Tableau Prep Help

Last Updated 25/07/2024

© 2024 Salesforce, Inc.

Contents

New Features in Tableau Prep ... 1

Related resources ... 1

Get Started with Tableau Prep Builder ... 3

Sample files ... 3

Here's the story... ... 4

1. Connect to data ... 4

Check your work: Watch "Connect to data" in action. ... 9

2. Explore your data ... 9

3. Clean your data ... 11

Clean Orders_Central ... 11

Review your changes ... 17

Check your work: Watch "Clean Orders_Central" in action. ... 18

Clean Orders_East ... 18

Clean Orders_West ... 21

4. Combine your data ... 25

Union your data ... 25

Check your work: Watch "Union your data" in action. ... 30

Clean the product returns data ... 30

Join your data ... 35

Clean your join results ... 40

5. Run your flow and generate output ... 43

Wrap up and resources ... 46

About Tableau Prep ... 47

Using Tableau Prep ... 47

Watch a video: See Tableau Prep Builder in action ... 49

50
A tour of the Tableau Prep workspace 50
Connections pane (1) 51
Flow pane (2) 52
Profile pane (3) 53
Data grid (4) 55
How Tableau Prep stores your data 55
Tableau Prep on the Web 56
Installation and Deployment 56
Sample data and processing limits 57
Available features on the web 57
Autosave and working with drafts 59
Publishing flows on the web 60
Embed credentials 60
Publish a flow 61
Who can do this 62
Tableau Prep Visual Dictionary 63
Start or Open a Data Flow 67
Start a new flow 67
Open an existing flow 71
Open a flow in Tableau Prep Builder 71
Open a flow in Tableau Prep on the web 72
Connect to Data 75
Connect via built-in connectors for popular data types 75
Considerations when using built-in connectors 76
Tableau Prep Builder 76
Tableau Prep on the web 76

Versions 2023.2 and later 78
Version 2021.4 78
Versions 2021.1 - 2021.3 79
Configure SSL to connect to Google BigQuery (MacOS only) 79
Set up and manage your Google BigQuery credentials 80
Sign In using Service Account (JSON) file 81
Sign In using OAuth 81
Supported cleaning operations 84
Before you connect 84
Connect to spatial files 84
Before you connect 86
Connect using Other Databases (ODBC) 87
Connect using custom connectors 90
Use partner-built connectors 90
Connect to published data sources 91
About credentials and permissions: 92
Using published data sources in your flow 92
Connect to Virtual Connections 97
Considerations when connecting to virtual connections: 97
Connect to Tableau data extracts 99
Connect to data via Tableau Catalog 100
Other connection options 100
Use Custom SQL to connect to data 100
Use Initial SQL to query your connections 102
Run Initial SQL 103
Include parameters in your Initial SQL statement 104
Configure your Data Set 104

Include row numbers from your data set 105
Add the Source Row Number field to your flow 106
Source Row Number details 107
Set header and data start row 107
Configure the header and start row 109
Multiple schema in a single file 110
Union multiple tables 112
Connect to a custom SQL query 113
Apply cleaning operations in an input step 114
Select fields to include in the flow 114
Apply filters to fields in the Input step 115
Apply a calculation filter 115
Apply a relative date filter 117
Change field names 118
Change data types 119
Configure field properties 121
Configure text settings in text files 121
Set your data sample size 121
Preparing your data for sampling 122
Changing the data sample settings 122
Examples 124
Add More Data in the Input Step 126
Refresh input step data or change your connection 127
Refresh your data source 127
Replace your data source 127
Edit the connection 128
Replace the input connection 128

Union files and database tables in the Input step 130
Union files 131
Core filter criteria 131
Additional filters 131
Create an input union 134
Union database tables 139
Merge fields after a union 141
Join data in the Input step 141
Build and Organise your Flow 144
Add or insert steps 144
Add steps 145
Insert steps 147
Group steps 151
Requirements to group steps 151
Create a group 152
Change the flow colour scheme 154
Remove steps from the flow 155
Add descriptions to flow steps and cleaning actions 155
Add a description to flow steps 155
Add a description to a change entry 157
Reorganise the layout of your flow 160
Use the flow navigator tool 161
Examine Your Data 163
Review the data types assigned to your data 163
See size details about your data 164
See the distribution of values or unique values 166
Search for fields and values 168

Copy field values in the data grid 170
Sort values and fields 171
Reorder fields 172
Highlight fields and values in a flow 173
Trace fields in a flow 173
See related values 174
Highlight identical values 174
Filter Your Data 175
Keep or remove fields 176
Hide fields 177
Hide and unhide fields 178
Filters available for each data type 180
Where are my filter options? 180
Calculation filter 181
Selected Values filter 182
Range of Values filter 182
Range of Dates filter 183
Relative Date filter 183
Wildcard Match filter 184
Null Values filter 184
Remove Duplicate Rows 185
Identify and remove duplicates 185
Duplicates in sample data 189
Sort order for the calculated field 189
Write your own filter calculation to find and remove duplicates 190
Use Data Roles to Validate your Data 190
Assign standard data roles to your data 191

Create customised data roles 194
Requirements 194
Create a custom data role 195
Apply a custom data role 199
View and manage custom data roles 201
Group similar values by data role 202
Create and Use Parameters in Flows 205
Where can I apply parameters? 206
Create user parameters 208
Change the user parameter default value 210
Edit user parameters 210
Reset user parameter default values 211
Apply parameters to your flow 213
Apply parameters to input steps 213
Filename or file path 213
Database table 214
Custom SQL 215
Apply user parameters to output steps 215
Filename or file path 216
Published data source name 216
Database table and Before and After Custom SQL 217
Apply system parameters to output steps 218
Filename 218
Published data source name 219
Apply user parameters to filter calculations 219
Apply user parameters to calculated fields 220
Delete user parameters 221

Run flows with parameters 223
Run flows manually 224
Run flows on a schedule 225
Clean and Shape Data 227
About cleaning operations 227
Available cleaning operations 227
Order of operations 229
Apply cleaning operations 231
Select your view 232
Pause data updates to boost performance 235
Apply cleaning operations 236
Rename fields in bulk 239
View your changes 241
Merge fields 243
Apply cleaning operations using recommendations 245
Apply recommendations 246
Edit field values 248
Edit a single value 248
Edit multiple values 249
Edit multiple values using quick cleaning operations 249
Group and edit multiple values inline 251
Replace one or more values with Null 251
Manually map multiple values to a standard value 252
Map multiple values to a single selected field 253
Create a group by selecting multiple values 253
Add and identify values that aren't in the data set 255
Automatically map values to a standard value using fuzzy match 257

Group similar values using fuzzy match 258
Adjust your results when grouping field values 260
Copy steps, actions and fields 262
Copy and paste steps 263
Copy and paste cleaning operations 264
Copy fields 267
Create reusable flow steps 269
Create reusable steps 269
Insert reusable steps in a flow 270
Fill Gaps in Sequential Data 272
Generate new rows 273
Create Level of Detail, Rank and Tile Calculations 275
Calculate level of detail 276
Create Level of Detail (LOD) calculations 277
Calculation editor 277
Visual calculation editor 278
Calculate rank or row number 280
Supported analytic functions 281
Create Rank or Row Number calculations 285
Calculation editor 286
Visual Calculation editor 289
Calculate tiles 292
Create Tile calculations 294
Visual Calculation editor 294
Calculation editor 297
Calculate Values Across Multiple Rows 299
Calculate Difference From 300

Visual calculation editor 300
Calculation editor 303
Calculate Percent Difference From 305
Visual Calculation editor 305
Calculation editor 308
Calculate Moving Average or Sum 310
Visual Calculation editor 310
Calculation editor 313
Get Previous Value 315
Fill Down Calculation 317
Prep Visual calculation editor 318
Calculation editor 320
Examples 320
Running Calculation 323
Prep Visual calculation editor 323
Calculation editor 325
Use Einstein Copilot 327
Create Calculated Fields 327
Guidelines for Writing Instructions 329
Help Einstein Improve 329
Examples 330
Example Calculation 1: 330
Example Calculation 2: 331
Pivot Your Data 332
Pivot columns to rows 333
Watch "pivot on multiple field" in action. 336
Use wildcard search to pivot 336

Pivot rows to columns 338
Use R and Python scripts in your flow 341
Use R (Rserve) scripts in your flow 342
Prerequisites 343
Resources 343
Configure Rserve Server for Tableau Server 343
Additional Rserve configuration (optional) 344
Create your R script 345
Connect to your Rserve server 346
Add a script to your flow 347
Use Python scripts in your flow 350
Prerequisites 351
Configure the Tableau Python (TabPy) server for Tableau Server 351
Create your python script 352
Connect to your Tableau Python (TabPy) server 354
Add a script to your flow 355
Aggregate, Join or Union Data 359
Aggregate and group values 359
Join your data 360
Inspect the results of the join 363
Common join issues 365
Fix mismatched fields and more 366
Union your data 367
Inspect the results of the union 369
Fix fields that don’t match 370
Additional merge field options 373
Add Einstein Discovery Predictions to your flow 374

What is Einstein Discovery? 374
Prerequisites 375
Salesforce Requirements 375
Tableau Prep Requirements 376
Add prediction data to your flow 377
Reviewing your results 381
Save and Share Your Work 383
Save a flow 383
Automatically save your flows on the web 384
Automatic file recovery 385
View flow output in Tableau Desktop 386
Create data extract files and published data sources 387
Tableau Prep Builder 387
Tableau Prep Builder and on the web 388
Include parameters in your flow output 388
Create an extract to a file 388
Create an extract to a Microsoft Excel Worksheet 390
Create a published data source 392
Save flow output data to external databases 394
Output options 395
Additional options 395
Supported databases and database requirements 396
Save flow data to a database 397
Save flow output data to Datasets in CRM Analytics 401
Prerequisites 402
Salesforce Requirements 402
Tableau Prep Requirements 403

Save flow data to CRM Analytics 403
Save flow output data to Data Cloud 405
Permission prerequisites 405
Save flow data to Data Cloud 406
Considerations 409
Data Cloud setup prerequisites 409
Set up an Ingestion API Connector 410
Create a data stream 410
Add your data stream to a data space 411
Map the Data Lake Object to Salesforce objects 411
Create a Connected App for Data Cloud Ingestion API 412
Schema requirements 412
YMAL file example 413
Save flow output data to Amazon S3 414
Permissions 415
Save flow data to Amazon S3 415
Refresh Flow Data Using Incremental Refresh 416
Flow refresh options 417
Configure incremental refresh 418
Incremental refresh with Append 421
Configure write options 421
Run your flow 423
Refresh flow output files from the command line 425
Before running the flow 426
Credentials .json file requirements 427
Version 2020.3.1 and later 428
Run flows that include parameter values 429

Examples 430
Connecting to a server connection 430
Connecting to a server connection and output to a database connection 430
Flow includes Rserve and TabPy script connections and outputs to a database connection 431
Connecting to and outputting to different database connections 432
Version 2020.2.3 and earlier 433
Examples 434
Connecting to a published data source 434
Connecting to two databases 434
Flow includes script steps for Rserve and TabPy and connects to a database 435
Tips for creating your credentials file 436
Run the flow 437
Run the flow with incremental refresh enabled 438
Command options 439
Syntax examples 442
The flow connects to and publishes to local files 442
The flow connects to and publishes to local files and uses the short form for incremental refresh 442
The flow connects to databases and publishes to a server 443
The flow publishes to a server and the credentials file is stored on a network share 443
Version Compatibility with Tableau Prep 444
Version number format 444
Finding your version 445
Compatibility between different versions of Tableau Prep Builder 447
Fix compatibility issues with Tableau Prep Builder 448
Compatibility between different versions of Tableau Prep Builder and Tableau Server 448

Detect incompatible features 448
Tableau Prep Builder (version 2020.1.1 and later) 449
Tableau Prep Builder (version 2019.3.1 and later) 450
Tableau Prep Builder (all versions) 451
Fixing compatibility issues 453
Identify incompatible features 453
Remove incompatible features from the flow 455
Incompatible data sources 455
Incompatible features 455
Keep Flow Data Fresh 457
Run your Flow 460
Flow run options 460
Run flows manually 461
Publish a Flow to Tableau Server or Tableau Cloud 463
Before you publish 463
Publish a flow from Tableau Prep Builder 467
Tableau Server 467
Files 469
Databases 472
Tableau Cloud 476
Files 477
Databases 478
Who can do this 483
Day in the Life Scenarios 484
Hospital Bed Use with Tableau Prep 484
The Data 485
Preliminary Analysis 485

Desired Data Structure 486
Restructuring the Data 488
Bed Hour Matrix 488
Patient Bed Use 491
Analysis in Tableau Desktop 494
Recap and Resources 497
Finding the Second Date with Tableau Prep 499
The Data 500
Desired Data Structure 500
Restructuring the Data 501
Initial Aggregation for 1st Infraction Date 501
Second Aggregation for 2nd Infraction Date 505
Create full data sets for the 1st and 2nd infractions 507
Create the complete data set 509
Recap 510
Continue to Analysis with the Second Date in Tableau Desktop on page 1. 511
Analysis with the Second Date in Tableau Desktop 511
Analysis in Tableau Desktop 512
Go Further – Pivoted Data 519
The benefits of pivoted data 528
Go Further Still – Calculations Only 529
Reflection on Methods 534
Driver Infractions 535
Pivoted Driver Infractions 535
LOD Driver Infractions 536
Troubleshoot Tableau Prep Builder 539
Running LogShark 539

