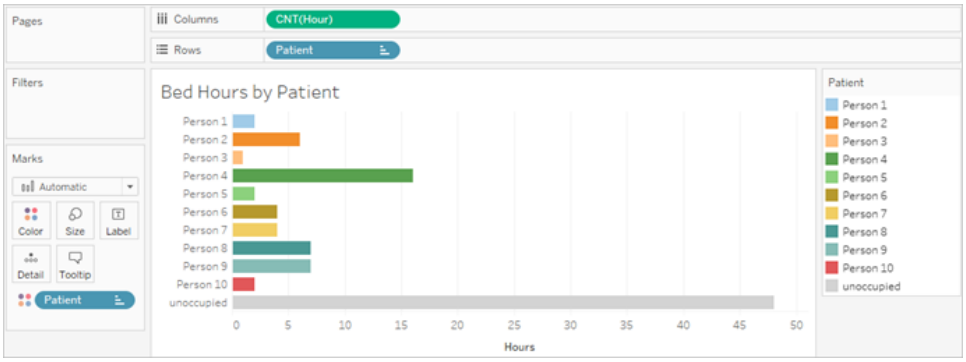
This view lets us quickly see when a given bed was occupied or open.



But we can go further and count the number of hours each bed was unoccupied.

16. Click the new sheet tab  icon at the bottom to open a clean sheet.
17. Drag **Patient** to **Rows**.

18. Drag **Hour** to **Columns**. Right click to open the menu. Choose **Measure** > **Count**.
19. Drag another copy of the **Patient** field from the **Data** pane to the **Color** shelf.
20. Right click on the axis and select **Edit Axis**. Change the title to **Hours** and close the dialog.
21. Rename the sheet tab **Bed Hours by Patient.**



This view lets us identify how many unoccupied bed hours we had, something we couldn't do with the original data set. What other charts or dashboards can you create? Give it a try now that your data is in the right structure.

Recap and Resources

To build this data structure using Tableau Prep, we needed to perform the following actions:

1. Build a data set for each aspect we want to analyze, in this case, **Beds** and **Hours**.
2. Cross join those data sets to create a **Bed Hour Matrix** data set with every possible combination of beds and hours.
3. Join the **Bed Hour Matrix** with the **Patient Bed** data, making sure the join keeps all bed-slot hours and the join clauses appropriately match patient bed data with the bed-hour slots.

We used the following calculations to create fields we could join on. The second and third pull out the hour information from the original datetime fields.

- **Cross Join** = `1`
  - This simply assigns the value 1 to every row
- **Start Hour** = `DATEPART('hour',[Start Time])`

    - This takes the hour of the start time and pulls it out. Therefore, "1/1/18 9:35 AM" becomes simply "9".
- **End Hour** = `IFNULL(DATEPART('hour',[End Time]), 24)`
    - We could use `DATEPART('hour',[End Time])`, as we did for **Start Time**. This takes the hour of the end time and pulls it out. Therefore, "1/1/18 4:34 PM" becomes simply "4".
    - But we want to indicate that the patient bed that is still occupied (no end time) is in use, not empty. To do so, we'll assign an end time of 24 (midnight) to any missing end time using the `IFNULL` function. If the first argument `DATEPART('hour',[End Time])` is null, the calculation will return "24" instead.

Note: Want to check your work? Download the Tableau Prep packaged flow file (Hospital Beds.tflx) and the Tableau Desktop packaged workbook file (Hospital Beds.twbx).

Resources: Need more training? Check out the new training videos for Tableau Prep or take an in-person training course. Curious about the features we covered? Check out the other topics in the Tableau Prep online help. Looking for additional resources? The Master Tableau Prep with this list of learning resources blog post is for you.

Finding the Second Date with Tableau Prep

A common need in analytics is to determine the date a *second* event happens, such as when a customer made a second purchase—thereby becoming a repeat customer—or when a driver gets a second traffic violation. Finding the date of a first event is easy, it's simply the minimum date. Finding the second date is trickier.

In this two-part tutorial, we'll shape traffic infraction data and answer the following questions:

1. What was the length of time in days between the first and second infraction for each driver?
2. Compare the fine amounts for the first and second infractions. Are they correlated?
3. Which driver paid the most overall? Who paid the least?

4. How many drivers had multiple infraction types?
5. What was the average fine amount for drivers who never attended traffic school?

In the first stage, we'll use Tableau Prep to restructure the data for our analysis. In the second stage, **Analysis with the Second Date in Tableau Desktop** on page 177, we'll move on to analysis in Tableau Desktop.

The goal of this tutorial is to present various concepts in the context of a real-life scenario and work through options—not prescriptively establishing which is best. At the end, you should have a better sense of how data structure impacts calculations and analysis, as well as greater familiarity with various aspects of Tableau Prep and calculations in Tableau Desktop.

Note: To complete the tasks in this tutorial, you need Tableau Prep and optionally Tableau Desktop installed and the data downloaded.

To install Tableau Prep and Tableau Desktop before continuing with this tutorial, see Install Tableau Prep and Install Tableau Desktop in the Tableau Desktop and Tableau Prep Deployment guide. Otherwise you can download the Tableau Prep and Tableau Desktop free trials.

The data set is Traffic Violations.xlsx. It is recommended to save it in your My Tableau Prep Repository > Datasources folder.

In this article

The Data

For this example, we're looking at traffic infraction data. Each infraction is a row. The driver, date, type of infraction, if the driver was required to attend traffic school, and fine amount are recorded.

	A	B	C	D	E
1	Driver ID	Infraction Date	Infraction Type	Traffic School	Fine Amount
2	JO-151451402	1/8/2017	Speeding	Yes	115
3	CM-127151402	3/1/2017	Running a red light	No	55
4	AP-109151404	3/2/2017	Non-moving violation	No	95
5	SH-199751404	3/4/2017	Speeding	Yes	130
6	BT-114401404	3/20/2017	Non-moving violation	No	130
7	MO-175001406	5/30/2017	Speeding	Yes	118
8	RA-1988558	6/2/2017	Speeding	Yes	144
9	BT-1168027	6/5/2017	Speeding	Yes	128
10	MO-175001406	6/18/2017	Speeding	Yes	115
11	MP-174701406	6/19/2017	Speeding	No	125
12	AA-106451404	7/5/2017	Running a red light	No	60
13	RA-199151402	7/20/2017	Speeding	Yes	146
14	SC-202601404	8/31/2017	Running a red light	No	150
15	MO-175001406	9/7/2017	Non-moving violation	No	320
16	AS-109451404	9/26/2017	Running a red light	No	50

Desired Data Structure

The data is currently structured such that each *infraction* is a row. A driver with multiple infractions appears on multiple rows, and there's no easy way to tell which was their first or second infraction.

To investigate our repeat offenders, we want a data set that separates out the first and second infraction dates, and the information associated with each of those infractions, and each row is a *driver*.

	A	B	C	D	E	F	G	H	I
1	Driver ID	1st Infraction Date	1st Infraction Type	1st Traffic School	1st Fine Amount	2nd Infraction Date	2nd Infraction Type	2nd Traffic School	2nd Fine Amount
2	BD-117701406	12/25/2017	Speeding	Yes	140	2/7/2018	Speeding	Yes	125
3	JO-151451402	1/8/2017	Speeding	Yes	115	11/21/2018	Reckless driving	Yes	550
4	SN-207101402	12/27/2017	Speeding	Yes	280	4/26/2018	Speeding	Yes	130
5	CJ-120101402	11/26/2017	Speeding	Yes	122	3/28/2018	Speeding	Yes	116
6	JR-156701404	12/24/2017	Speeding	No	148	7/28/2018	Speeding	Yes	310
7	AP-109151404	3/2/2017	Non-moving violation	No	95	9/24/2018	Speeding	No	105
8	PC-187451406	11/11/2017	Speeding	Yes	220	12/30/2018	Non-moving violation	No	600
9	TS-214301406	9/13/2018	Speeding	Yes	115	11/10/2018	Non-moving violation	No	95
10	NP-187001404	12/11/2018	Non-moving violation	No	80	12/20/2018	Speeding	No	120
11	DB-129701402	5/13/2018	Running a red light	No	110	11/11/2018	Speeding	Yes	80
12	AJ-107951404	10/15/2017	Speeding	Yes	130	12/31/2017	Running a red light	No	85
13	BT-114401404	3/20/2017	Non-moving violation	No	130	11/13/2018	Speeding	Yes	96
14	AF-108851406	5/9/2018	Non-moving violation	No	200	9/2/2018	Speeding	No	130
15	SC-202601404	8/31/2017	Running a red light	No	150	11/10/2018	Speeding	Yes	50
16	KL-166451406	10/4/2017	Speeding	No	115	11/13/2017	Speeding	Yes	104
17	MO-175001406	5/30/2017	Speeding	Yes	118	6/18/2017	Speeding	Yes	115
18	CM-127151402	3/1/2017	Running a red light	No	55	8/1/2018	Running a red light	No	160
19	KT-164801402	5/31/2018	Non-moving violation	No	190	11/10/2018	Speeding	No	74
20	JB-160001402	11/18/2018	Speeding	Yes	220	12/5/2018	Non-moving violation	No	195
21	LH-170201404	5/6/2018	Running a red light	No	110	9/17/2018	Speeding	Yes	230
22	BG-1103555	12/25/2017	Speeding	Yes	195	12/8/2018	Speeding	Yes	315
23	MP-174701406	6/19/2017	Speeding	No	125	10/12/2017	Running a red light	No	175
24	MK-179051406	10/23/2017	Reckless driving	Yes	890	9/8/2018	Speeding	Yes	134

Restructuring the Data

So how do we get there with Tableau Prep? We'll build out the flow in stages, beginning with pulling out the first infraction date, then the second, then shaping the final data set as desired. Make sure you've downloaded the Excel file (Traffic Violations.xlsx) to follow along.

Initial Aggregation for 1st Infraction Date

First, we'll connect to the **Traffic Violations.xlsx** file.

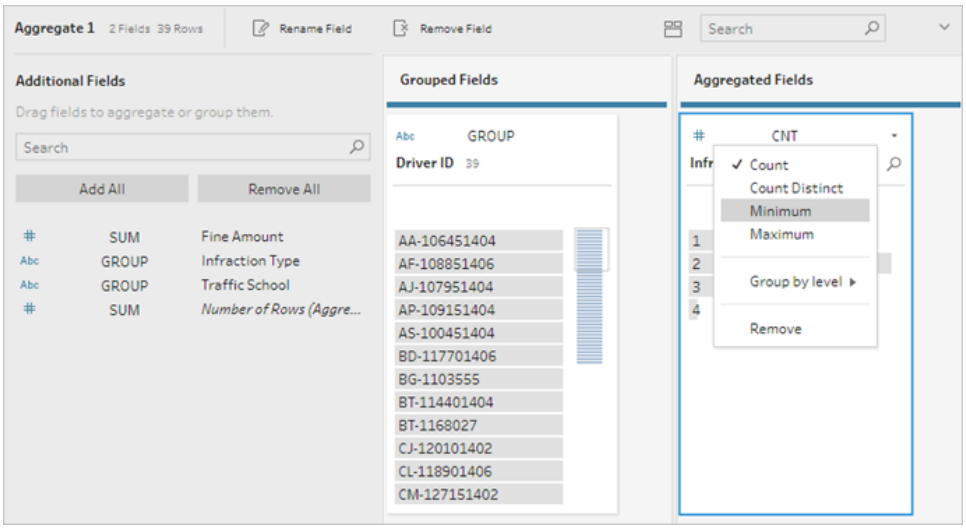
1. Open Tableau Prep.
2. From the start screen, click **Connect to Data**.
3. In the **Connections** pane, click **Microsoft Excel**. Navigate to where you saved **Traffic Violations.xlsx** and click **Open**.
4. The **Infractions** sheet should automatically be brought out to the **Flow** Pane.