Common errors when using the command line to run flows 539

Error: "These features were found that prevent this version of the application from using this file" 544

Error: "You are using Server version: null..." when signing in to an SSL-enabled Tableau Server using Tableau Prep 545

Maintain Licences for Tableau Desktop and Tableau Prep 545

View data about your licence 546

Automatically refresh product keys using zero downtime licensing 547

Track Tableau Desktop licence usage and expiration data 548

Additional resources 549

Tableau Prep Functions Reference 551

Number functions 551

String functions 556

Aggregate functions 560

Type conversion functions 562

Date functions 564

Logical functions 566

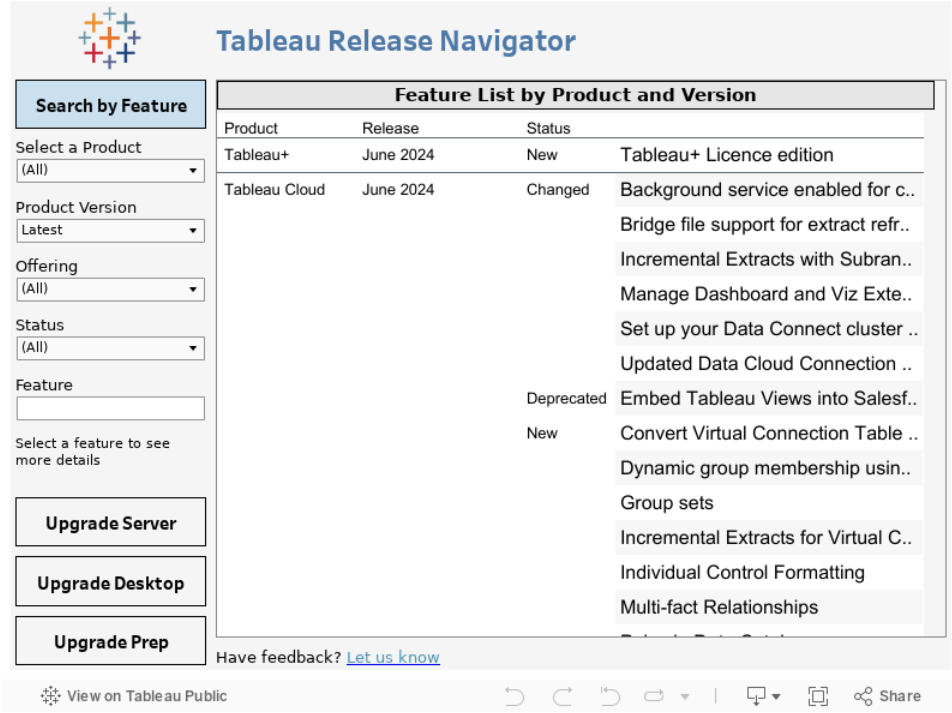
Analytic functions 567

Keywords 570

New Features in Tableau Prep

Use the viz below to explore new features in Tableau Prep. Click on a feature to bring up the tooltip with a link to detailed documentation for that feature. Explore the filters to refine your search. Download the data to create a customised list.

- Use the **Search by Feature** dashboard to see a list of new features for a product or version, or explore when a feature was released. The dashboard currently defaults to Tableau Prep as the product (which includes Prep Builder and Prep Conductor features) for the version Tableau Prep Builder.
- Use the **Upgrade Prep** dashboard to see a list of features specific to your upgrade. If you publish flows to Tableau Server to run them on a schedule, some new features require a minimum Tableau Server version to run. The view lists the minimum Tableau Server version that supports scheduling the flows created in a specific version of Tableau Prep Builder to help you quickly spot features with compatibility requirements.



Related resources

New Features

Browse summaries of new features for currently supported versions.

All Known Issues | Downloads

Get Started with Tableau Prep Builder

> **Note**: Starting in version 2020.4.1, as a Creator, you can also create and edit flows on the web. This tutorial was designed using Tableau Prep Builder, but can also be done on the web with some noted exceptions.

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. This tutorial uses the most current version of Tableau Prep Builder. If you are using a previous version, your results may differ.

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 Builder before continuing with this tutorial, see Install Tableau Desktop or Tableau Prep Builder from the User Interface in the Tableau Desktop and Tableau Prep Builder Deployment guide. Otherwise you can download the free trial.

Sample files

To complete the tasks in this tutorial, you need to install Tableau Prep Builder, or if web authoring is enabled on your server version 2020.4 or later, you can also try the steps on the web.

After installing Tableau Prep Builder on your machine, you can also find the sample files in the following location:

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

Alternatively, download the sample files from these links and create a Samples directory and a South sub-directory. You'll need to do this if completing this tutorial on the web.

Download to Samples directory	Download to South Sub-directory
• Orders_Central	• Orders_South_2015

Download to Samples directory

- Orders_East
- Orders_West
- returns_reasons_new

Download to South Sub-directory

- Orders_South_2016
- Orders_South_2017
- Orders_South_2018

Here's the story...

You work at the headquarters for a large retail chain. Your boss wants to analyse 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!

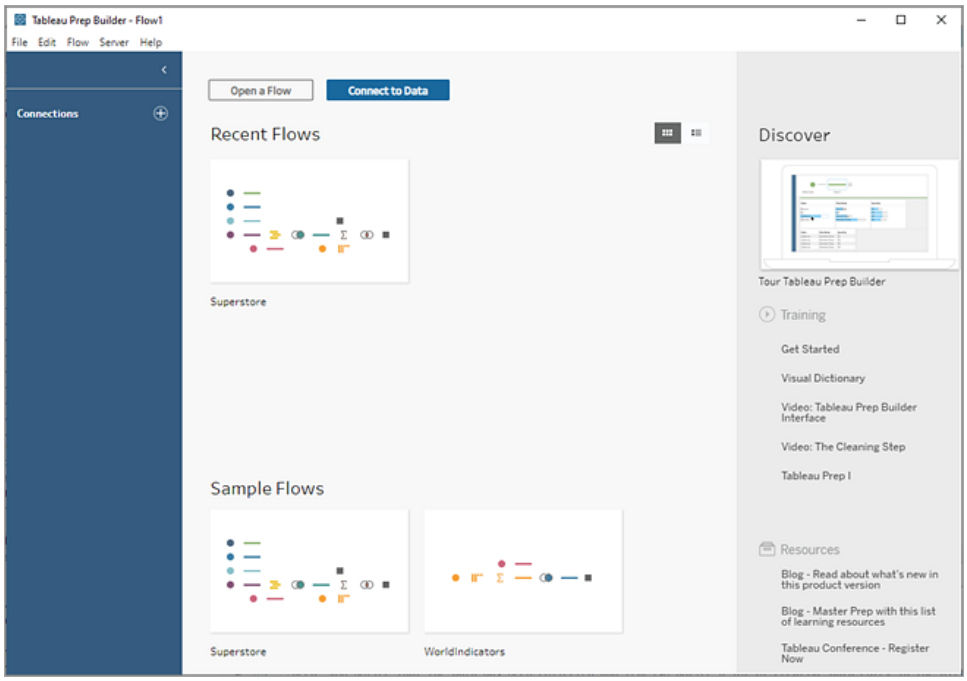
Before you can start analysing the data in Tableau, you'll have to do some serious data cleaning first, and it's going to be a long night.

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

You download the product, or 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 Builder 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 75.

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.

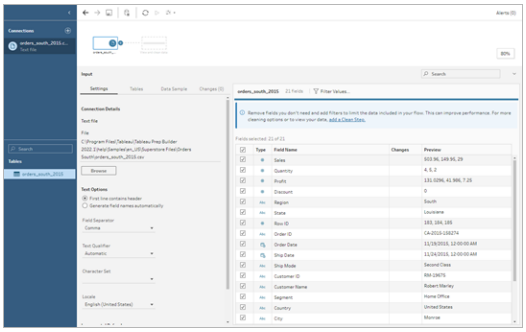
   In web authoring, from the **Home** page, click **Create** > **Flow** or from the **Explore** page, click **New** > **Flow**. Then click **Connect to Data**.

2. The files are .csv files, so select **Text file** in the list of connections.

3. Navigate to the directory for your files. In the Orders South subdirectory, 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 46.)

After you connect to your first file, the Tableau Prep Builder 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.



Much like Tableau Desktop, this **Flow** pane is your workspace, 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 examples of your values from 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, and how you combine them depends on where you're working.

   In Tableau Prep Builder:

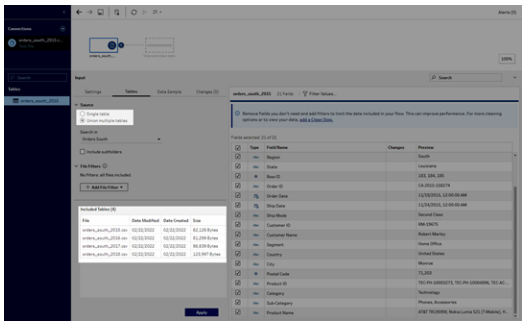
   a. You could add each file individually, but you want to combine all the files together into one Input step so you click the **Tables** tab in the **Input** pane.

   b. You see an option for **Union multiple tables**. 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 and combine 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 and database tables in the Input step** on page 130.



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

d. 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.

> **Note**: On the web, files can only be uploaded individually.

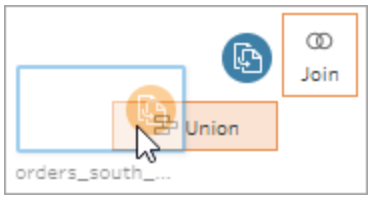
In Tableau Server or Tableau Cloud:

The wildcard option isn't currently available for Tableau Server or Tableau Cloud. Still, you want to include all of the files from the South and handle the data alike, so combining them makes sense.

a. Repeat steps 2 and 3 to add the rest of the files from the Orders South subdirectory.

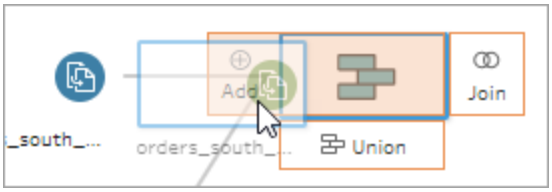
b. Combine them with a union step. (For more details, see **Union files and database tables in the Input step** on page 130.)

   i. Drag **Orders_South_2016** on top of **Orders_South_2015** and drop it on the **Union** option.



   ii. Drag **Orders_South_2017** on top of the new Union step and drop it on **Add**.

Repeat this step with the final file.

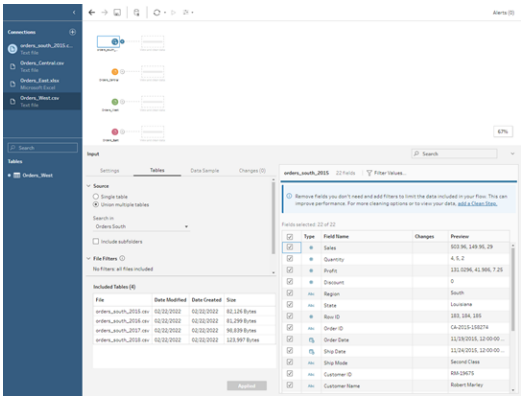


5. Add the remaining files.

In Tableau Prep Builder:

- Open File Explorer or Finder and navigate to the directory for the files. Ctrl-click or Cmd-click (MacOS) 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 46.)
    - Orders_Central.csv
    - Orders_East.xlsx
    - Orders_West.csv

> **Note**: These are different file types. If you don't see all of these files, make sure your file explorer or finder is set to view all file types.

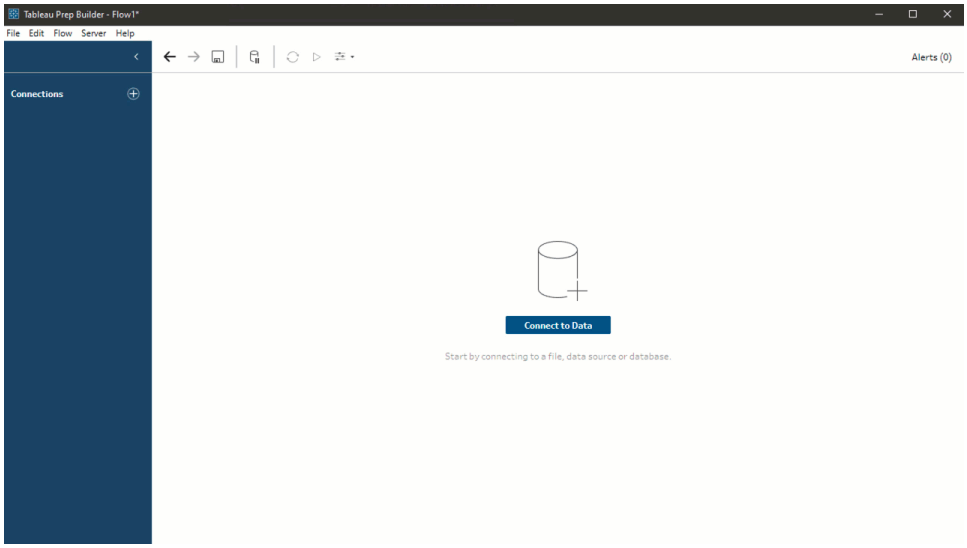


In Tableau Server or Tableau Cloud:

- Follow steps 2 and 3 to add Orders_Central.csv and Orders_West.csv.
- On the **Connections** pane, click the **Add connection** button. Click **Microsoft Excel** and select **Orders_East.xlsx**.

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 all of 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 a preview of your values.

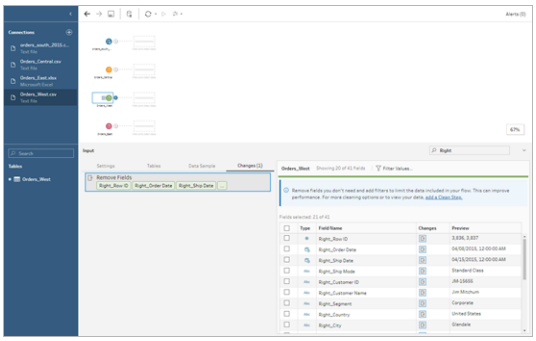
This is a good place to decide how much data you want to include in your flow and remove or filter fields that you don't want. You can also change any data types that were assigned incorrectly.

Tip: If you are working with large data sets, Tableau Prep automatically brings in a sample of the data to maximise 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 **Set your data sample size** on page 121.

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.

- Select the **Orders_West** Input step.
  - 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 right here in the **Input** step:

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



    Tip: When you perform cleaning operations in a step, like removing fields, Tableau Prep tracks your changes in the **Changes** pane and adds an annotation (in the form of a little icon) in the Flow pane to help you keep track of the actions you take on your data. For Input steps, an annotation is also added to each field.

- 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 delivery 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 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.

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. For more information about cleaning options, see **Clean and Shape Data** on page 227

Steps come in many flavours, depending on what you are trying to do. For example, add a cleaning step any time you want to apply cleaning operations to your fields like filter, merge, split, rename and so on. Add an aggregation step 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 and Organise your Flow** on page 144.

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.

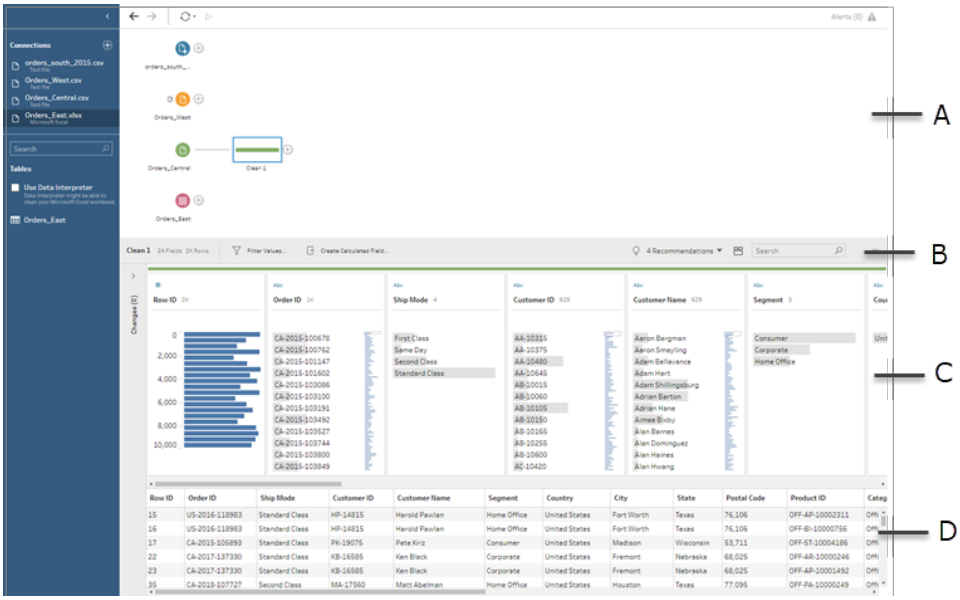
Clean Orders_Central

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**, do one of the following:
    - Click the plus icon and add a cleaning step. Depending on your version, this menu option is **Add Step**, **Add Clean Step** or **Clean Step**.
    - Click on the suggested clean step (Tableau Prep Builder version 2020.3.3 and later

and on the web)

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.

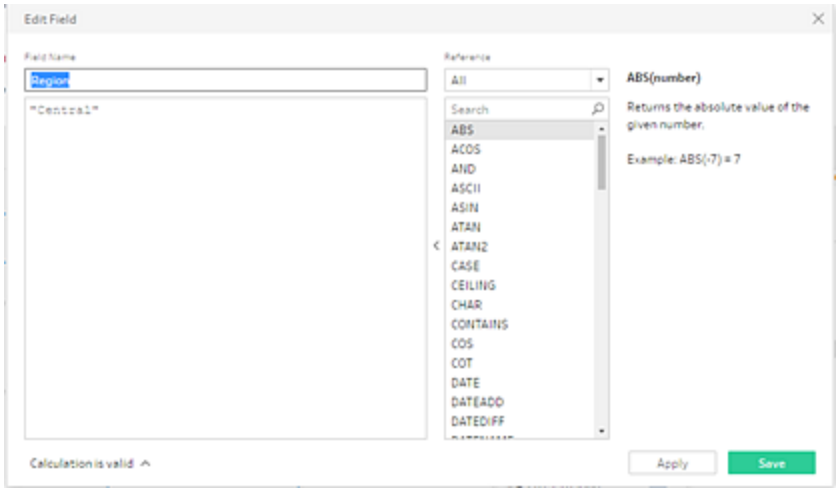
The **Profile** pane shows you the structure of your data, summarising the field values into bins so that you can quickly see related values and spot outliers and null values. The **Data** grid shows you the row level detail for your fields.

Tip: Each field in the **Profile** pane is shown on a profile card. Use the **More options** ··· menu (drop-down arrow in prior versions) 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, assign a data role to the field 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. In the Calculation editor, 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 on the left. An icon (annotation) 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.

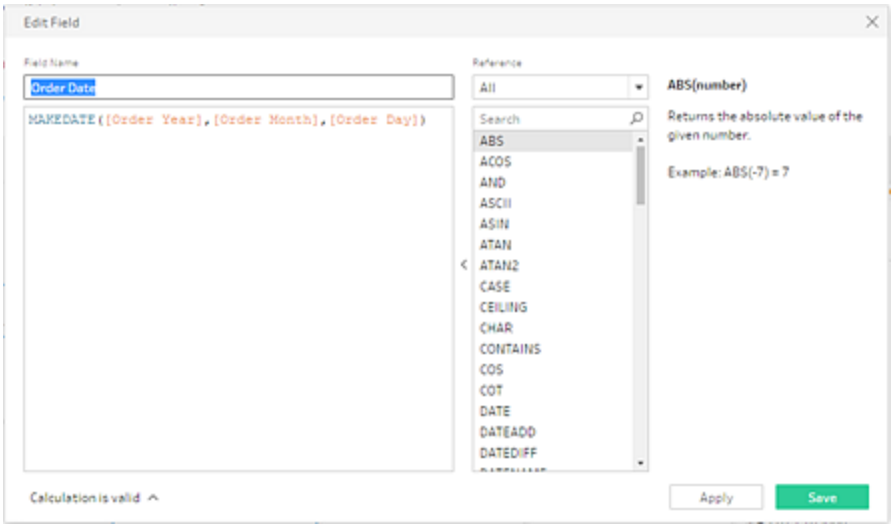
Next you want to address the separate order date and delivery date fields. You want to combine them into two single fields, one for **Order Date** and one for **Delivery Date** so they align with the same fields in the other data sets. Making sure your tables have the same fields will enable you to combine the tables using a union later.

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. In the Calculation editor, name the calculated field **Order Date**. Then enter the following calculation 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**.

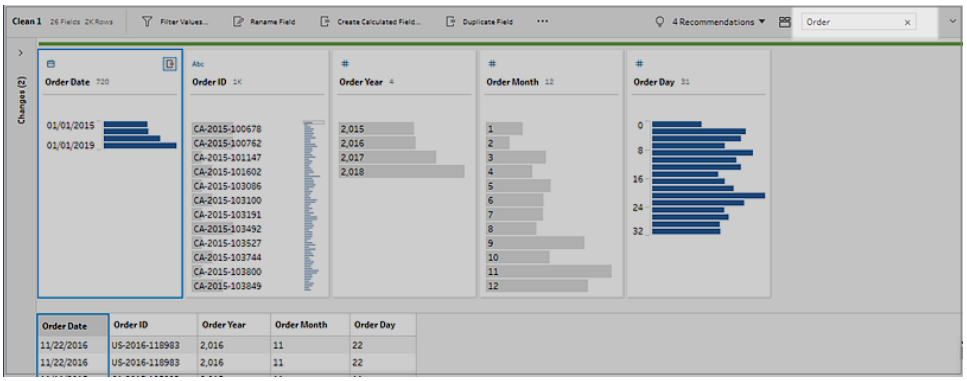
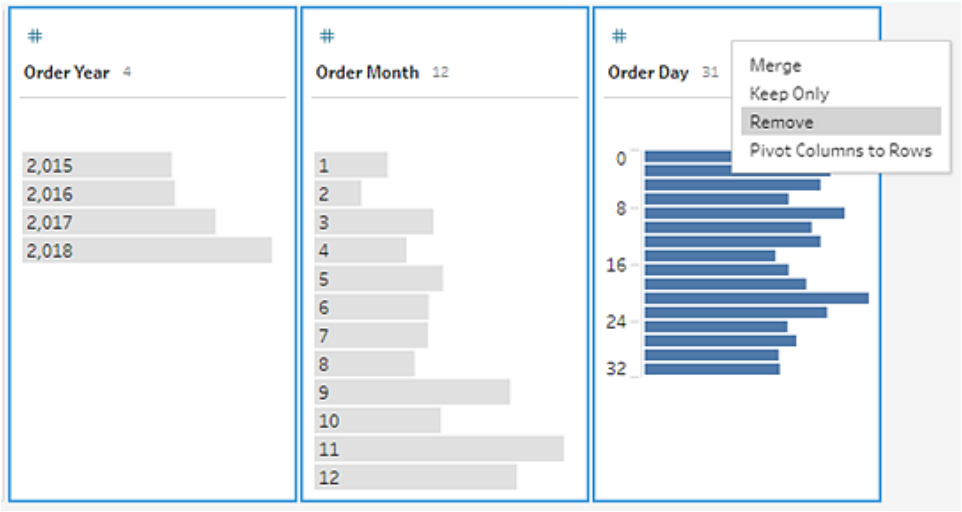


Tableau Prep quickly scrolls all the fields with **Order** in the name into view. Cool!

7. Ctrl-click or Cmd-click (MacOS) to select the fields for **Order Year**, **Order Month**, and **Order Day**. Then right-click on the selected fields and select **Remove** (**Remove Field** in prior versions) from the menu to remove them.



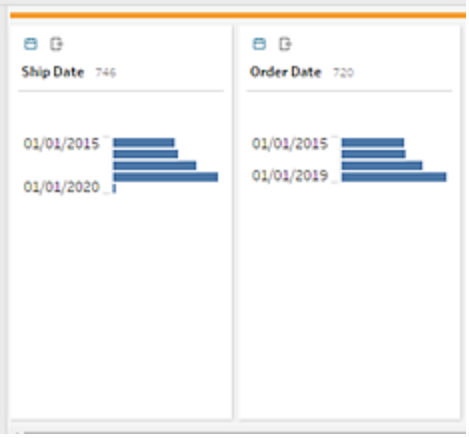
8. Now repeat steps 4 though 7 above to create a single field for **Delivery Date**. Try it on your own or use the steps below to help you.

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

   Tip: Tableau Prep summarises 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 delivery dates can now be summarised by year. Each bin represents a year from January of the starting year to January of the following year and is labelled accordingly. Because there are sales dates and delivery dates that fall in the latter part of 2018 and 2019, we get a bin for that data that is labelled with the ending year 2019 and 2020 accordingly.

To change this view to the actual dates, click the **More options** ··· menu (drop-down arrow in prior versions) in the Profile card and select **Detail**.



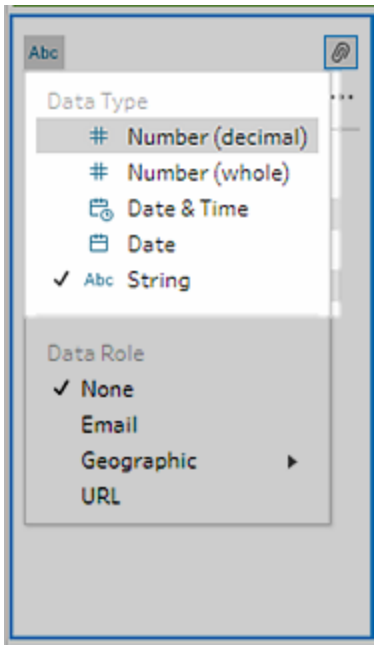
Interact directly with fields to clean your data

Your data is starting to look good. But, as you finish removing the extra fields for the order and delivery 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. To change the data type for the **Discount** field from **String** to **Number (decimal)**, click **Abc** and select **Number (decimal)** from the drop-down menu.