> **Tip**: For more information about connecting to data, see **Connect to Data** on page 71.

Next, we need to identify the first infraction date per driver. We'll use an **Aggregate** step to do this, creating a mini data set of **Driver ID** and **Minimum Infraction Date**.

> **Note**: When using an **Aggregate** step in Tableau Prep, any field that should define what makes a row is a **Grouped Field**. (For us, that's **Driver ID**.) Any field that will be aggregated and presented at the level of the grouped fields is an **Aggregated Field**. (For us, that's **Infraction Date**).

5. In the **Flow** pane, select **Infractions**, click the plus icon, and select **Add Aggregate**.
6. Drag **Driver ID** to the **Grouped Fields** drop area.
7. Drag **Infraction Date** to the **Aggregated Fields** area. The default aggregation is **CNT** (count). Click **CNT** and change the aggregation to **Minimum**.

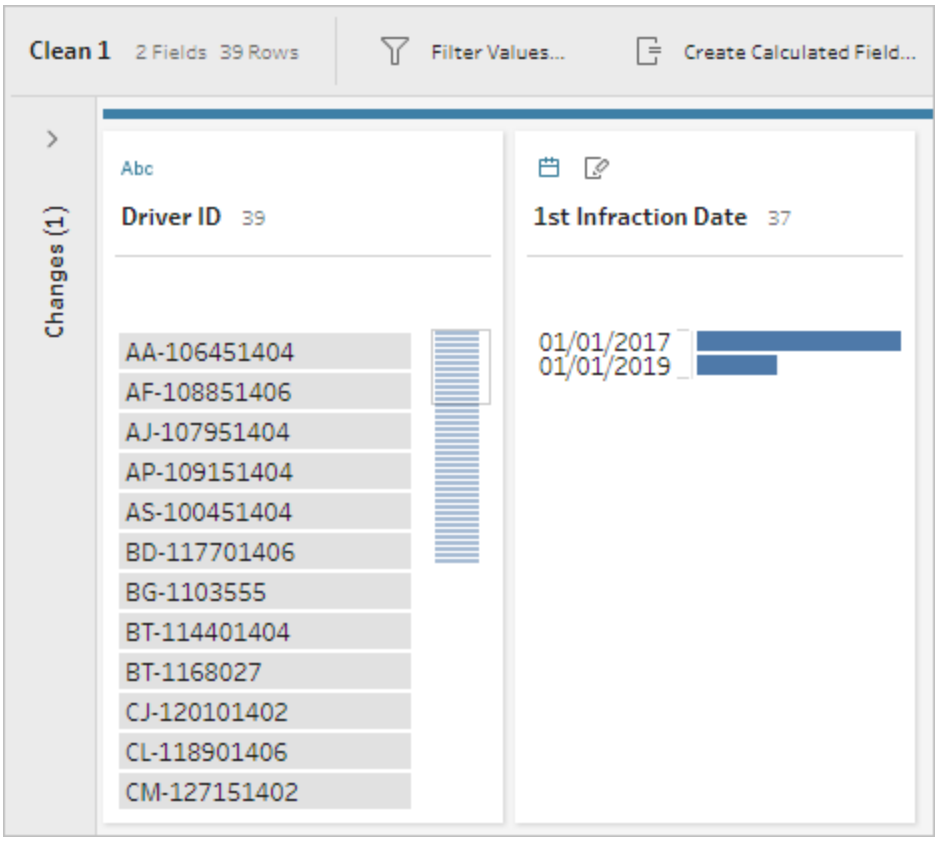


This identifies the smallest (earliest) date, which is the first infraction date per driver.

Tip: For more information about aggregations, see **Aggregate and group values** on page 105.

8. In the **Flow** pane, select **Aggregate 1**, click the plus ⊕ icon, and select **Add Step** so we can clean up the output of the aggregation.
9. In the **Profile** pane, double-click on the field name **Infraction Date** and change it to **1st Infraction Date**.

At this stage, the flow and profile should look like this:

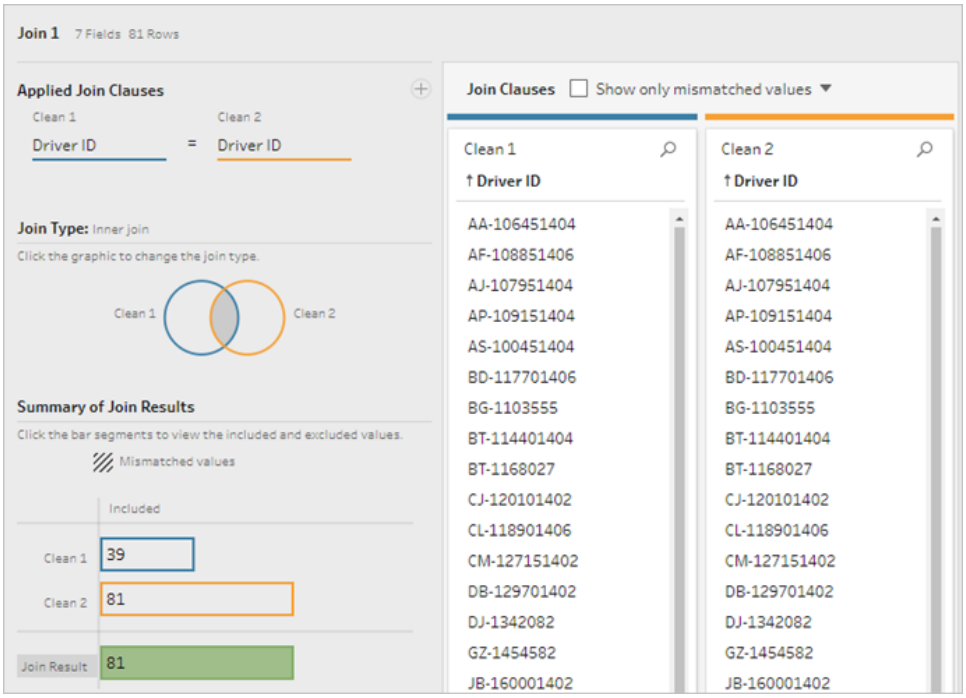


From the **Profile** pane in this **Clean** step, we can see that our data now consists of 39 rows and only 2 fields. Any field not used for grouping or aggregation is lost. But we want to be able to keep some of the original information. We could either add those fields to the grouping or aggregation (but doing so would change the level of detail or require the fields to be aggregated), or join this mini data set back to the original (essentially adding a new column to the original data for **1st Infraction Date**). Let's do the join.

10. In the **Flow** pane, select **Infractions**, click the plus ⊕ icon, and select **Add Branch**.

This branch has all the original data. We'll join the results of the aggregation to this copy of the full data. By joining on **Driver ID**, we're adding the minimum date from our aggregation into the original data.

11. Select step **Clean 2** and drag it on top of step **Clean 1**, and drop it on **Join**.
12. The default join configuration should be correct: an inner join on **Driver ID = Driver ID**.

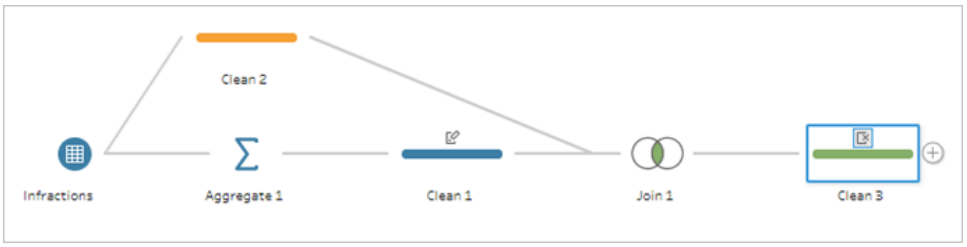


Tip: For more information about joins, see **Join your data** on page 124.

Because some fields may be duplicated during a join, such as the fields in the join clause, it's often a good idea to clean up extraneous fields after performing a join.

13. In the **Flow** pane, select **Join 1**, click the plus ⊕ icon, and choose **Add a Step**.
14. In the **Profile** pane, click the card for **Driver ID-1**, then click **Remove Field** in the toolbar.
15. To change the field order, drag the **1st Infraction Date** card between **Driver ID** and **Infraction Date** where you see the black line appear.

At this stage, the flow should look like this:



Looking at the data grid below, we can see our new, combined data set. We have the minimum—that is, first—infraction date for each driver added to each row in the data set.

Driver ID	1st Infraction Date	Infraction Date	Infraction Type	Traffic School	Fine Amount
JO-151451402	01/08/2017	01/08/2017	Speeding	Yes	115
CM-127151402	03/01/2017	03/01/2017	Running a red light	No	55
AP-109151404	03/02/2017	03/02/2017	Non-moving violation	No	95
SH-199751404	03/04/2017	03/04/2017	Speeding	Yes	130
BT-114401404	03/20/2017	03/20/2017	Non-moving violation	No	130
MO-175001406	05/30/2017	05/30/2017	Speeding	Yes	118
RA-1988558	06/02/2017	06/02/2017	Speeding	Yes	144
BT-1168027	06/05/2017	06/05/2017	Speeding	Yes	128
MO-175001406	05/30/2017	06/18/2017	Speeding	Yes	115
MP-174701406	06/19/2017	06/19/2017	Speeding	No	125
AA-106451404	07/05/2017	07/05/2017	Running a red light	No	60
RA-199151402	07/20/2017	07/20/2017	Speeding	Yes	146
SC-202601404	08/31/2017	08/31/2017	Running a red light	No	150
MO-175001406	05/30/2017	09/07/2017	Non-moving violation	No	320
AS-100451404	09/26/2017	09/26/2017	Running a red light	No	50
SH-199751404	03/04/2017	09/27/2017	Speeding	Yes	225
AA-106451404	07/05/2017	09/28/2017	Running a red light	No	195

Second Aggregation for 2nd Infraction Date

We need to also determine the second infraction date. To do this, we want to filter out any row where the infraction date is equal to the minimum—thus removing the first date. We can then take the minimum of the remaining dates using another aggregate step, leaving us with the second infraction date, which we'll rename for clarity.

> **Note**: Because we'll want to use the data as it currently is in **Clean 3** later on in the flow, we'll add another **Clean** step to break out the process of getting the second infraction date.

16. In the **Flow** pane, select **Clean 3**, click the plus ⊕ icon, and select **Add Step**.
17. On the toolbar in the **Profile** pane, choose **Filter Values**. Create a filter `[Infraction Date] != [1st Infraction Date]`.

18. Remove the field **1st Infraction Date**.

19. In the **Flow** pane, select **Clean 4**, click the plus ⊕ icon, and select **Add Aggregate**.

20. Drag **Driver ID** to the **Grouped Fields** drop area. Drag **Infraction Date** to the **Aggregated Fields** area and change the aggregation to **Minimum**.

21. In the **Flow** pane, select **Aggregate 2**, click the plus ⊕ icon, and select **Add Step**. Rename **Infraction Date** to **2nd Infraction Date**.

At this stage, the flow should look like this:

We now have our second infraction date identified for each driver. To get all the other information associated with each infraction (type, fine, traffic school) we again need to join this back to the entire data set.

22. Select **Clean 5** and drag it on top of **Clean 3**, dropping it on **Join**.

23. Again, the default join configuration should be correct: an inner join on **Driver ID = Driver ID**.

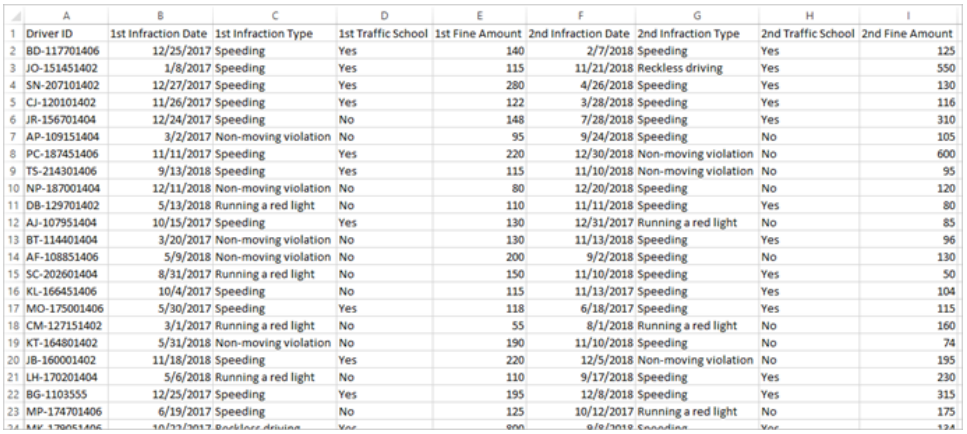
24. In the **Flow** pane, select **Join 2**, click the plus ⊕ icon, and select **Add Step**. Delete the fields **Driver ID-1** and **1st Infraction Date** as they are no longer needed.

At this stage, the flow should look like this:

Create full data sets for the 1st and 2nd infractions

Before we go any further, let's step back and think about everything we have and how we want to bring it all together. Our desired end state is a data set that looks like this, with a column for

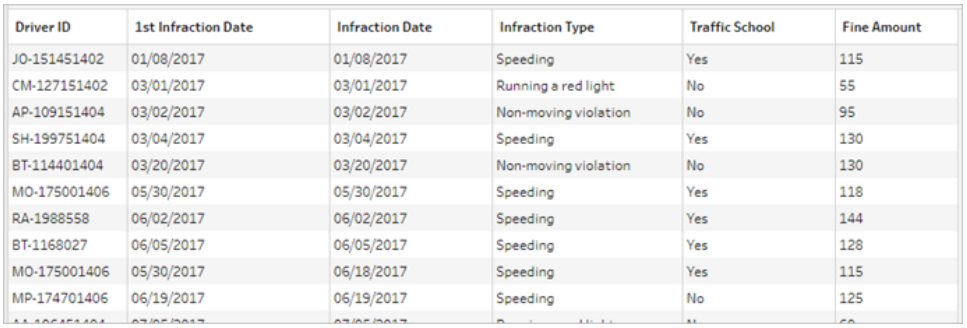
Driver ID, then columns for date, type, traffic school, and fine amount for the 1st and 2nd infractions.



	A	B	C	D	E	F	G	H	I
1	Driver ID	1st Infraction Date	1st Infraction Type	1st Traffic School	1st Fine Amount	2nd Infraction Date	2nd Infraction Type	2nd Traffic School	2nd Fine Amount
2	BD-117701406	12/25/2017	Speeding	Yes	140	2/7/2018	Speeding	Yes	125
3	JO-151451402	1/8/2017	Speeding	Yes	115	11/21/2018	Reckless driving	Yes	550
4	SN-207101402	12/27/2017	Speeding	Yes	280	4/26/2018	Speeding	Yes	130
5	CJ-120101402	11/26/2017	Speeding	Yes	122	3/28/2018	Speeding	Yes	116
6	JR-156701404	12/24/2017	Speeding	No	148	7/28/2018	Speeding	Yes	310
7	AP-109151404	3/2/2017	Non-moving violation	No	95	9/24/2018	Speeding	No	105
8	PC-187451406	11/11/2017	Speeding	Yes	220	12/30/2018	Non-moving violation	No	600
9	TS-214301406	9/13/2018	Speeding	Yes	115	11/10/2018	Non-moving violation	No	95
10	NP-187001404	12/11/2018	Non-moving violation	No	80	12/20/2018	Speeding	No	120
11	DB-129701402	5/13/2018	Running a red light	No	110	11/11/2018	Speeding	Yes	80
12	AJ-107951404	10/15/2017	Speeding	Yes	130	12/31/2017	Running a red light	No	85
13	BT-114401404	3/20/2017	Non-moving violation	No	130	11/13/2018	Speeding	Yes	96
14	AF-108851406	5/9/2018	Non-moving violation	No	200	9/2/2018	Speeding	No	130
15	SC-202601404	8/31/2017	Running a red light	No	150	11/10/2018	Speeding	Yes	50
16	KL-166451406	10/4/2017	Speeding	No	115	11/13/2017	Speeding	Yes	104
17	MO-175001406	5/30/2017	Speeding	Yes	118	6/18/2017	Speeding	Yes	115
18	CM-127151402	3/1/2017	Running a red light	No	55	8/1/2018	Running a red light	No	160
19	KT-164801402	5/31/2018	Non-moving violation	No	190	11/10/2018	Speeding	No	74
20	JB-160001402	11/18/2018	Speeding	Yes	220	12/5/2018	Non-moving violation	No	195
21	LH-170201404	5/6/2018	Running a red light	No	110	9/17/2018	Speeding	Yes	230
22	BG-1103555	12/25/2017	Speeding	Yes	195	12/8/2018	Speeding	Yes	315
23	MP-174701406	6/19/2017	Speeding	No	125	10/12/2017	Running a red light	No	175
24	MK-179951406	10/22/2017	Reckless driving	Yes	800	9/8/2018	Speeding	Yes	134

How do we get there from here?

In the step **Clean 3**, we have our compete data set with a column that repeats the first infraction date for each driver.



Driver ID	1st Infraction Date	Infraction Date	Infraction Type	Traffic School	Fine Amount
JO-151451402	01/08/2017	01/08/2017	Speeding	Yes	115
CM-127151402	03/01/2017	03/01/2017	Running a red light	No	55
AP-109151404	03/02/2017	03/02/2017	Non-moving violation	No	95
SH-199751404	03/04/2017	03/04/2017	Speeding	Yes	130
BT-114401404	03/20/2017	03/20/2017	Non-moving violation	No	130
MO-175001406	05/30/2017	05/30/2017	Speeding	Yes	118
RA-1988558	06/02/2017	06/02/2017	Speeding	Yes	144
BT-1168027	06/05/2017	06/05/2017	Speeding	Yes	128
MO-175001406	05/30/2017	06/18/2017	Speeding	Yes	115
MP-174701406	06/19/2017	06/19/2017	Speeding	No	125
AA-196451404	07/05/2017	07/05/2017	Running a red light	No	60

We want to eliminate all the rows for a driver that aren't the first infraction, building a data set of only first infractions. That is, we only want to keep the information for a given driver when **1st Infraction Date = Infraction Date**. Once we've filtered to keep only the row of the first infraction, we can remove the **Infraction Date** field and tidy up field names.

Similarly, after the second aggregation and join, we have our complete data set with a column for the second infraction date.

Driver ID	2nd Infraction Date	Infraction Date	Infraction Type	Traffic School	Fine Amount
JO-151451402	11/21/2018	01/08/2017	Speeding	Yes	115
CM-127151402	08/01/2018	03/01/2017	Running a red light	No	55
AP-109151404	09/24/2018	03/02/2017	Non-moving violation	No	95
SH-199751404	09/27/2017	03/04/2017	Speeding	Yes	130
BT-114401404	11/13/2018	03/20/2017	Non-moving violation	No	130
MO-175001406	06/18/2017	05/30/2017	Speeding	Yes	118
MO-175001406	06/18/2017	06/18/2017	Speeding	Yes	115
MP-174701406	10/12/2017	06/19/2017	Speeding	No	125
AA-106451404	09/28/2017	07/05/2017	Running a red light	No	60
RA-199151402	12/31/2017	07/20/2017	Speeding	Yes	146
SC-202601404	11/10/2018	08/31/2017	Running a red light	No	150

We can perform a similar filter of **2nd Infraction Date = Infraction Date** to keep only the row of information for each driver's 2nd infraction. Again, we can also remove the now-redundant **Infraction Date** and tidy up field names.

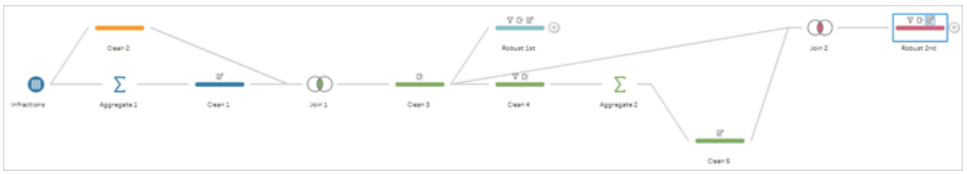
We'll start with the first infraction data set.

25. In the **Flow** pane, select **Clean 3**, click the plus ⊕ icon, and select **Add Branch**.
26. With this new **Clean** step selected, in the **Profile** pane, click **Filter Values** in the toolbar. Create a filter `[1st Infraction Date] = [Infraction Date]`.
27. Remove the field **Infraction Date**.
28. Rename the **Infraction Type**, **Traffic School**, and **Fine Amount** fields to start with "1st".
29. Double-click on the name **Clean 7** under the step in the **Flow** pane and rename it **Robust 1st**.

Now for the second infraction data set.

30. In the **Flow** pane, select **Clean 6**, after the last join.
31. Click **Filter Values** in the toolbar. Create a filter `[2nd Infraction Date] = [Infraction Date]`.
32. Remove the field **Infraction Date**.
33. Rename the **Infraction Type**, **Traffic School**, and **Fine Amount** fields to start with "2nd".
34. Double-click on the name **Clean 6** under the step in the **Flow** pane and rename it **Robust 2nd**.

At this stage, the flow should look like this:



Create the complete data set

Now that we have these two tidy data sets with complete information for the first and second infractions per driver, we can join them back together on **Driver ID** and wind up with our desired data structure.

35. Select **Robust 2nd** and drag it on top of **Robust 1st**, dropping it on **Join**.
36. The default join clause should be correct as **Driver ID = Driver ID**.
37. Because we don't want to drop drivers who didn't have a second infraction, we need to make this a left join. In the **Join Type** area, click the unshaded area of the diagram next to **Robust 1st**, turning it into a **Left** join.
38. In the **Flow** pane, select **Join 3**, click the plus ⊕ icon, and select **Add Step**. Remove the field duplicate**Driver ID-1**.

The data is in the desired state, so we can create an output and proceed to analysis.

39. In the **Flow** pane, select the newly added **Clean 6**, click the plus ⊕ icon, and select **Add Output**.
40. In the **Output** pane, change the **Output type** to .csv then click **Browse**. Enter **Driver Infractions** for the name and choose the desired location before clicking **Accept** to save.
41. Click the**Run Flow** ▷ button at the bottom of the pane to generate your output. Click **Done** in the status dialog to close the dialog.

> **Tip**: For more information about outputs and running a flow, see **Save and Share Your Work** on page 136.

The final flow should look like this:

Note: You can download the completed flow file to check your work: Driver Infractions.tflx

Recap

For the first stage of this tutorial, our goal was to take our original data set and prepare it for analysis involving the first and second infraction dates. The process consists of three phases:

Identify the first and second infraction dates:

1. Create an aggregation that keeps **Driver ID** and MIN **Infraction Date**. Join this with the original data set to create an "intermediate data set" that has the first (minimum) infraction date repeated for every row.
2. On a new step, filter out all rows where the **1st Infraction Date** is the same as the **Infraction Date**. From that filtered data set, create an aggregation that keeps **Driver ID** and MIN **Infraction date**. Join this with the intermediate data set from the first step. This identifies the second infraction date.