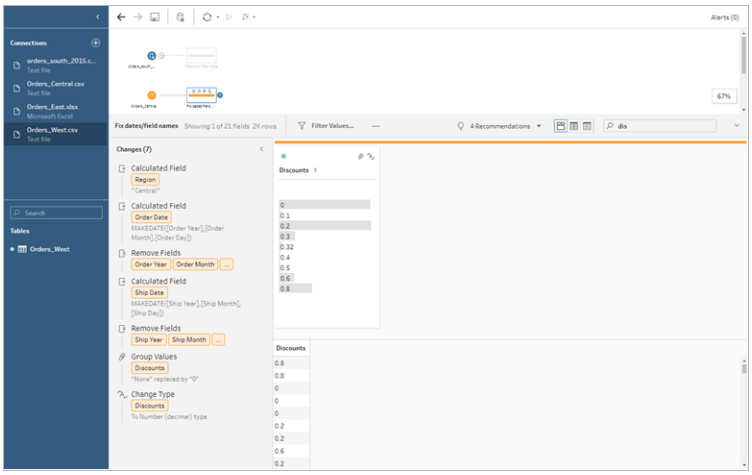
12. Finally name your step to help keep track of what you did in this step. In the **Flow** pane, double-click the step name **Clean 1** and type in **Fix dates/field names**.

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**.

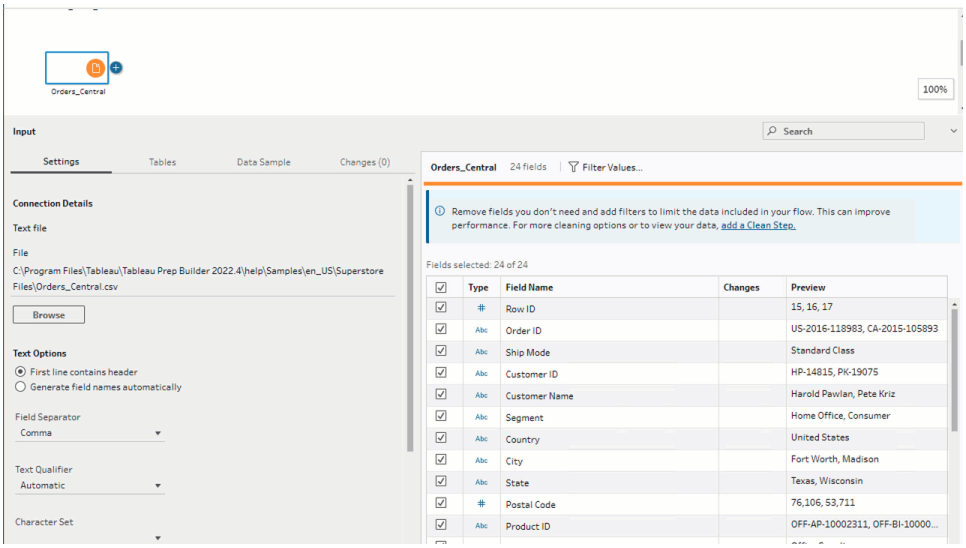
You click the arrow to open it and are delighted to see a list of every change you just made. As you scroll through the changes in the list, you notice that you can delete or edit your changes or even move them around to change the order that you did them in.

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** and do one of the following:

   - Click the plus ⊕ icon and add a clean step. Depending on your version, this menu option is **Add Step**, **Add Clean Step** or **Clean Step**.
   - Click on the suggested clean step (Tableau Prep Builder version 2020.3.3 and later and on the web).

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

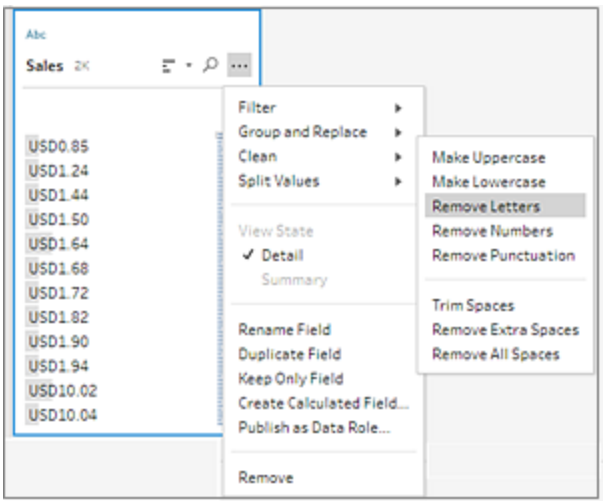
   You'll need to remove the currency code from this field and change the data type if you want to get accurate sales data.

   Fixing the data type is easy, you already know how to do that. But there are over 2000 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.

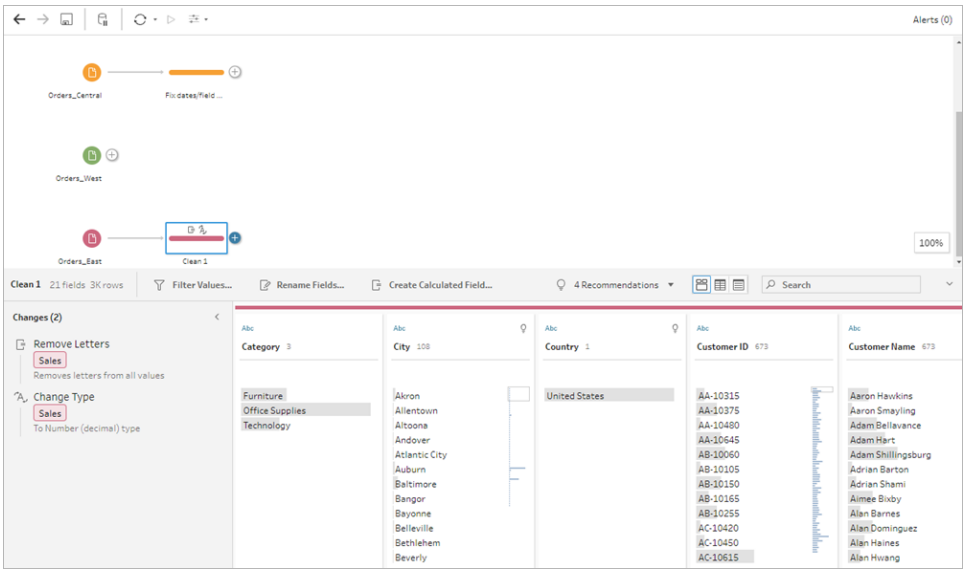
   When you click the **More options** ··· (drop-down arrow in prior versions) for the **Sales** field, you see a menu option called **Clean** and an option under that to remove letters. You decide to give that a try and see what it does.

2. Select the **Sales** field. Click the **More options** ··· menu (drop-down arrow in prior versions) 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 for the **Sales** field and select **Number (decimal)** from the drop-down list to change the data type.



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** and do one of the following:
   - Click the plus icon and add a clean step.
   - Click on the suggested clean step (Tableau Prep Builder version 2020.3.3 and later and on the web).
2. Scroll or use Search to find the **State** field.

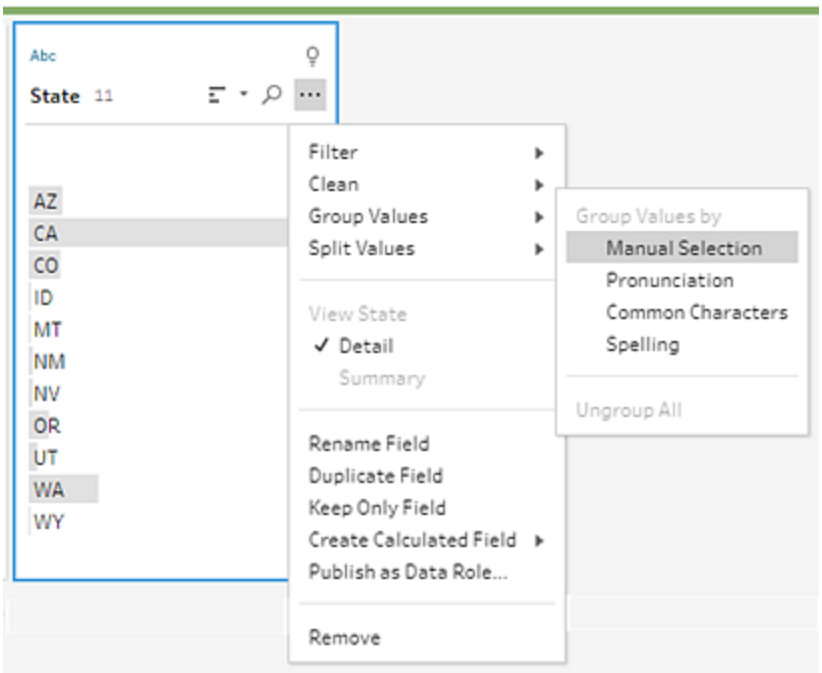
   You see that all the state name values use the short abbreviation. There are only 11 unique values for this field. You could manually change each one, but maybe Tableau Prep has another way to do this?

   You click the **More options** ··· menu (drop-down arrow in prior releases) for the field and see an option called **Group Values** (Group and Replace in previous versions). When you select it you see several options:

   - Manual Selection
   - Pronunciation
   - Common Characters
   - Spelling

   The state names don't sound alike, they aren't spelled incorrectly, and they 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 Values** option in the drop-down menu.
   For more information about editing and grouping values, see **Edit field values** on page 248.
3. Select the **State** field. Click the drop-down arrow and select **Group Values** (Group and Replace in previous versions) > **Manual Selection**.



A two column card opens. This is the **Group Values 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.

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

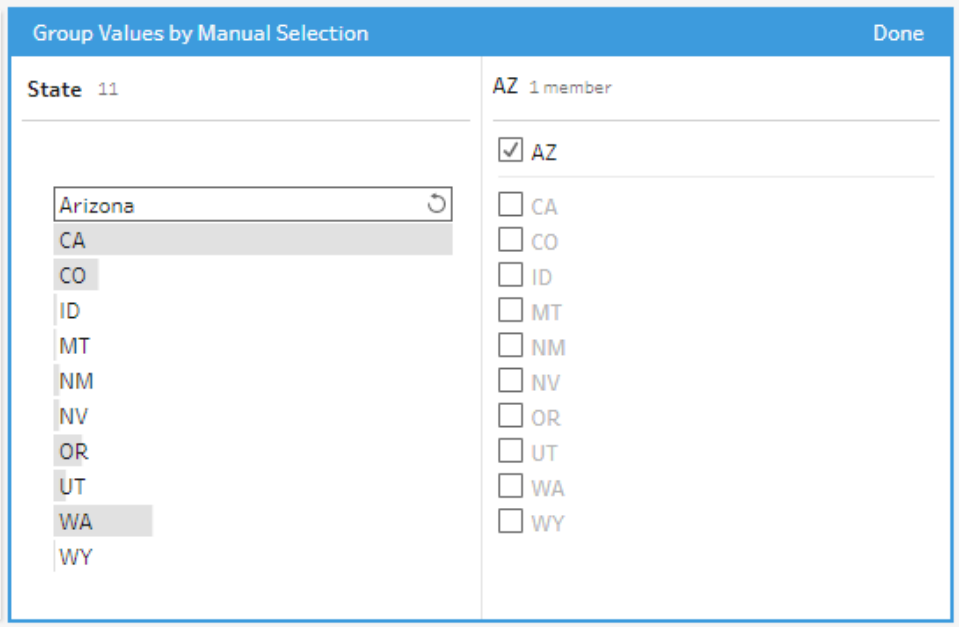


Tableau Prep created a mapped value for your new value **Arizona** and automatically mapped the old value, **AZ**, to it. Having a mapped relationship set up for these values will save you time if you get more data from this region entered like this.

Tip: You can add field values that aren't in your data sample to set up mapping relationships to organise 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.

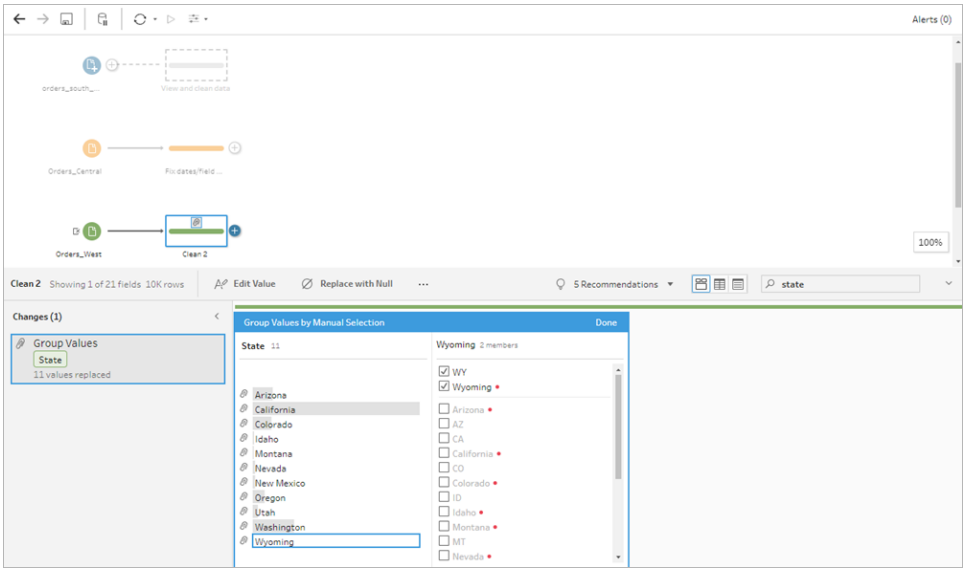
When you manually add a value that isn't in your data sample, the value is marked with a red dot to help you easily identify it.

5. 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 Values** editor.



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

Tableau Prep grouped 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.

6. 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.

> **Note**: If working on the web, your changes are automatically saved as you go, creating a draft flow. Click in the draft title to name your draft. For more information about authoring on the web, see **Tableau Prep on the Web** in the Tableau Server or Tableau Cloud help.

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 383.

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 after your clean up efforts, to pull all the rows together into a single table, you need to union the tables.