Build out clean data sets for the first and second infractions:

3. Go back and create a branch from the intermediate data set and filter to keep only rows where the **1st Infraction Date** is the same as the **Infraction Date**. This builds a data set for just the first infraction. Tidy it up by removing any unnecessary fields and rename all the desired fields (except **Driver ID**) to indicate they're for the first infraction. This is the **Robust 1st** data set.
4. Tidy the data set for the second infraction date. Clean the join results from step 2 by filtering to keep only rows where the **2nd Infraction Date** is the same as the **Infraction Date**. Remove any unnecessary fields and rename all the desired fields (except **Driver ID**) to indicate they're for the second infraction. This is the **Robust 2nd** data set.

Combine the first and second infraction data into one data set:

5. Join the **Robust 1st** and **Robust 2nd** data sets, making sure to keep all records from **Robust 1st** to prevent losing any drivers without a second infraction.

Next, we want to explore how this data can be analyzed in Tableau Desktop.

Continue to **Analysis with the Second Date in Tableau Desktop** below.

> **Note**: Special Thanks to Ann Jackson's Workout Wednesday topic Do Customers Spend More on Their First or Second Purchase? and Andy Kriebel's Tableau Prep Tip Returning the First and Second Purchase Dates that provided the initial inspiration for this tutorial. Clicking these links will take you away from the Tableau website. Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the owners if you have questions regarding their content.

Analysis with the Second Date in Tableau Desktop

This is the second stage of the tutorial and assumes the first stage, **Finding the Second Date with Tableau Prep** on page 164, has been completed.

In the first stage, we took our original data set and shaped it to answer the following questions:

1. What was the length of time in days between the first and second infraction for each driver?
2. Compare the fine amounts for the first and second infractions. Are they correlated?
3. Which driver paid the most overall? Who paid the least?
4. How many drivers had multiple infraction types?
5. What was the average fine amount for drivers who never attended traffic school?

As we now explore these questions, it becomes clear that there are some pros and cons to the first data structure we created. We'll go back into Tableau Prep and do some additional reshaping, then see how that impacts the same analysis in Tableau Desktop. Finally, we'll look at a Tableau Desktop-only approach to the analysis using Level of Detail (LOD) expressions with the original data.

The goal of this tutorial is to present various concepts in the context of a real-life scenario and work through options—not prescriptively establishing which is best. At the end, you should have a better sense of how data structure impacts calculations and analysis, as well as greater familiarity with various aspects of Tableau Prep and calculations in Tableau Desktop.

Note: To complete the tasks in this tutorial, you need Tableau Prep and optionally Tableau Desktop installed and the data downloaded.

To install Tableau Prep and Tableau Desktop before continuing with this tutorial, see Install Tableau Prep and Install Tableau Desktop in the Tableau Desktop and Tableau Prep Deployment guide. Otherwise you can download the Tableau Prep and Tableau Desktop free trials.

The data set is the output from Driver Infractions.tflx, as built in the first stage.

In this article

Analysis in Tableau Desktop

Now that we have our data configured, we'll bring it into Tableau Desktop. We can easily answer some questions, but others involve a few (or a lot of) calculations. Try your hand at the questions below; you can expand each one for basic information about how to proceed if you get stuck.

Note: You can download the workbook Driver Infractions.twbx to look at the solutions in context. Remember that there may be alternative ways to interpret the analysis or pursue answers.

1. What was the length of time in days between the first and

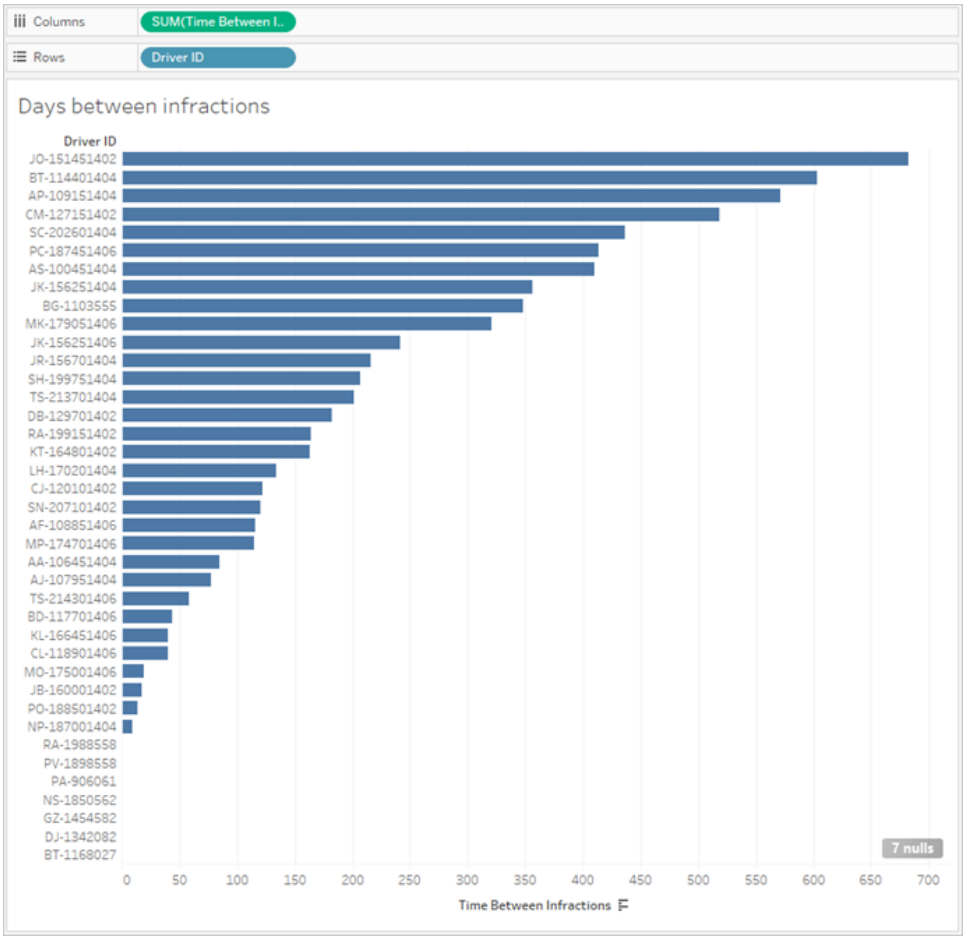
second infraction for each driver?

A. To answer this question in Tableau Desktop, we'll use the DATEDIFF function. This function takes three arguments—the date part, the start date, and the end date. Since we want to know the days between these events, we'll use the date part 'day'. Our start and end dates are in the data set as **1st Infraction Date** and **2nd Infraction Date**.

B. The calculation is:

   Time Between Infractions = `DATEDIFF('day', [1st Infraction Date], [2nd Infraction Date])`

C. We can plot that against **Driver ID** as a bar chart. Note that seven drivers had no second infraction, so there are seven nulls.

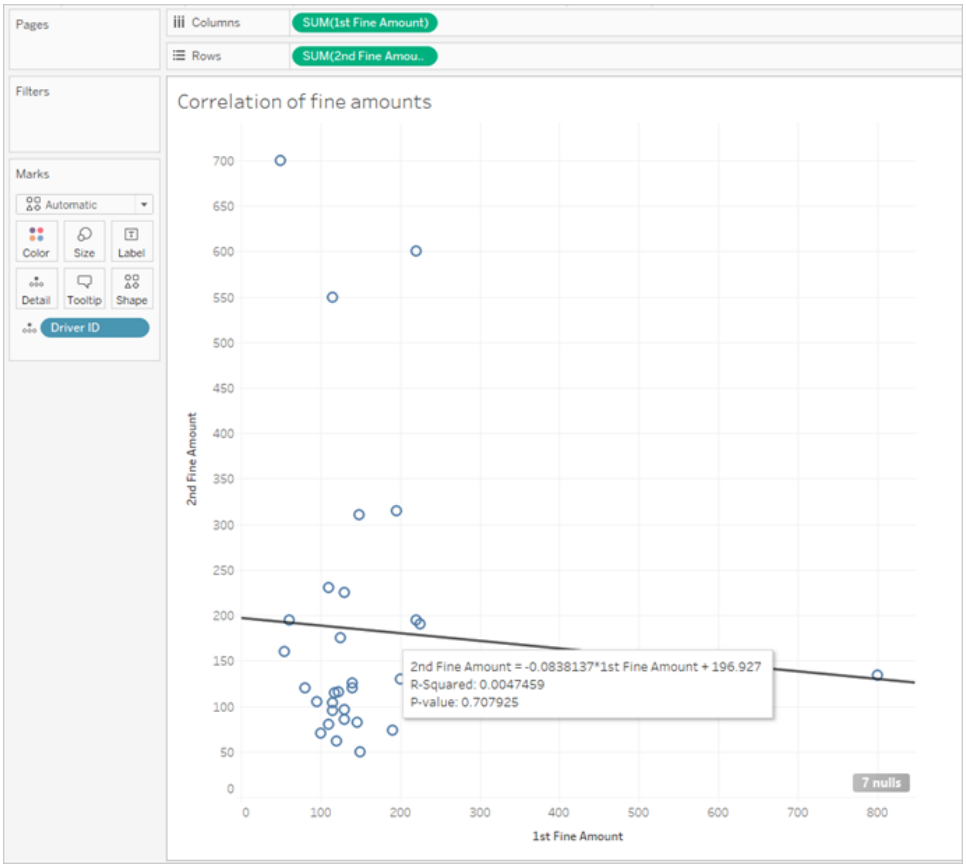


2. Compare the fine amounts for the first and second

infractions. Are they correlated?

A. To answer this question in Tableau Desktop, we'll create a scatter plot of **1st Fine Amount** and **2nd Fine Amount**. By bringing **Driver ID** to the **Detail** shelf on the **Marks** card, we can create a mark for each driver.

B. To add a trend line, use the **Analytics** tab in the left-hand pane and bring out a linear trend line. Hovering over the trend line, we can see the R-squared value is practically zero, and the p-value is far above any cutoff for significance. We can determine that there is no correlation between first and second fine amount.



3. Which driver paid the most overall? Who paid the least?

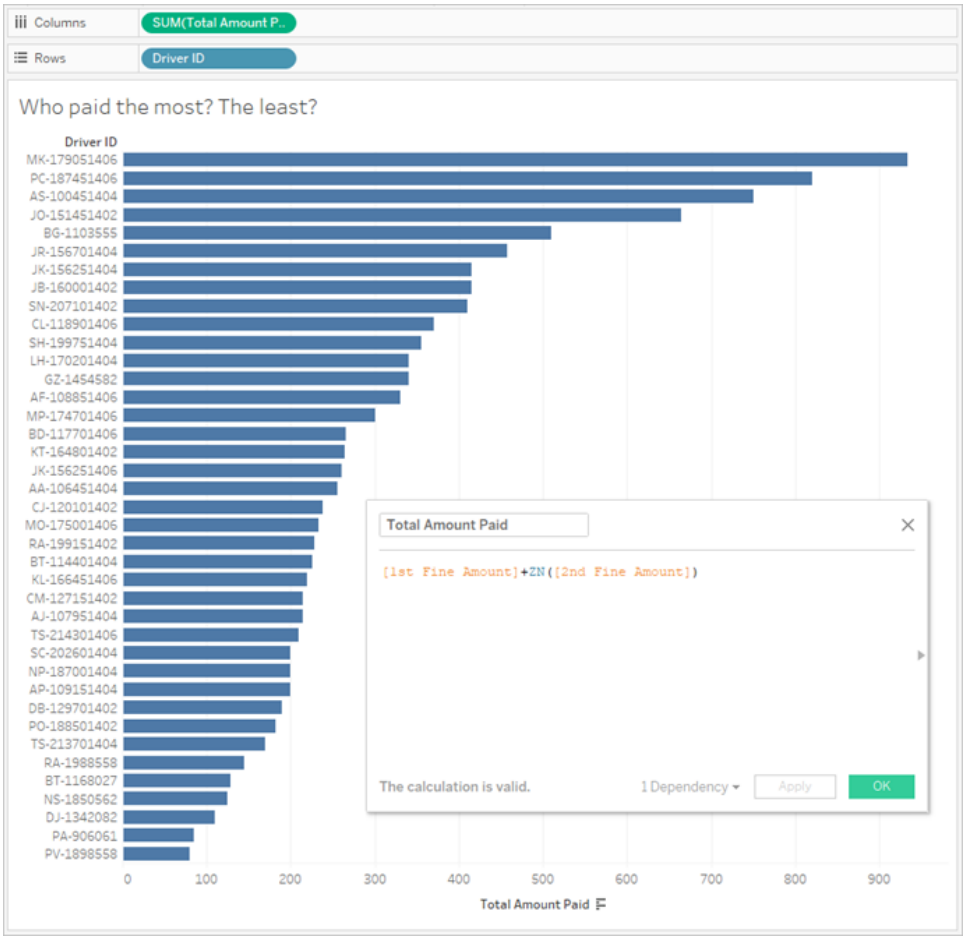
When we want to go deeper in our analysis, we may need to create some calculations.

A. To answer this in Tableau Desktop, we need to add the fines for both infractions into a single field. Because some drivers may not have had a second infraction, we need to use the zero null `ZN` function to turn any nulls for **2nd Fine Amount** into zeros.

B. The calculation is:

Total Amount Paid = `[1st Fine Amount] + ZN([2nd Fine Amount])`

C. We can plot **Total Amount Paid** against **Driver ID** and sort the bar chart.



4. How many drivers had multiple infraction types?

A. To answer this in Tableau Desktop, we need to do a fancier `IF` calculation, comparing if the first and second infraction types are the same. If they are, we want to assign the

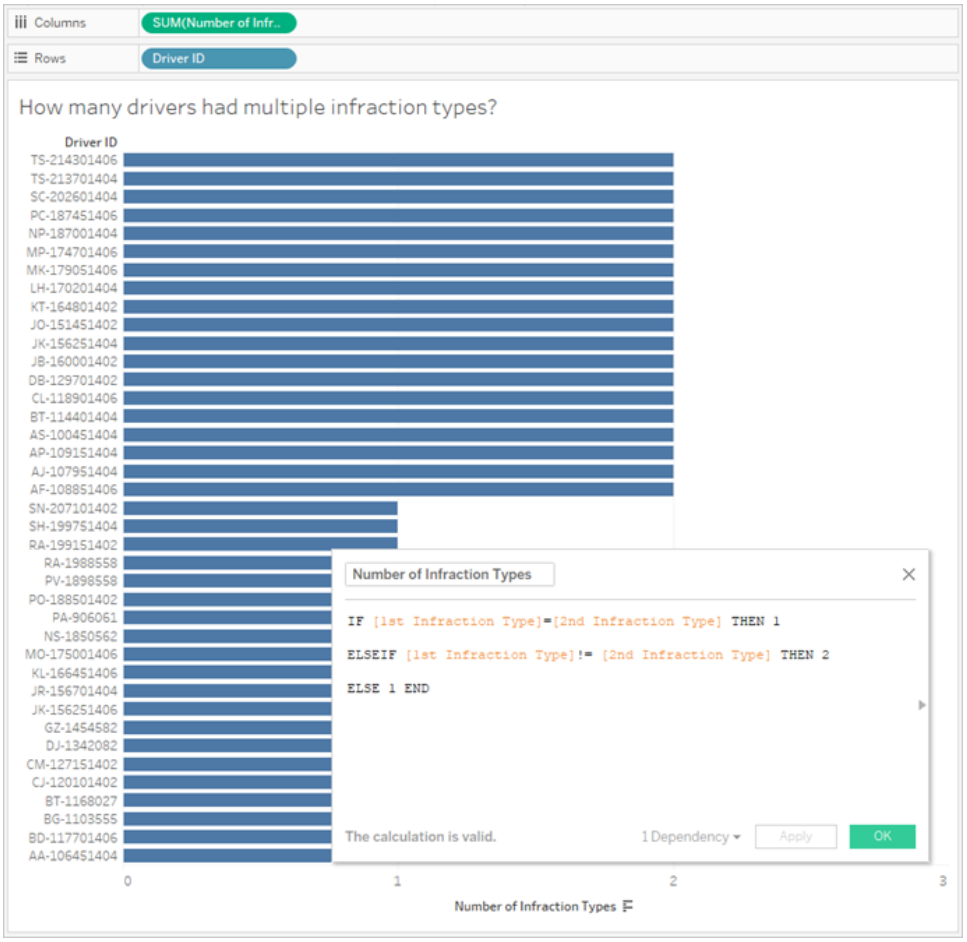
value "1". If they are not the same, we'll assign "2". Ay other result, such as a null second infraction type, will be assigned "1".

B. The calculation is:

Number of Infraction Types =

```
IF [1st Infraction Type]=[2nd Infraction Type] THEN 1
ELSEIF [1st Infraction Type]!= [2nd Infraction Type] THEN 2
ELSE 1 END
```

C. We can then plot **Number of Infraction Types** against **Driver ID** and sort the bar chart.



5. What was the average fine amount for drivers who never

attended traffic school?

A. To answer this in Tableau Desktop, we cannot simply divide the total fine amount by two, since some drivers only had one infraction. We also can't calculate the average fine per driver and take the average of those values, because averaging averages can lead to inconsistencies. Instead, we need to calculate the total amount paid by drivers who never attended traffic school, then divide by the total number of infractions associated with those fines.

    1. First, we need to determine if each driver had a second infraction. We can leverage the fact the information in all the "2nd" fields will be null if there was no second infraction and start building the calculation:

       ```
       IFNULL([2nd Infraction Type], 'no')
       ```

       This will return an infraction type if it exists, or "no" if there was no second infraction.

    2. Next, we need to turn this information into the number of infractions, 1 or 2. If the result of our `IFNULL` calculation is "no", then the driver should be marked as having one fine. Any other result should be marked as having two fines. The calculation is:

       Number of Infractions =

       ```
       IF IFNULL([2nd Infraction Type], 'no') = 'no' THEN 1
       ELSE 2
       END
       ```

    3. Now we need to consider the total fine amount. Similarly to question 3 above, we'll add the first and second fine amounts, with a `ZN` function around the second. However, because we want this to be computed at the level of the entire data set, it's a best practice to specify the aggregations, **SUM**, in the calculation itself. The calculation is:

       ```
       SUM([1st Fine Amount]) + SUM( ZN([2nd Fine Amount]) )
       ```

    4. To bring it all together, we'll take this total fine amount and divide it by our new **Number of Infractions** calculated field to determine the average fine amount:

Average Fine = `( SUM([1st Fine Amount]) + SUM( ZN([2nd Fine Amount]) ) ) / SUM([Number of Infractions])`

B. We also need to filter out drivers who ever attended traffic school—but that information is also stored across two fields.

   1. Tableau is very efficient at numerical calculations. We'll phrase this with numbers to help performance as much as we can. To combine these two fields, we'll create a calculation for each one that says "Yes = 1" and "No = 0" (null should also = 0, for drivers with no second infraction). By summing the outcome of these calculations, any driver with an overall value of 0 never went to traffic school (and a value of 1 or 2 represents how many times they went). We can then filter to keep only drivers with a value of 0.
   2. This time, we'll use a `CASE` statement instead of `IF`. They function very similarly but have different syntax. The start of the calculation should look like this:

      ```
      CASE [1st Traffic School]
      WHEN 'Yes' THEN 1
      WHEN 'No' THEN
      ELSE 0
      END
      ```

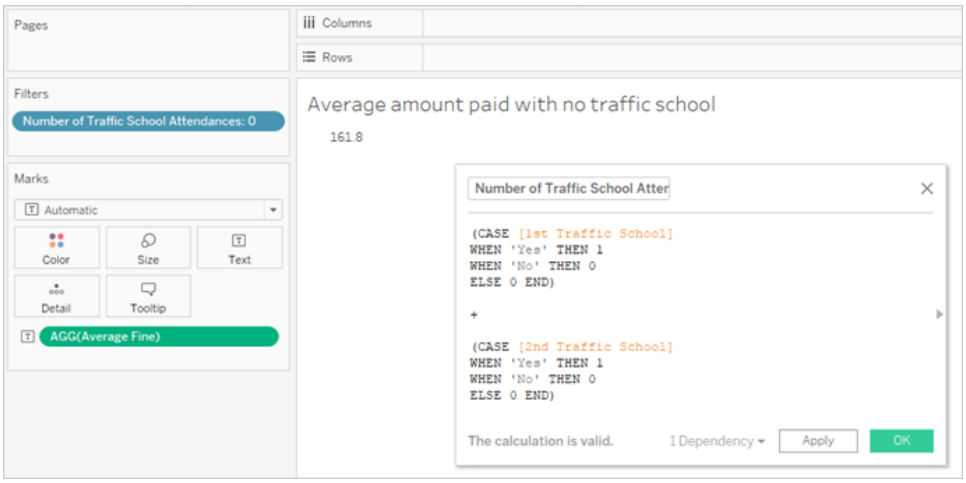
   3. And then we'll do the same thing for 2nd Traffic School. We can add both pieces in the same calculation by wrapping each case statement in parentheses and adding a plus between them. Removing some of the line breaks, it looks like this:

      Number of Traffic School Attendances =

      ```
      (CASE [1st Traffic School] WHEN 'Yes' THEN 1 WHEN 'No'
      THEN 0 ELSE 0 END)
      +
      (CASE [2nd Traffic School] WHEN 'Yes' THEN 1 WHEN 'No'
      THEN 0 ELSE 0 END)
      ```

   4. If we drag **Number of Traffic School Attendances** to the **Dimensions** area of the **Data** pane, the values 0–2 will become discrete.
   5. Now if we filter on **Number of Traffic School Attendances**, we can select just the 0 and know we're getting drivers who have never attended traffic school.

C. To answer the original question, we'll simply bring **Average Fine** to the **Text**shelf on the **Marks** card. Because we built the aggregations into the calculation, the aggregation on the field will be **AGG** and we cannot change it. This is as expected.



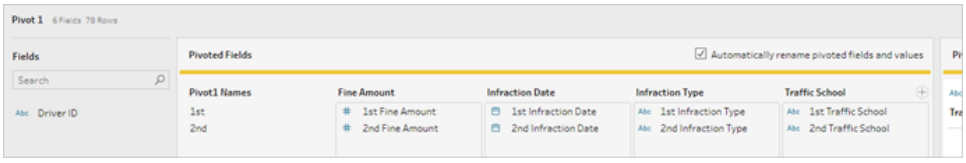
Go Further–Pivoted Data

While the data we've been working with is well structured to address questions specifically around first and second infractions, it isn't the standard structure recommended for use with Tableau Desktop. The more our analysis diverges from basic questions around the infraction dates, the more complicated our calculations become to combine the relevant information into useable form.

Usually, when data is stored with multiple columns for the same type of data (such as two columns for date, two columns for fine amount, etc.) and unique information is stored in the field name (such as whether it's the first or second infraction), this is an indication the data should be pivoted.