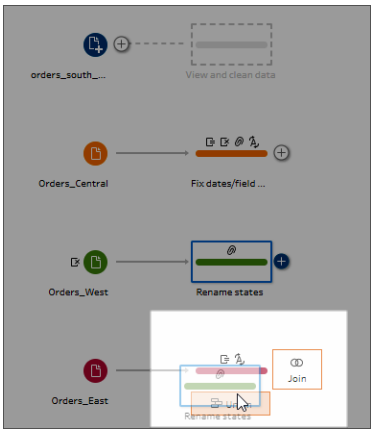
You remember that there was a step option called **Union**, but you wonder if you can simply drag and drop the steps to union them. You decide to try it and see.

Union your data

1. Follow the steps for where you are working.

   Tableau Prep Builder

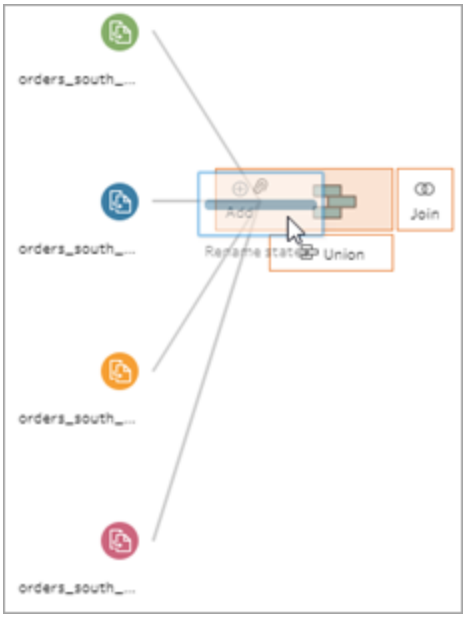
   - In the **Flow** pane, drag the cleaning step **Rename states** on to the cleaning step **Changed data type** step and drop it on the **Union** option.



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

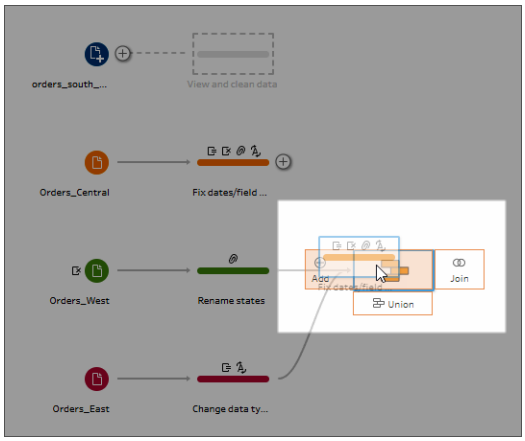
Tableau Server or Tableau Cloud

- In the **Flow** pane, drag the cleaning step **Rename states** on to the Union step you created earlier for your South files and drop it on the **Add** option.



You see that Tableau Prep added your new files to your previous union. Great! Now you want to add the other files to this union too.

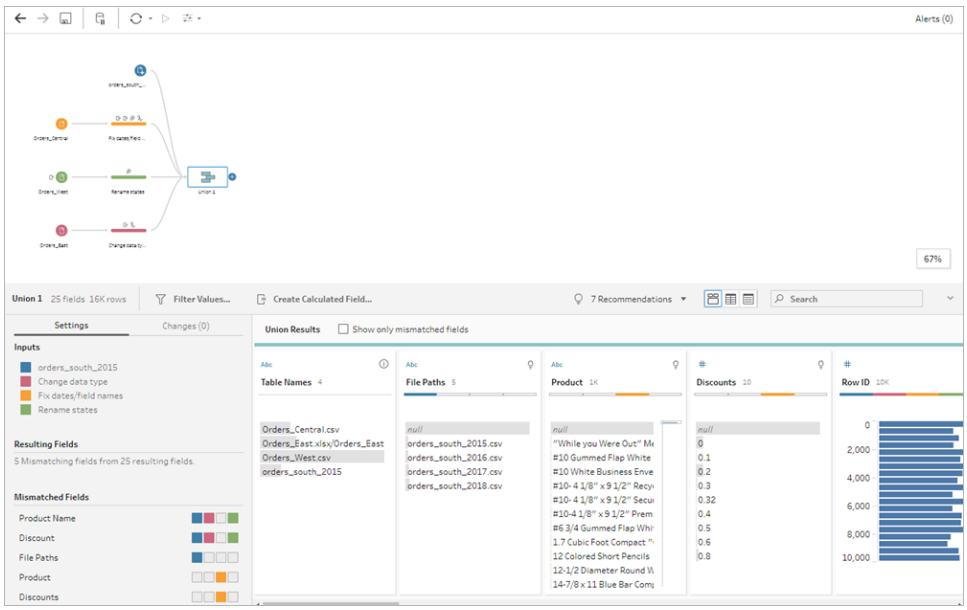
2. Drag the next cleaning step in the flow on to the **Union** step, then drop it on **Add** to add it to the existing union.



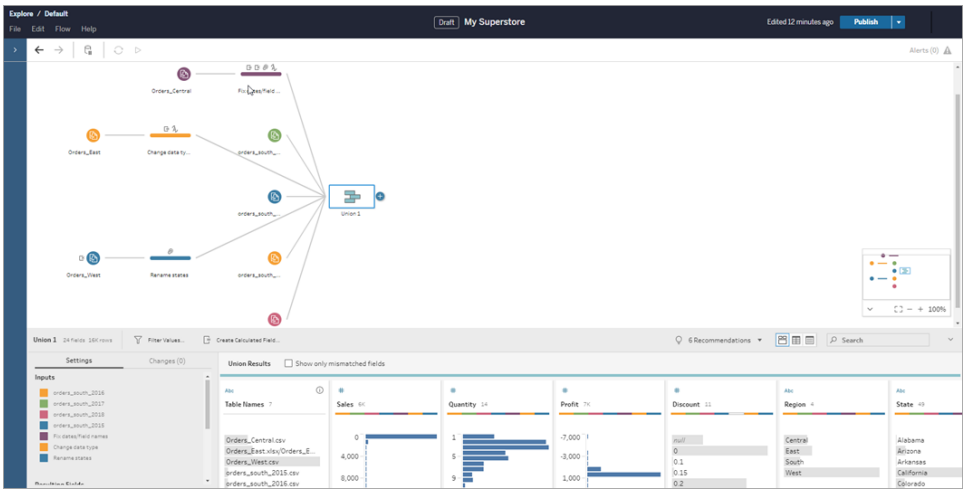
3. Drag the remaining step (**orders_south_2015** Input step if working in Tableau Prep Builder or your cleaning step if working on the web) to the new **Union** step. Drop it on **Add** to add it to the existing union.

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

   On Tableau Prep Builder:



   On Tableau Server or Tableau Cloud:



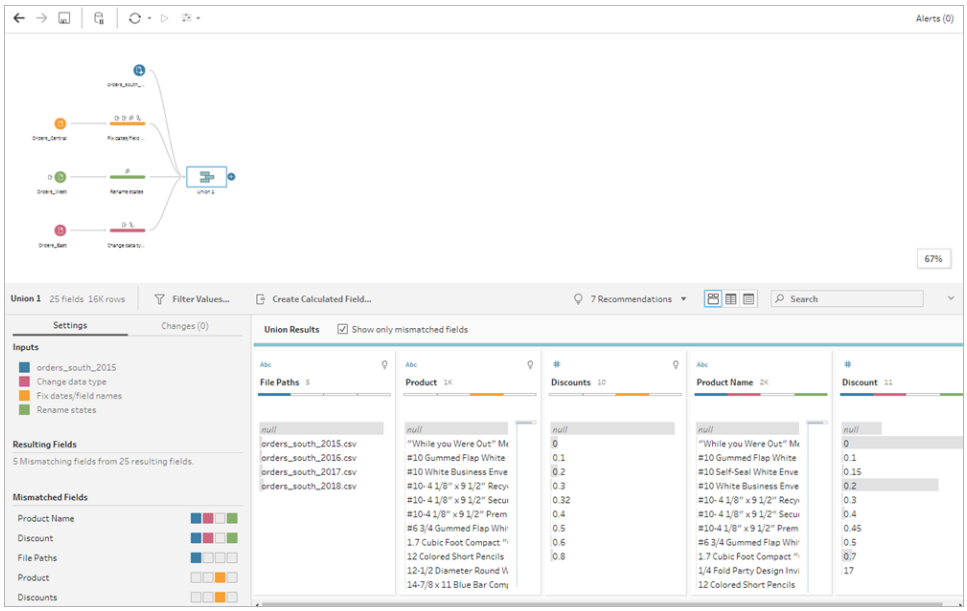
You notice that Tableau automatically matched up the fields that had the same names and types.

You also see that the **colours** 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 **coloured 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.

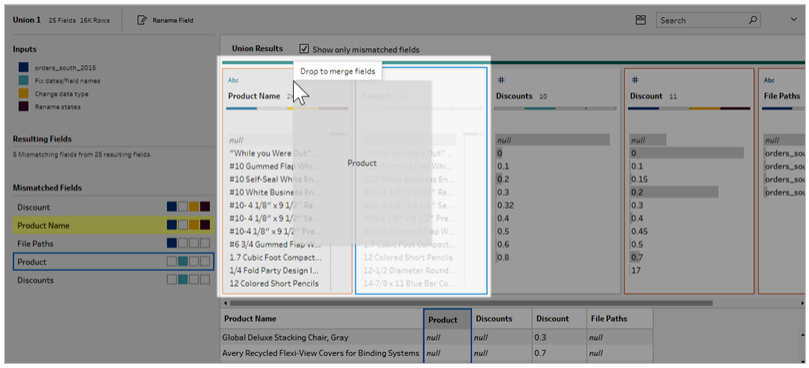
A list of mismatched fields also shows in the summary pane and you can see right away that the fields **Product** and **Discounts** only appear in the **Orders_Central** file.

4. To take a closer look at these fields, in the **Union Results** pane, tick the **Show only mismatched fields** 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. In Tableau Prep Builder, 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. If Tableau Prep detects a possible match, it will highlight it in yellow. To merge the fields hover over the

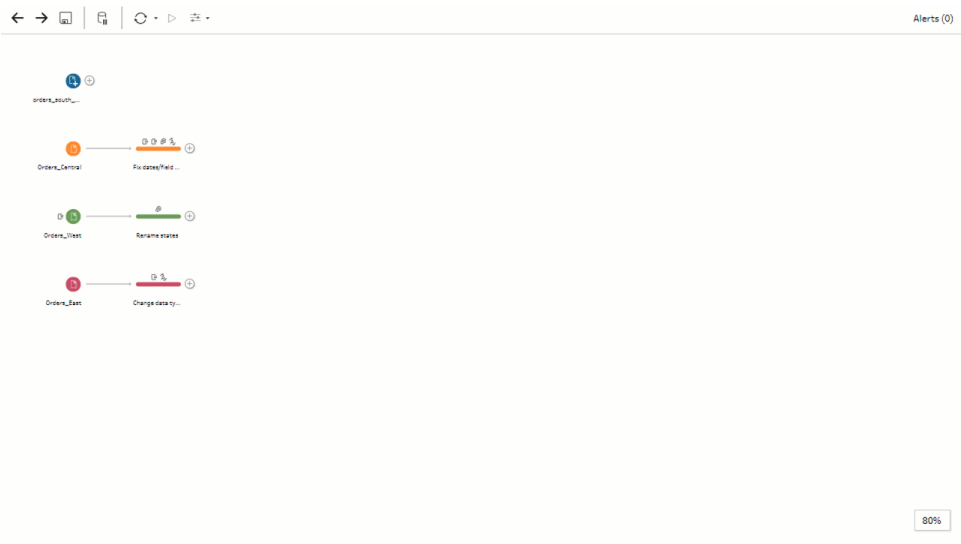
highlighted field and click the plus button that appears.

For more ways to merge fields in a union, see **Fix fields that don't match** on page 370.

7. Untick the **Show only mismatched fields** 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



You are a cleaning genius! As you are admiring your results, 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 no problem at all!

Clean the product returns data

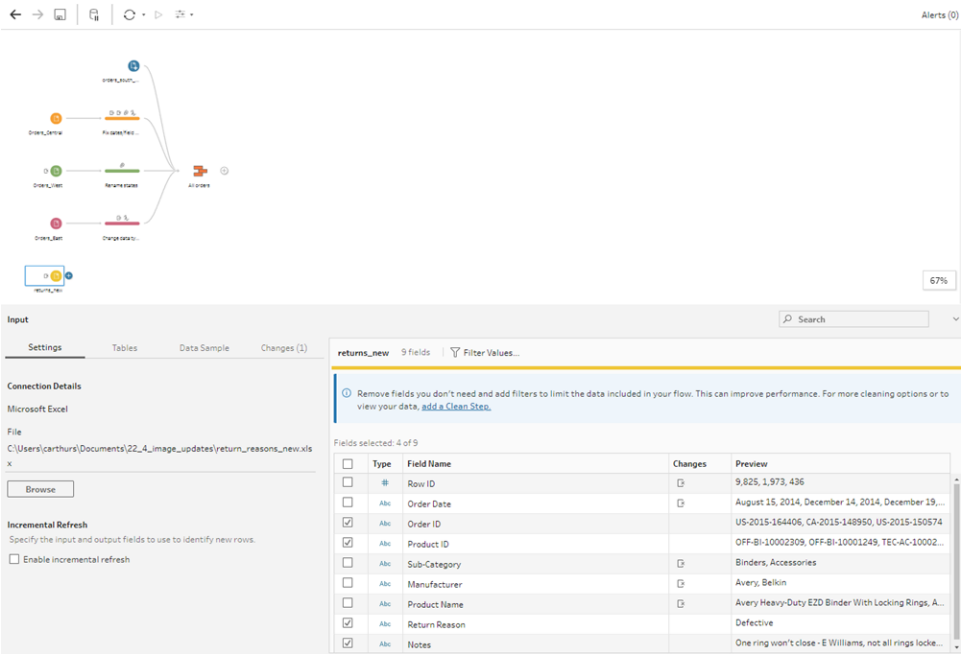
You look over the Excel file that your boss sent you for product returns and it looks a little messy. You add the new file **return_reasons new** to your flow to take a closer look.