Performing a multiple pivot can handle this nicely. Back in Tableau Prep, and working from the end of the **Driver Infraction** Tableau Prep flow created in the step-by-step tutorial above:

1. From the final clean step, add a **Pivot** step that pivots by every duplicated field.



> **Tip**: Use the plus ⊕ icon in the upper right corner of the **Pivoted Fields** area to add more **Pivot Values**. Each set of fields (such as 1st and 2nd Fine Amounts) should be pivoted together.
>
> For more information about pivoting, see **Pivot your data** on page 91.

The results can be tidied by removing null dates as well as renaming and reordering fields.

2. Add a cleaning step after the pivot. In the **Infraction Date** column, right-click on the null bar and choose **Exclude**.
3. Double-click the field name **Pivot1 Names** and rename it **Infraction Number.**
4. Drag fields as appropriate to reorder them as below:

Driver ID	Infraction Number	Infraction Date	Infraction Type	Traffic School	Fine Amount
MO-175001406	1st	05/30/2017	Speeding	Yes	118
SH-199751404	1st	03/04/2017	Speeding	Yes	130
AA-106451404	1st	07/05/2017	Running a red light	No	60
MP-174701406	1st	06/19/2017	Speeding	No	125
PO-188501402	1st	10/30/2017	Speeding	Yes	120
KL-166451406	1st	10/04/2017	Speeding	No	115
RA-199151402	1st	07/20/2017	Speeding	Yes	146
AJ-107951404	1st	10/15/2017	Speeding	Yes	130
BD-117701406	1st	12/25/2017	Speeding	Yes	140
CJ-120101402	1st	11/26/2017	Speeding	Yes	122
SN-207101402	1st	12/27/2017	Speeding	Yes	280
TS-213701404	1st	10/23/2017	Speeding	Yes	100
JR-156701404	1st	12/24/2017	Speeding	No	148
CM-127151402	1st	03/01/2017	Running a red light	No	55
JK-156251406	1st	12/25/2017	Speeding	Yes	140
AE-108851406	1st	05/09/2018	Non-moving violation	No	200

5. From the new, pivoted data, create an output named **Pivoted Driver Infractions** and bring it into Tableau Desktop. (Don't forget to run the flow after adding the **Output** step.)

Now we can look at our five questions again with this pivoted data structure; you can expand each one for basic information about how to proceed if you get stuck.

> **Note**: You can download the completed flow file Pivoted Driver Infractions.tflx to check your work, or download the workbook Pivoted Driver Infractions.twbx to look at the solutions in context. Remember that there may be alternative ways to interpret the analysis or pursue answers.

1. What was the length of time in days between the first and

second infraction for each driver?

A. To answer this in Tableau Desktop, as we did with the first data set, we'll use the `DATEDIFF` function. This function requires a start date and an end date. This information is present in our data, but all in one field. We need to pull it out into two fields.

1. Create two preliminary calculated fields:

   1st Infraction = `IF [Infraction Number] = "1st" THEN [Infraction Date] END`

   2nd Infraction = `IF [Infraction Number] = "2nd" THEN [Infraction Date] END`

2. Because we want to make sure both of these values are available to be compared for each driver, we need to fix them to the level of **Driver ID**.

   Note: Don't believe me? Try to do a `DATEDIFF` calculation with these two fields as they are: **Time Between Infractions** = `DATEDIFF('day', [1st Infraction], [2nd Infraction])`
   You'll get null results everywhere, because Tableau is trying to compare across a data structure that looks like this:

Driver ID	1st Infraction Date	2nd Infraction Date	Time between infractions
AA-106451404	Null	9/28/2017	Null
	7/5/2017	Null	Null
AF-108851406	Null	9/2/2018	Null
	5/9/2018	Null	Null
AJ-107951404	Null	12/31/2017	Null
	10/15/2017	Null	Null

   Here, the row that knows what the first date is doesn't know what the second date is, and vice versa. To get around this, we'll use a FIXED Level of Detail expression to force these first and second dates to be related by **Driver ID**.

   Edit each calculation as follows:

   1st Infraction = `{ FIXED [Driver ID] : MIN ( IF [Infraction Number] = "1st" THEN [Infraction Date] END ) }`

2nd Infraction = `{ FIXED [Driver ID] : MIN ( IF [Infraction Number] = "2nd" THEN [Infraction Date] END ) }`

> **Note**: The original calculation must be aggregated when embedded in an LOD expression. We can use any basic aggregation that will preserve the date value (so aggregations like SUM, AVG, or MIN work, but not CNT or CNTD).

3. Now we can create the `DATEDIFF` calculation as follows:

   Time Between Infractions = `DATEDIFF('day', [1st Infraction], [2nd Infraction])`

4. If we want to look at weeks or months, simply modify the date part (currently `'day'`).

The results will be identical to the outcome with the unpivoted data structure.

2. Compare the fine amounts for the first and second infractions. Are they correlated?

A. To answer this in Tableau Desktop, we'll use very similar logic to the previous question. We'll use **Infraction Number** to identify if a given row is the first or second infraction, then pull out the fine amount accordingly.

   1. If all we want to do is make a scatter plot, we can skip the LOD portion and just use the `IF` calculation:

      1st Fine Amount = `IF [Infraction Number] = "1st" THEN [Fine Amount] END`

      2nd Fine Amount = `IF [Infraction Number] = "2nd" THEN [Fine Amount] END`

   2. However, if we wanted to compare and see the difference in amount between the first and second fines for a single driver, we'd run into the same null issue as with the dates. It can't hurt to wrap these calculations in a `FIXED` LOD, so it might be good just to do so from the start:

1st Fine Amount = `{ FIXED [Driver ID] : MIN ( IF [Infraction Number] = "1st" THEN [Fine Amount] END ) }`

2nd Fine Amount = `{ FIXED [Driver ID] : MIN ( IF [Infraction Number] = "2nd" THEN [Fine Amount] END ) }`

3. Create a scatterplot and bring out a linear trend line as before.

The results will be identical to the outcome with the unpivoted data structure.

3. Which driver paid the most overall? Who paid the least?

A. To answer this question in Tableau Desktop, the pivoted data structure is ideal. All we need to do is bring out **Driver ID** and **Fine Amount** into a bar chart. The default aggregation is already **SUM**, so the total amount paid by the driver will automatically be plotted.

The results will be identical to the outcome with the unpivoted data structure.

4. How many drivers had multiple infraction types?

A. To answer this question in Tableau Desktop, the pivoted data structure is ideal. All we need to do is bring out **Driver ID** and a **Count Distinct** of **Infraction Type** as a bar chart, and we'll have our answer.

The results will be identical to the outcome with the unpivoted data structure.

5. What was the average fine amount for drivers who never attended traffic school?

A. To answer this in Tableau Desktop, we cannot simply divide the total fine amount by two, since some drivers only had one infraction. We also can't calculate the average fine per driver and take the average of those values, because averaging averages can lead to inconsistencies. Instead, we need to calculate the total amount paid by drivers who never attended traffic school, then divide by the total number of infractions associated with those fines.

1. First, we need to determine if each driver had a second infraction. We can leverage the fact **2nd Infraction Date** will be null if there was no second infraction and start building the calculation:

   ```
   IFNULL(STR([2nd Infraction]), 'no')
   ```

   This will return the date of the second infraction if it exists, or "no" if there was no second infraction.

   > **Note**: The `STR` portion of this calculation is necessary because `IFNULL` needs consistency of data type in its arguments. Because we want to return the string "no" for null values, we need to convert the date to a string as well.

2. Next, we need to turn this information into the number of infractions, 1 or 2. If the result of our `IFNULL` calculation is "no", then the driver should be marked as having one fine. Any other result should be marked as having two fines. The calculation is:

   Number of Infractions =

   ```
   IF IFNULL(STR([2nd Infraction]), 'no')= 'no' THEN 1
   ELSE 2
   END
   ```

3. Now we need to consider the average fine amount. We already have a single field for **Fine Amount**. All we need to do is divide that by our new **Number of Infractions** field, wrapping both in **SUM**:

   Average Fine = `(SUM([Fine Amount]) / SUM([Number of Infractions])`

B. We also need to filter out drivers who attended traffic school. It looks like we could use the **Traffic School** field and filter on **Traffic School = no**. However, this would filter on *infractions* not associated with traffic school, not *drivers* who never went to traffic school. If a driver went to traffic school for one infraction but not a second, we don't want the second infraction to be considered here.

   What we want to do is filter out any driver who's attended traffic school. In terms of the data, we want to filter out any driver who has a "Yes" for **Traffic School** on any row.

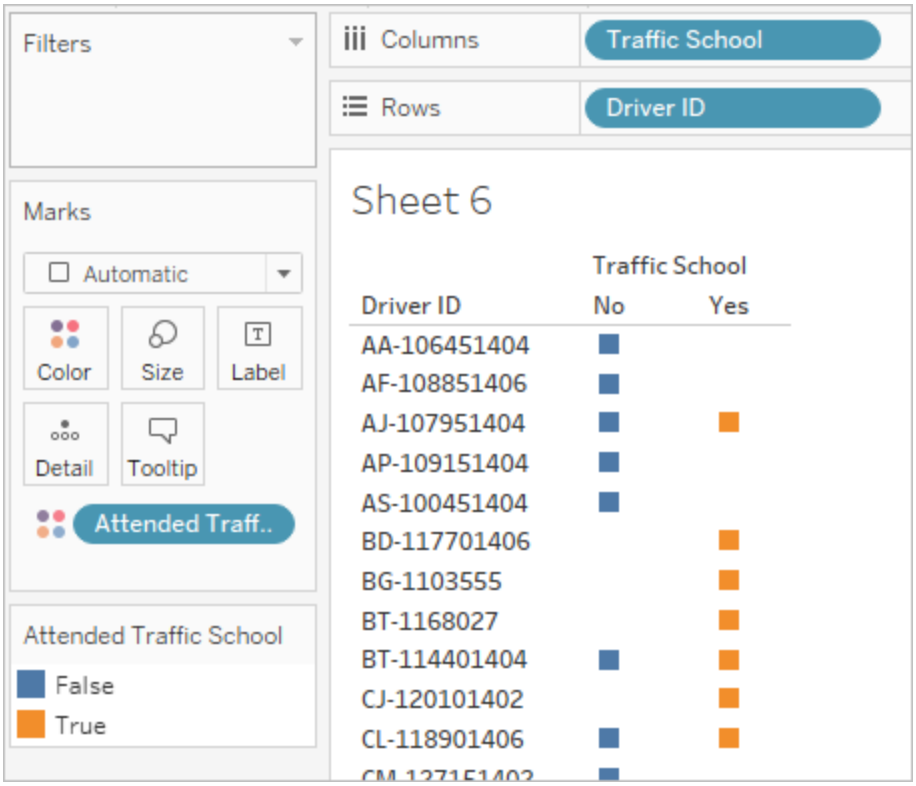
Let's build our calculation in stages, using a simple view to help keep track of what's happening:

1. First, we want to know if a driver has a "Yes" for Traffic School. Drag **Driver ID** to **Rows** and **Traffic School** to **Columns**. We'll get a text table with placeholder "Abc" text indicating the relevant values for each driver.