1. In the **Connections** pane, click **Add connection**. Select **Microsoft Excel** and navigate to the sample data files you've been using for this exercise. (See **Sample files** on page 3 to download the file.)

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

   There are only four 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 tick box at the top of the left-most column to clear all the tick boxes. Then tick the 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 this input. 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 or on the suggested clean step to add a clean step.

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

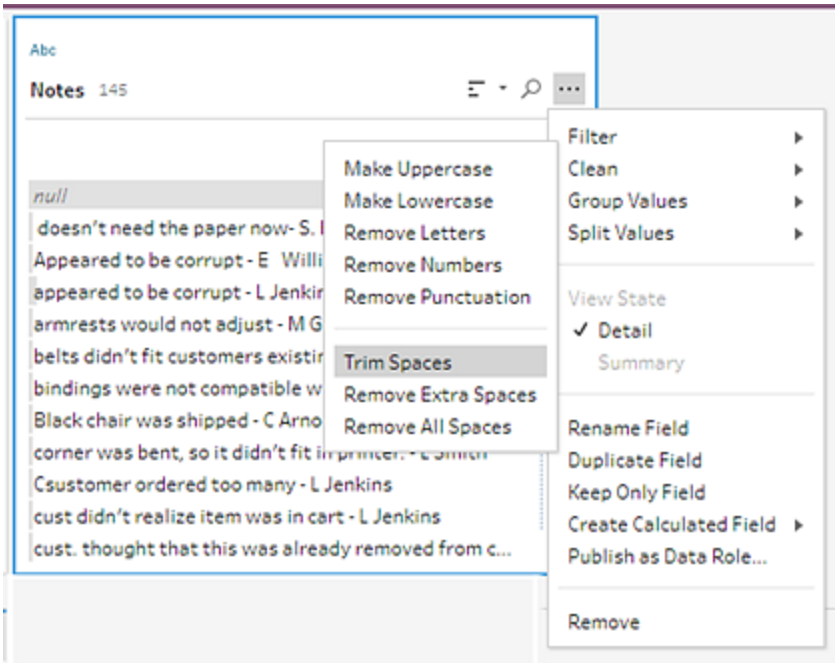
6. In the **Notes** field, use the visual scroll bar to the right of the field values to scan the values.

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 **More options** ··· menu (drop-down arrow in prior releases) and select **Clean** > **Trim Spaces**.

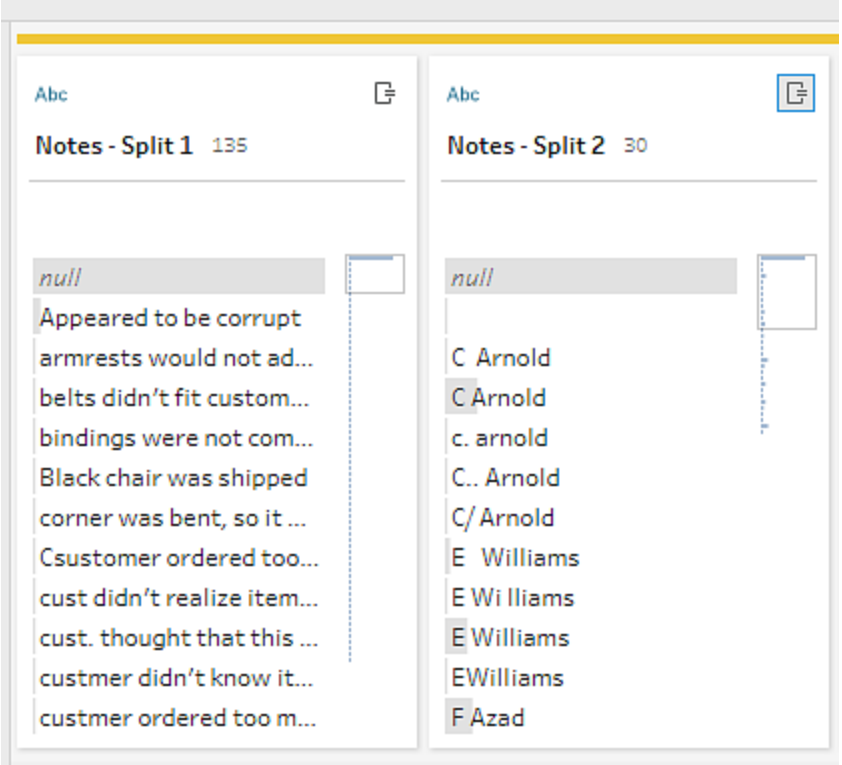


Yes! It did exactly what you wanted it to do. The extra spaces are gone.

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

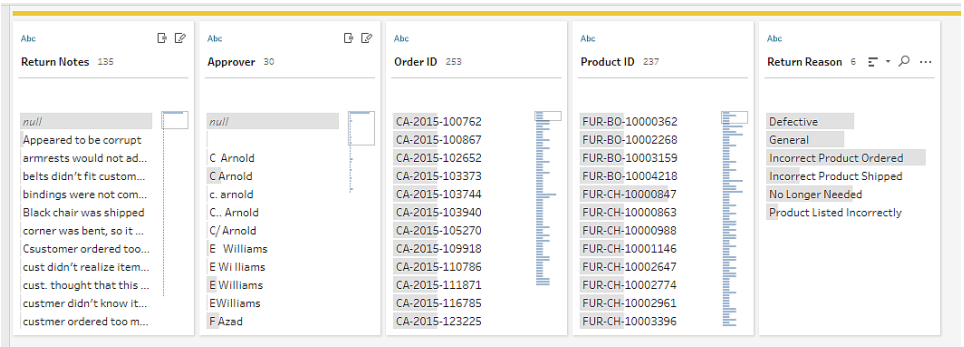
8. Select the **Notes** field. Click the **More options** ··· menu (drop-down arrow in prior releases) and select **Split Values** > **Automatic Split**.

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 **More options** ··· menu (drop-down arrow in prior versions), and select **Remove** (**Remove Field** in prior versions) from the menu.

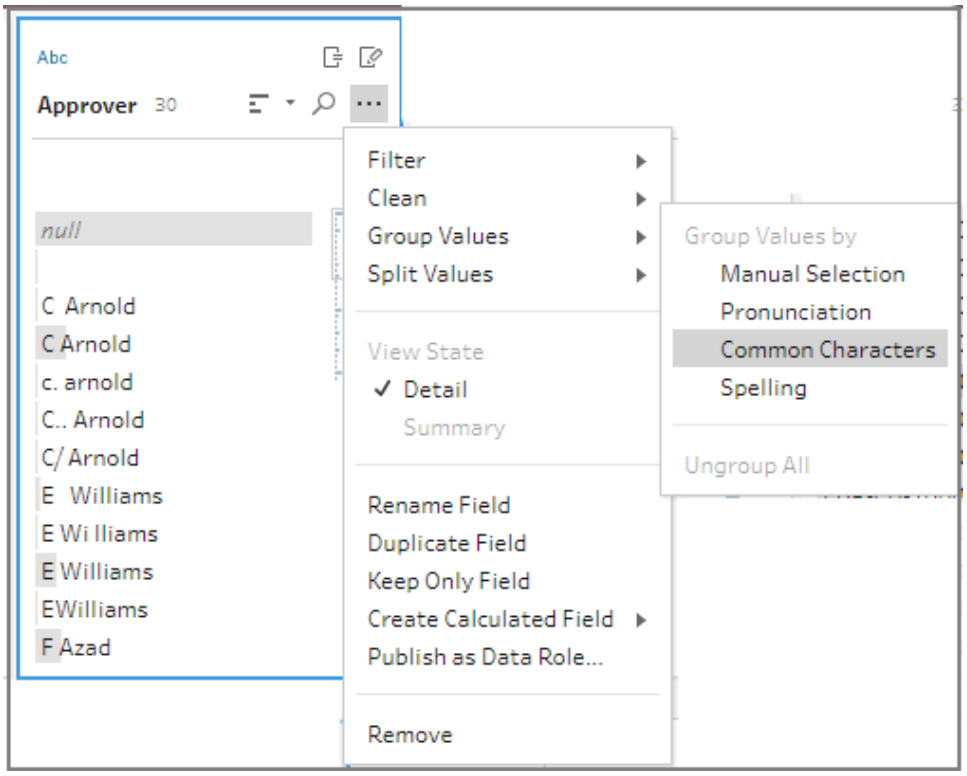


Looking at the new **Approver** field, you notice that the field values lists the same names but they are entered differently. You want to group them to eliminate multiple variations of the same value.

Maybe the **Group Values** (Group and Replace in prior versions) option can help with that?

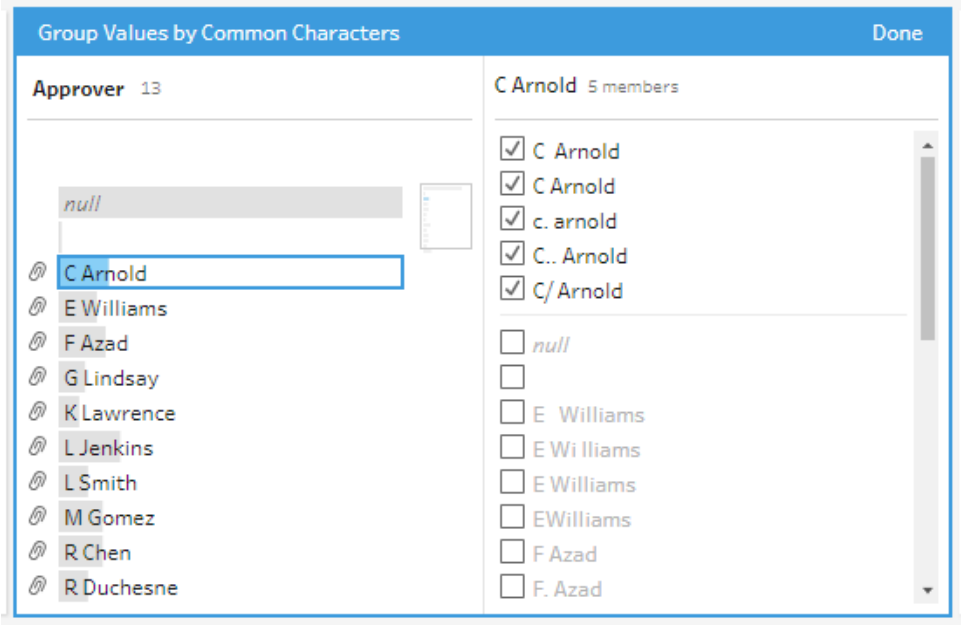
You remember there was an option for **Common Characters**. Since these values share the same letters, you decide to try that.

12. Select the **Approver** field. Click the **More options** ··· menu (drop-down arrow in prior versions) and select **Group Values** (Group and Replace in prior versions) > **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 Values editor**.



This file is looking pretty good.

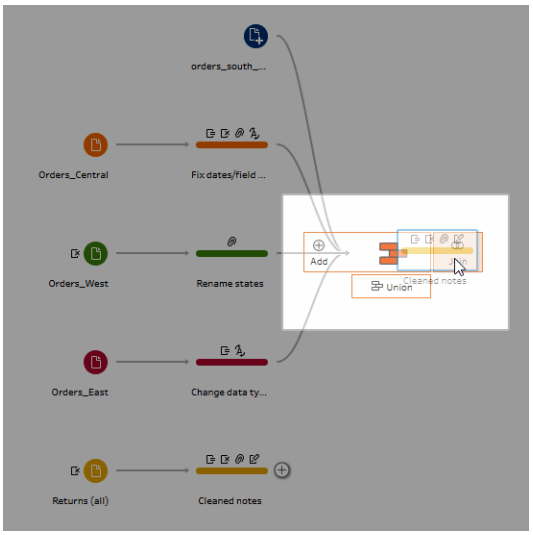
13. Name your cleaning step to keep track of your work. For example **Cleaned notes**.

Now that the product return data is all cleaned up, you want to add this data to the orders data from 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.