2. Next, we want to build a calculation that will identify if the value of **Traffic School** is "Yes". The first stage of the calculation is:

   Attended Traffic School = `CONTAINS([Traffic School], 'Yes')`

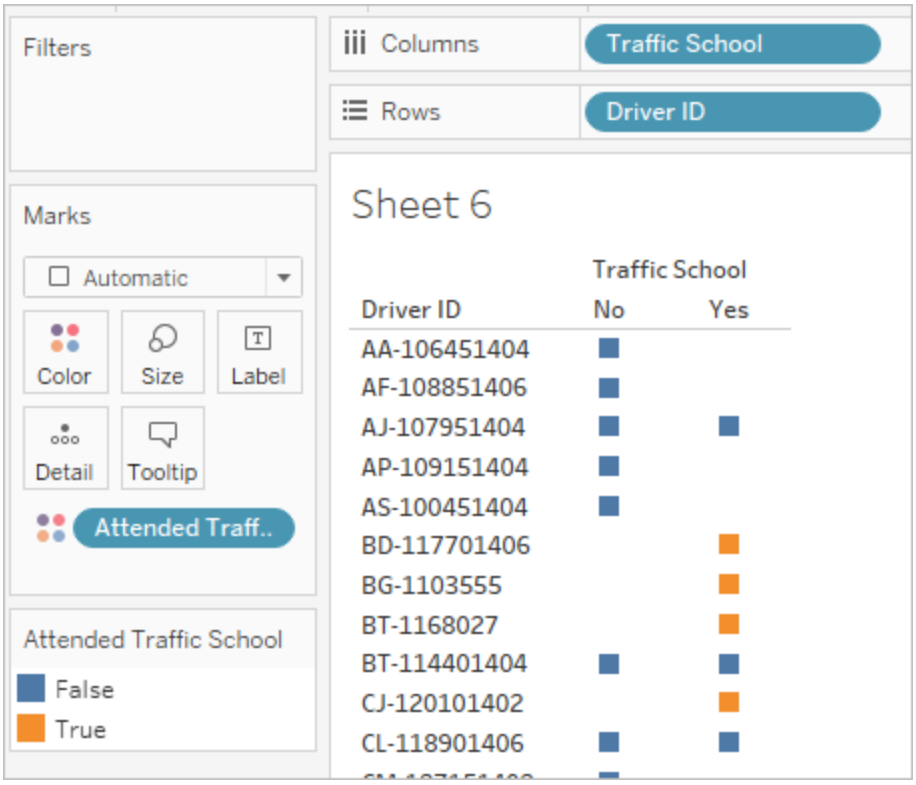
   If we bring **Attended Traffic School** to the **Color** shelf on the **Marks** card, we see it accurately labels "False" for every mark in the "No" column, and "True" for every mark in the "Yes" column.



3. However, what we really want is this information at the level of the *driver*, not the *infraction*. An LOD expression is a natural fit when trying to compute a result at a different level of detail than the basic structure of the data. We'll make this a `FIXED` LOD expression. But, as we know, the aggregate expression portion of an

LOD must be aggregated. Previously, we've used **MIN**. Will that work here? We'll modify the calculation to be:

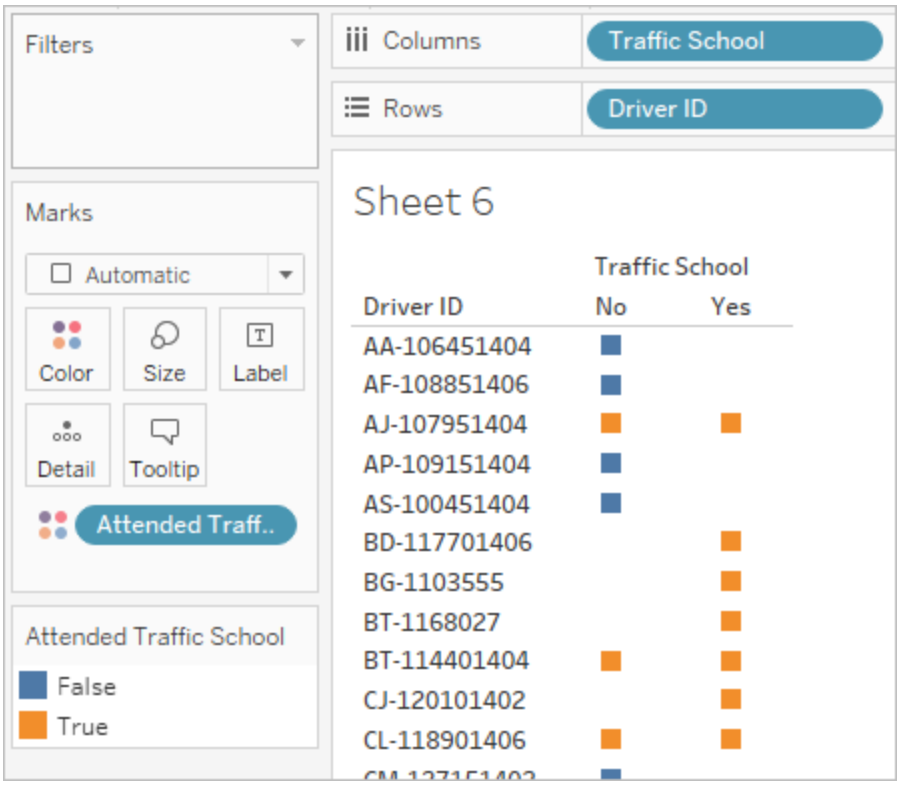
Attended Traffic School = `{ FIXED [Driver ID] : MIN( CONTAINS ([Traffic School], 'Yes'))}`



With that change applied in the view, we see the opposite of what we want. Any driver that has a "No" is marked as "False" across the board. Instead, we want to carry the "Yes" as a "True" for every record for that driver. What is MIN doing here? It's picking the first response alphabetically, that is, "No".

4. What if we changed it to **MAX**? Would that take the last response alphabetically? We'll modify the calculation to be:

   Attended Traffic School = `{ FIXED [Driver ID] : MAX ( CONTAINS ( [Traffic School], 'Yes') ) }`



And here we have it: if a driver has "Yes" anywhere in the data, they are marked as "True" for having attended traffic school, even on the infraction that didn't involve traffic school.

5. If we bring **Attended Traffic School** to the **Filter** shelf and select only "False", we'll be left with only drivers who never attended traffic school.

C. To answer the original question, with our filter in place we'll simply bring **Average Fine** to the **Text**shelf on the **Marks** card. Because we built the aggregations into the calculation, the aggregation on the field will be **AGG** and we cannot change it. This is as expected.

The results will be identical to the outcome with the unpivoted data structure.

The benefits of pivoted data

We could stick with the original data structure from the tutorial if we know we'd only need to answer questions that are easy to answer with that structure. However, the pivoted data format is more flexible. Even though it requires some calculations, once they're in place the resulting data set is well suited to answer broader questions.

Go Further Still–Calculations Only

What if you don't have access to Tableau Prep? Are you out of luck entirely if you're stuck with the original data? Not at all!

Tableau Desktop and LOD expressions can answer all of our analytical questions. If we connect to the original **Traffic Violations.xlsx**, it looks very similar to the pivoted data set—just without the crucial **Infraction Number** field. We'll need to mimic the outcome of the aggregation steps via LOD expressions.

> **Note**: You can download the workbook LOD Driver Infractions.twbx to look at the solutions in context. Remember that there may be alternative ways to interpret the analysis or pursue answers.

1. What was the length of time in days between the first and second infraction for each driver?

A. To answer this in Tableau Desktop, we'll again use the `DATEDIFF` function. This function requires a start date and an end date. This information is present in our data, but all in one field. We need to pull it out into two fields. Because we want to make sure both of these values are available to be compared for each driver, we need to fix them to the level of **Driver ID**.

   1. To find the first infraction date, we use the calculation:

      1st Infraction = `{ FIXED [Driver ID] : MIN ( [Infraction Date] ) }`

   2. We'll do the second infraction date in stages.

      a. To start, we need to look at just the dates that are larger than the first date:

      ```
      IF [Infraction Date] > [1st Infraction] THEN [Infraction Date] END
      ```

      b. But this will give us *every* infraction after the first, and we only want the second. So we want the smallest of these dates. Wrap the whole thing in **MIN**:

```
MIN( IF [Infraction] : [1st Infraction] THEN
[Infraction Date] END )
```

c. We also want to recalculate the second infraction date for each driver. That's where LOD expressions come in. We'll fix this to the level of **Driver ID**:

2nd Infraction = `{ FIXED [Driver ID] : MIN ( IF [Infraction Date] > [1st Infraction] THEN [Infraction Date] END ) }`

3. And we can now create the `DATEDIFF` calculation:

Time Between Infractions = `DATEDIFF('day', [1st Infraction], [2nd Infraction])`

The results will be identical to the outcomes with the other two data structures.

2. Compare the fine amounts for the first and second infractions. Are they correlated?

A. To answer this in Tableau Desktop, we'll use similar logic to the pivoted data version of this question. We'll use the **1st Infraction** and **2nd Infraction** fields we created for question I to identify if a given row is the first or second infraction, then pull out the fine amount accordingly.

1. If all we want to do is make a scatter plot, we can skip the LOD portion and just use an `IF` calculation:

1st Fine Amount = `IF [1st Infraction] = [Infraction Date] THEN [Fine Amount] END`

2nd Fine Amount = `IF [2nd Infraction] = [Infraction Date] THEN [Fine Amount] END`

2. However, if we want to compare and see the difference in amount between the first and second fines for a single driver, we'd run into issues with nulls, as in the first data structure. It can't hurt to wrap these calculations in a FIXED LOD, so it might be good just to do so from the start:

1st Fine Amount = `{ FIXED [Driver ID] : MIN ( IF [1st Infraction] = [Infraction Date] THEN [Fine Amount] END ) }`

2nd Fine Amount = `{FIXED [Driver ID] : MIN( IF [2ndInfraction] = [Infraction Date] THEN [Fine Amount] END ) }`

The results will be identical to the outcomes with the other two data structures.

3. Which driver paid the most overall? Who paid the least?

A. To answer this in Tableau Desktop, we need to first realize something about the LOD-only method. Both methods using Tableau Prep filter out records that are not the first or second infraction for a driver. The LOD method in Tableau Desktop keeps all records. This means that if we were to create a viz of **SUM(Amount Paid)** by **Driver ID**, the Tableau Desktop-only version will show higher amounts for drivers with more than two infractions. To get a **Total Amount Paid** value from the complete data that matches the other methods, instead of using the original **Fine Amount** field, we instead need to sum the first and second fines like we did with the first data structure.

B. Using the fields we created for question 2, we'll add the two fine amounts. `ZN` is necessary to prevent a null result for any drivers who only had one infraction. The calculation is:

   Total Amount Paid = `[1st Fine Amount] + ZN([2nd Fine Amount])`

The results will be identical to the outcomes with the other two data structures.

4. How many drivers had multiple infraction types?

A. To answer this question in Tableau Desktop, we can't simply bring out **Driver ID** and a **Count Distinct** of **Infraction Type**. Because this data set has infractions beyond the second, some drivers may have more than two infraction types. To match the results with the other methods, we need to limit the scope to just the first two infractions.

B. We can to pull out the 1st and 2nd infraction types, wrap them in LOD expressions to make them `FIXED` to the driver, then use an `IF` calculation to count the types:

1. **1st Infraction Type** = `{ FIXED [Driver ID] : MIN ( IF [1st Infraction] = [Infraction Date] THEN [Infraction Type] END ) }`
2. **2nd Infraction Type** = `{ FIXED [Driver ID] : MIN ( IF [2nd Infraction] = [Infraction Date] THEN [Infraction Type] END ) }`
3. **Number of Infraction Types** =

```
IF [1st Infraction Type] = [2nd Infraction Type] THEN 1
ELSEIF [1st Infraction Type] != [2nd Infraction Type]
THEN 2
ELSE 1 END
```

> **Note**: It's also possible to create many of these calculations as a single field by nesting the initial calculations directly in the larger calculation. Here, the combined calculation would look like this:
>
> ```
> IF
>  {FIXED [Driver ID] : MIN(IF [1st Infraction]=
> [Infraction Date] THEN [Infraction Type] END)}
>  =
>  {FIXED [Driver ID] : MIN(IF [2nd Infraction]=
> [Infraction Date] THEN [Infraction Type] END)}
> THEN 1
>
> ELSEIF
>  {FIXED [Driver ID] : MIN(IF [1st Infraction]=
> [Infraction Date] THEN [Infraction Type] END)}
>  !=
>  {FIXED [Driver ID] : MIN(IF [2nd Infraction]=
> [Infraction Date] THEN [Infraction Type] END)}
> THEN 2
> ```

```
ELSE 1
END
```

Which is a bit harder to make sense of, but works if preferred. (Note that line breaks and some spaces do not impact how a calculation is interpreted by Tableau.)

C. We can then plot **Number of Infraction Types** against **Driver ID** and sort the bar chart.

The results will be identical to the outcomes with the other two data structures.

5. What was the average fine amount for drivers who never attended traffic school?

A. To answer this in Tableau Desktop, we cannot simply divide the total fine amount by two, since some drivers only had one infraction. We also can't calculate the average fine per driver and take the average of those values, because averaging averages can lead to inconsistencies. Instead, we need to calculate the total amount paid by drivers who never attended traffic school, then divide by the total number of infractions associated with those fines.

   1. First, we need to determine if each driver had a second infraction. We can leverage the fact the information in all the "2nd" fields will be null if there was no second infraction and start building the calculation:

      `IFNULL([2nd Infraction Type], 'no')`

      This will return an infraction type if it exists, or "no" if there was no second infraction.

   2. Next, we need to turn this information into the number of infractions, 1 or 2. If the result of our `IFNULL` calculation is "no", then the driver should be marked as having one fine. Any other result should be marked as having two fines. The calculation is:

      Number of Infractions =

```
IF IFNULL([2nd Infraction Type], 'no') = 'no' THEN 1
ELSE 2
END
```

3. For the Total Amount Paid, we can use the calculation from question 3. To bring it all together, we'll take this total fine amount and divide it by our new **Number of Infractions** calculated field to determine the average fine amount:

   Average Fine = `SUM([Total Amount Paid]) / SUM([Number of Infractions])`

B. We also need to filter out drivers who attended traffic school. Because this data set contains some drivers with a third or fourth infraction, we can't use the same method as the pivoted data structure. Instead, we'll follow the same method as the unpivoted data, summarized here:

1. First, we need to built two calculations identifying if the first and second infractions involved traffic school or not:

   1st Traffic School = `{ FIXED [Driver ID] : MIN (IF [1st Infraction] = [Infraction Date] THEN [Traffic School] END ) }`

   2nd Traffic School = `{ FIXED [Driver ID] : MIN (IF [2nd Infraction] = [Infraction Date] THEN [Traffic School] END ) }`

2. Then we'll add those values to get the overall number of traffic school attendances:

   Number of Traffic School Attendances =

   ```
   (CASE [1st Traffic School] WHEN 'Yes' THEN 1 WHEN 'No'
   THEN 0 ELSE 0 END)
   +
   (CASE [2nd Traffic School] WHEN 'Yes' THEN 1 WHEN 'No'
   THEN 0 ELSE 0 END)
   ```

3. If we drag **Number of Traffic School Attendances** to the **Dimensions** area of the **Data** pane, the values 0–2 become discrete.

4. Now if we filter on **Number of Traffic School Attendances**, we can select just the 0 and know we're getting drivers who have never attended traffic school.

C. To answer the original question, we'll simply bring **Average Fine** to the **Text**shelf on the **Marks** card. Because we built the aggregations into the calculation, the aggregation on the field will be **AGG** and we cannot change it. This is as expected.

The results will be identical to the outcomes with the other two data structures.

It's important to remember that this solution has a lot of nested calculations and LOD expressions. Depending on the size of the data set and the complexity of the data, performance could be an issue.

Reflection on Methods

So which route should you go? That's entirely up to you and the tools at your disposal.

- If you want to steer clear of LODs, there's a data-shaping solution, though calculations might be necessary for some analysis (**Analysis in Tableau Desktop** on page 178).
- If you can shape the data and are comfortable with calculations—including LODs—the middle-of-the-road option provides the best flexibility (**Go Further—Pivoted Data** on page 185).
- If you're comfortable with LODs, there's minimal impact on performance, and/or you don't have access to Tableau Prep, solving this with LODs alone is a viable option (**Go Further Still—Calculations Only** on page 194).

At the very least, it's valuable to understand how aggregation in Tableau Prep and Level of Detail expressions in Tableau Desktop are interrelated and impact data analysis. As with most things in Tableau, there's more than one way to do anything. Exploring all the various options can help bring concepts together and let you pick the best solution for you.

Calculations used:

Driver Infractions

- **Time Between Infractions** = `DATEDIFF('day', [1st Infraction Date], [2nd Infraction Date])`

- **Total Amount Paid** = `[1st Fine Amount] + ZN([2nd Fine Amount])`
- **Number of Infraction Types** = `IF [1st Infraction Type]=[2nd Infraction Type] THEN 1 ELSEIF [1st Infraction Type]!= [2nd Infraction Type] THEN 2 ELSE 1 END`
- **Number of Infractions** = `IF IFNULL([2nd Infraction Type], 'no') = 'no' THEN 1 ELSE 2 END`
- **Average Fine** = `( SUM([1st Fine Amount]) + SUM( ZN([2nd Fine Amount]) ) ) / SUM([Number of Infractions])`
- **Number of Traffic School Attendances** = `(CASE [1st Traffic School] WHEN 'Yes' THEN 1 WHEN 'No' THEN 0 ELSE 0 END) + (CASE [2nd Traffic School] WHEN 'Yes' THEN 1 WHEN 'No' THEN 0 ELSE 0 END)`

Pivoted Driver Infractions

- **1st Infraction** = `{FIXED [Driver ID] : MIN(IF [Infraction Number] = "1st" THEN [Infraction Date] END)}`
- **2nd Infraction** = `{FIXED [Driver ID] : MIN(IF [Infraction Number] = "2nd" THEN [Infraction Date] END)}`
- **Time Between Infractions** = `DATEDIFF('day', [1st Infraction], [2nd Infraction])`
- **1st Fine Amount** = `{FIXED [Driver ID] : MIN( IF [Infraction Number] = "1st" THEN [Fine Amount] END ) }`
- **Number of Infractions** = `IF IFNULL(STR([2nd Infraction]), 'no')= 'no' THEN 1 ELSE 2 END`
- **Average Fine** = `SUM([Fine Amount])/SUM([Number of Infractions])`
- **Attended Traffic School** = `{ FIXED [Driver ID] : MAX( CONTAINS ([Traffic School], 'Yes'))}`

LOD Driver Infractions

- **1st Infraction** = `{FIXED [Driver ID] : MIN([Infraction Date])}`
- **2nd Infraction** = `{ FIXED [Driver ID] : MIN( IF [Infraction Date]`

> [1st Infraction] THEN [Infraction Date] END) }

- **Time Between Infractions** = `DATEDIFF('day', [1st Infraction], [2nd Infraction])`
- **1st Fine Amount** = `{FIXED [Driver ID] : MIN( IF [1st Infraction] = [Infraction Date] THEN [Fine Amount] END ) }`
- **2nd Fine Amount** = `{FIXED [Driver ID] : MIN( IF [2nd Infraction] = [Infraction Date] THEN [Fine Amount] END ) }`
- **Total Amount Paid** = `[1st Fine Amount] + ZN([2nd Fine Amount])`
- **1st Infraction Type** = `{FIXED [Driver ID] : MIN( IF [1st Infraction] = [Infraction Date] THEN [Infraction Type] END ) }`
- **2nd Infraction Type** = `{FIXED [Driver ID] : MIN( IF [2nd Infraction] = [Infraction Date] THEN [Infraction Type] END ) }`
- **Number of Infraction Types** = `IF [1st Infraction Type]=[2nd Infraction Type] THEN 1 ELSEIF [1st Infraction Type]!= [2nd Infraction Type] THEN 2 ELSE 1 END`
- **Number of Infractions** = `IF IFNULL([2nd Infraction Type], 'no') = 'no' THEN 1 ELSE 2 END`
- **Average Fine** = `SUM ([Total Amount Paid]) / SUM([Number of Infractions])`
- **1st Traffic School** = `{FIXED [Driver ID] : MIN (IF [1st Infraction] = [Infraction Date] THEN [Traffic School] END ) }`
- **2nd Traffic School** = `{FIXED [Driver ID] : MIN (IF [2nd Infraction] = [Infraction Date] THEN [Traffic School] END ) }`
- **Number of Traffic School Attendances** = `(CASE [1st Traffic School] WHEN 'Yes' THEN 1 WHEN 'No' THEN 0 ELSE 0 END) + (CASE [2nd Traffic School] WHEN 'Yes' THEN 1 WHEN 'No' THEN 0 ELSE 0 END)`

Note: Special Thanks to Ann Jackson's Workout Wednesday topic Do Customers Spend More on Their First or Second Purchase? and Andy Kriebel's Tableau Prep Tip

Returning the First and Second Purchase Dates that provided the initial inspiration for this tutorial. Clicking these links will take you away from the Tableau website. Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the owners if you have questions regarding their content.

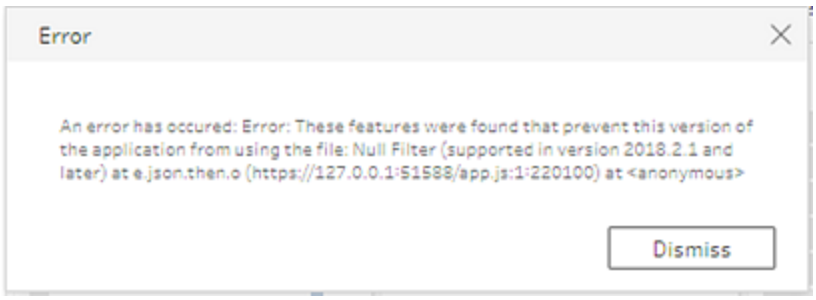
Troubleshoot Tableau Prep

This article lists problems you might encounter when using Tableau Prep and suggestions for how to resolve them.

In this article

Error: "These features were found that prevent this version of the application from using this file"

If you open a flow that was created in version 2018.2.1 or later in an earlier version of Tableau Prep, you may see the following error:



Flows that include features that are not supported in earlier releases will result in this incompatibility error. To resolve the error, open the flow in the later version, and save a copy of the flow without the indicated feature. In the above example, remove the null filter from the field where it is applied.

Then open the copy that has the feature removed in the earlier version of Tableau Prep.

Error: "Failed to parse response from Tableau Server" when publishing Tableau Prep output

To successfully publish output from Tableau Prep to Tableau Server, the REST API must be enabled on Tableau Server. If the REST API is not enabled, you will see the following error:

```
Failed to parse response from Tableau server due
to:javax.xml.bind.UnmarshalException - with linked
exception: [org.xml.sax.SAXParseException; lineNumber: 1;
columnNumber: 10; DOCTYPE is disallowed when the feature
"http://apache.org/xml/features/disallow-doctype-decl" set
to true.]
```

For information about enabling the REST API on Tableau Server, see REST API Requirements in the REST API Help.

For information about publishing output from Tableau Prep, see **Create and publish data extracts and data sources** on page 137.

Error: "You are using Server version: null..." when signing in to an SSL-enabled Tableau Server using Tableau Prep

When you sign in to an SSL-enabled Tableau Server from Tableau Prep, you must have a root certificate installed on the computer where Tableau Prep is installed. If the certificate is not installed, you might see the following error:

```
You are using Server version: null but the minimum
compatible version is: 10.0. Please upgrade to a compatible
version.
```

If you see this error, work with your IT department or system administrator to install the required root certificate on the computer where Tableau Prep is installed. For more information, see System requirements in the Tableau Desktop and Tableau Prep Deployment Guide.