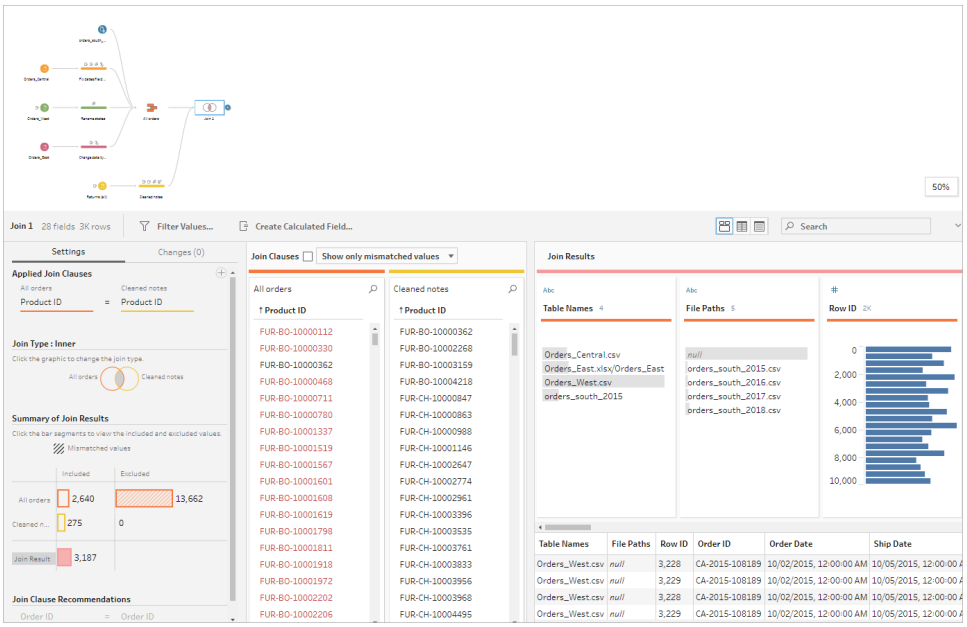
1. In the **Flow** pane, drag the **Cleaned notes** step on to the **All orders** Union step and drop it on **Join**.



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 edit values directly in the **Join Clauses** panes and perform cleaning operations in the **Join Results** pane.

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.

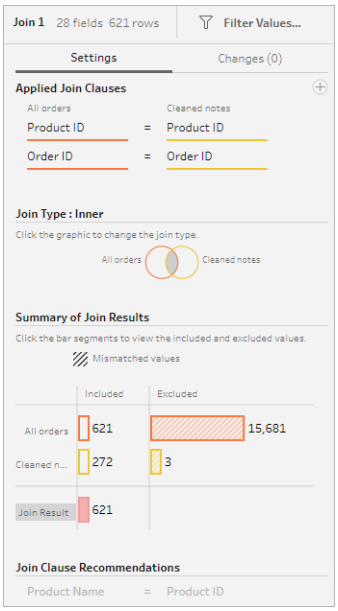
Select the fields that you want to join on in the Applied Join Clauses section or add suggested join clauses from the Join Clause Recommendations section.

For more information about working with joins, see **Aggregate, Join or Union Data** on page 359.

You see that you have over 13,000 rows excluded from your **All Orders** files. When you created your join, Tableau Prep automatically joined on the **Product ID** field, but you also wanted to join on the **Order ID** field.

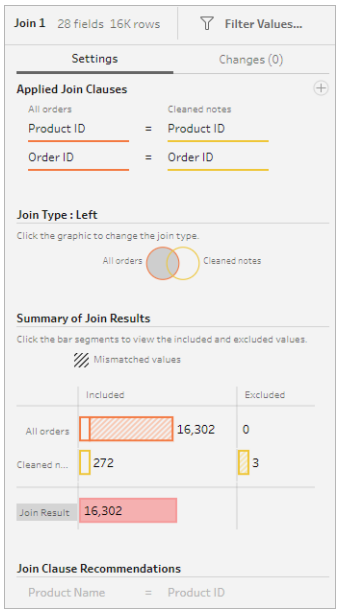
As you scan the left pane of the join profile, you see that **Order ID** is in the list of recommended join clauses, so you quickly add it from there.

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 side of the diagram to include all orders. In the example below, click the left side of the diagram to change the join type to a **Left** join and include all data from the **All orders** union step and any matching data from the **Cleaned notes** step.



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 analysing 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? From Tableau Prep Builder, 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 Builder. For more information see **View flow output in Tableau Desktop** on page 386.

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

Clean your join results

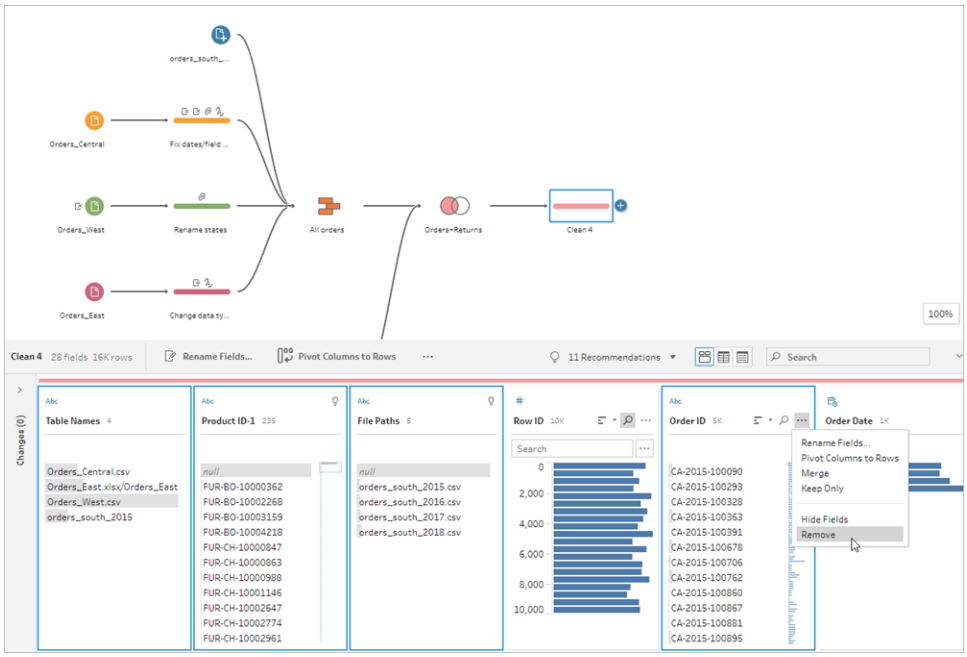
> **Note**: To clean up the fields in your join, you can perform cleaning operations directly in the Join step. For the purposes of this tutorial we will add a cleaning step so you can clearly see your cleaning operations. If you want to try performing these steps directly in the join step skip steps 1 and 3 below.

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 brings in both fields and renames 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.

1. In the **Flow** pane, select **Orders+Returns**, click the plus icon, and add a clean step.
2. In the **Profile** pane, select and remove the following fields:
    - **Table Names**
    - **Order ID**
    - **File Paths** (Tableau Prep Builder only)
    - **Product ID-1**



3. Rename the field **Order ID-1** to **Order ID**.

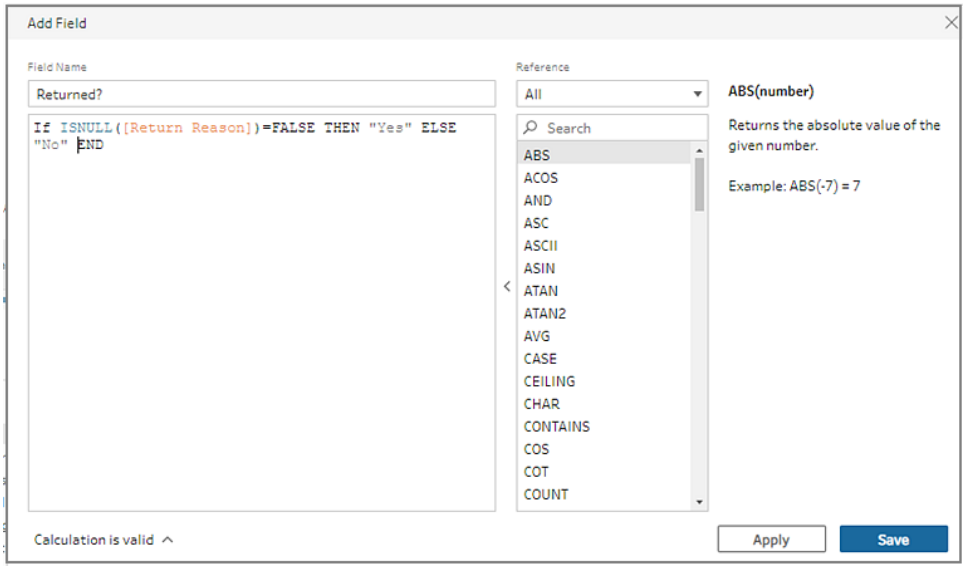
   You have quite a few null values where the product was returned but there was no return note or approver indicated. To make this data easier to analyse, you want to add a field with a value of **Yes** and **No** to indicate whether the product was returned.

   You don't have this field, but 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
```

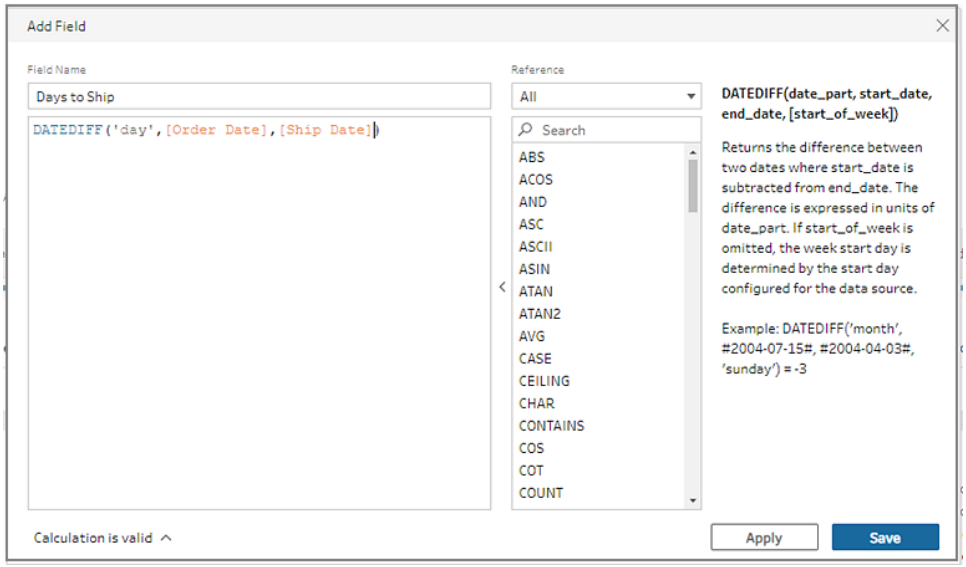


For your analysis you would also like to know the number of days it takes to deliver an order, but you don't have that field either.

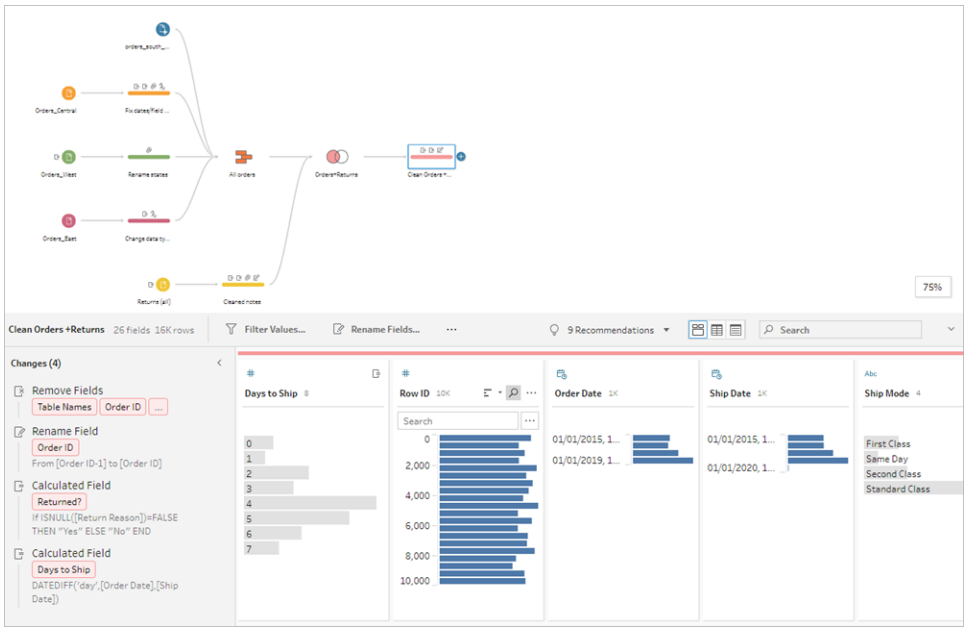
You have all the information that you need to create it though, so you add another calculated field to create it.

6. In the toolbar, click **Create Calculated Field**.
7. Name the field **Days to Deliver** 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

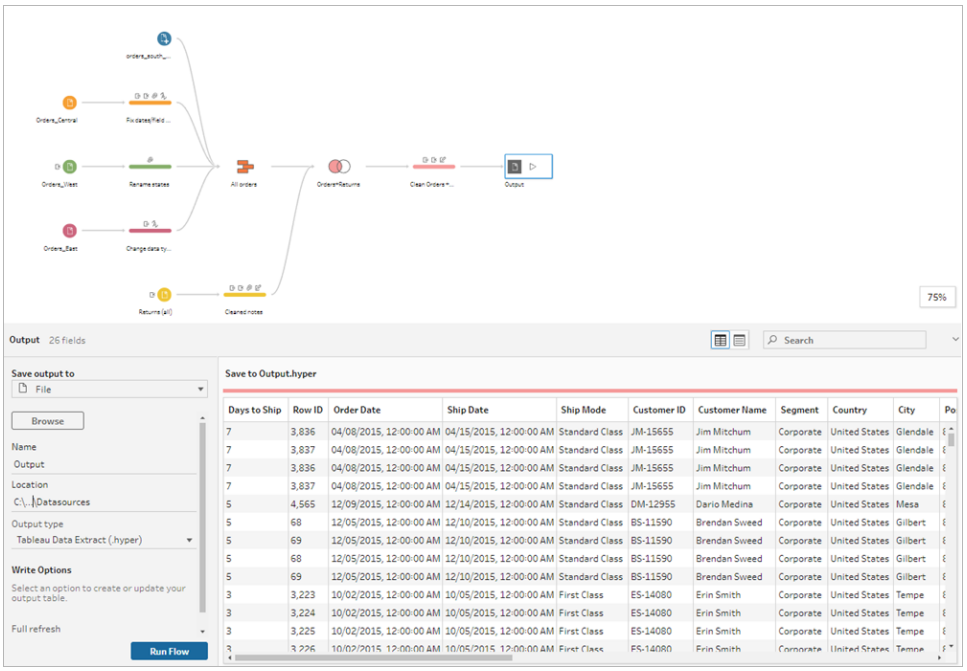
Your data is looking good and you're ready to generate your output file to start analysing it in Tableau Desktop. All you need to do is run your flow and generate your extract file. To do this you need to add an **Output** step.

Depending on where you're working, you can output your flow to a file (Tableau Prep Builder only), to a published data source or to a database.

1. In the **Flow pane**, select **Clean Orders+Returns**, click the plus icon and select **Output** (**Add Output** in prior versions).

   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 Builder** repository in your data sources folder.



2. In the left pane in the **Save output to** drop-down, depending on where you are working, do one of the following:

   Tableau Prep Builder

   a. Select **File** (select **Save to file** in previous versions).
   b. 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**.
   c. In the **Output type** field, select an output type. Select Tableau Data Extract (.hyper) for Tableau Desktop or Comma Separated Values (.csv) if you want to share the extract with a third party.

   Tableau Server or Tableau Cloud

   a. Select **Published data source**.
   b. Select a project.
   c. Enter a name for the file, for example **Orders_Returns_Superstore**.

   Tip: You have choices when generating output from your flow. You can generate an extract file (Tableau Prep Builder only), you can publish your data as a data source to Tableau Server or Tableau Cloud or you can write your data to a database. For more information about generating output files, see **Create data extract files and published data sources** on page 387.

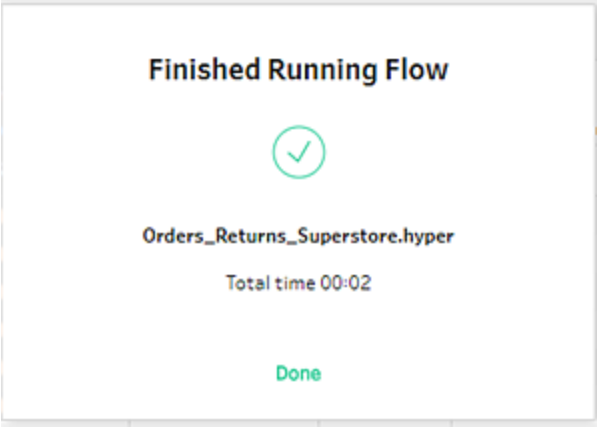
3. In the **Write Options** section, view the options to write the new data to your files. You want to use the default (**Create table**) and replace the table with your flow output, so there is nothing to change here.

   Tip: Starting in version 2020.2.1, you can choose how you want to write your flow data back to your table. You can choose from two options; **Create table** or **Append table**. By default, Tableau Prep uses the **Create table** option and overwrites your table data with the new data when you run your flow. If you choose **Append table**, Tableau Prep adds the flow data to the existing table so you can track both new and historical data on every flow run. For more information, see **Configure write options** on page 421

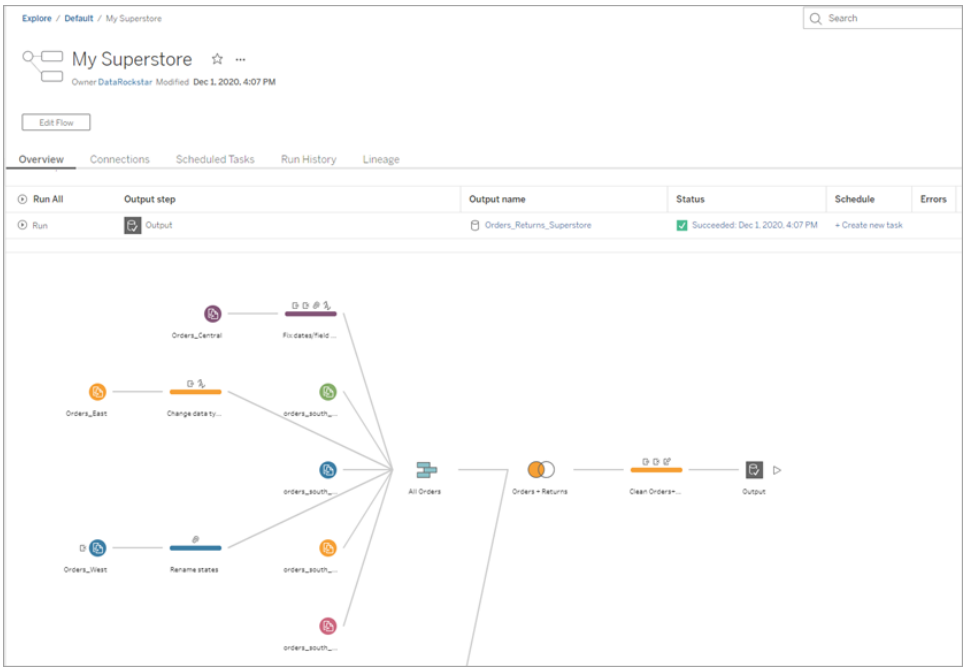
4. In the Output pane, click **Run Flow** or click the **Run Flow** ▷ button in the flow pane to generate your output.

   > **Note**: If you are working on the web, click **Publish** to publish your draft flow. Only published flows can be run.

5. 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.



   If working on the web, navigate to the **Explore**>**All Flows** page, and find your flow. You can see the status of your flow run on the Flow Overview page.



To keep your data fresh, you can run the flow manually or use the command line. If you have Data Management and have Tableau Prep Conductor enabled, you can also run your flow on a schedule in Tableau Server or Tableau Cloud.

Starting in Tableau Prep Builder version 2020.2.1 and on the web, you can also choose to refresh all your data every time the flow is run, or run your flow using incremental refresh and process only your new data each time.

For more information about keeping your data fresh, see the following topics:

- **Refresh flow output files from the command line** on page 425
- **Publish a Flow to Tableau Server or Tableau Cloud** on page 463
- **Refresh Flow Data Using Incremental Refresh** on page 416

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 data files found here:

- Orders_South_2015
- Orders_South_2016
- Orders_South_2017
- Orders_South_2018
- Orders_Central
- Orders_East
- Orders_West
- returns_reasons_new
- Quota

You can also find the files in the following location on your computer after installing Tableau Prep Builder:

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

Want more training? Check out these great resources, 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 Builder is a tool in the Tableau product suite designed to make preparing your data easy and intuitive. Use Tableau Prep Builder to combine, shape, and clean your data for analysis in Tableau.

> **Note**: Tableau Prep version 2019.1.2 had changed its name to Tableau Prep Builder and refers to the Desktop application. Starting in version 2020.4, as a Creator, you can also create and edit flows on Tableau Server and Tableau Cloud.

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 add flow steps where you can then use familiar operations such as filter, split, rename, pivot, join, union and more 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.

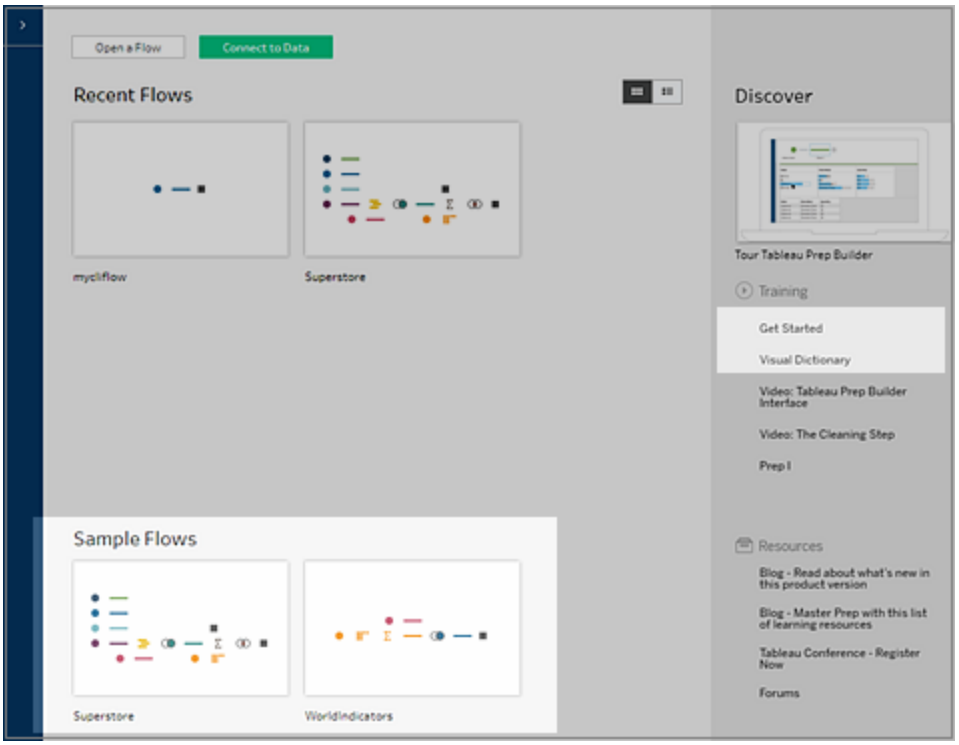
When you are finished with your flow, run it to apply the operations to the entire data set.

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 Cloud, publish your flow to Tableau Server or Tableau Cloud to continue editing on the web or refresh your data using a schedule. You can also open Tableau Desktop directly from within Tableau Prep Builder to preview your data.

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

Watch a video: See Tableau Prep Builder in action

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 Builder** on page 3 hands-on tutorial to learn how to create a flow or try stepping through one of our **Day in the Life Scenarios** on page 484 using Tableau Prep Builder.



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

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

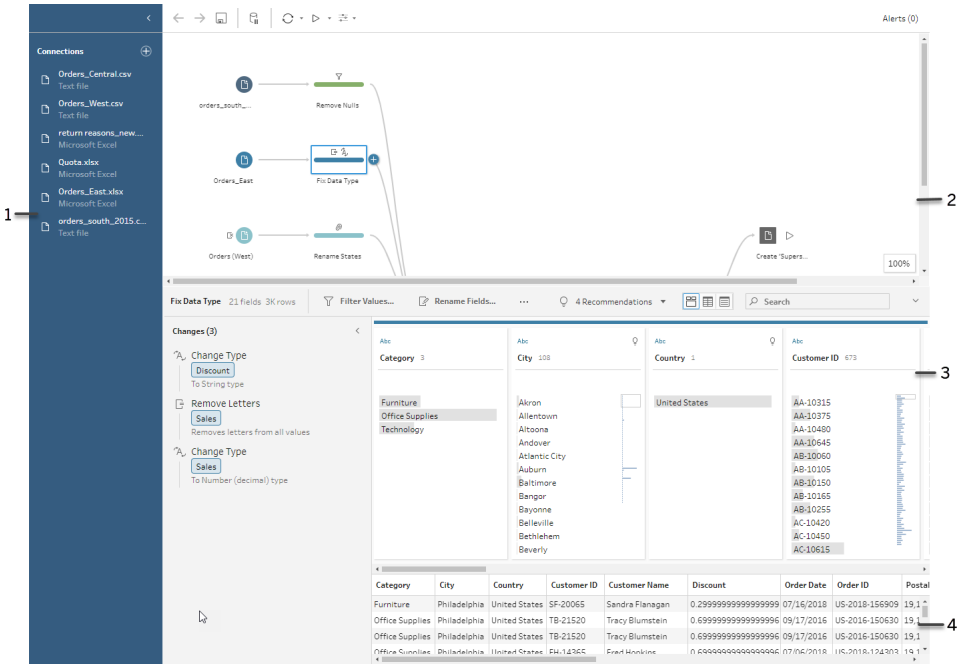
To learn more about how Tableau Prep Builder optimises your data for performance, see Tableau Prep under the hood. To learn more about Tableau Prep and the different features and functions it offers, review the topics in this guide.

A tour of the Tableau Prep workspace

The Tableau Prep workspace consists of the **Connections pane** (1) where you connect to your data sources, and three coordinated areas that help you interact with and explore your data:

- **Flow pane** (2): A visual representation of your operation steps as you prepare your data. This is where you add steps to build your flow.

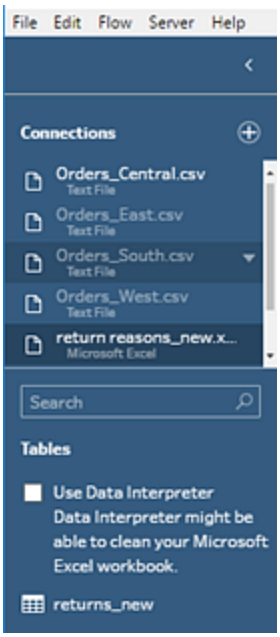
- **Profile pane** (3): A summary of each field in your data sample. See the shape of your data and quickly find outliers and null values.
- **Data grid** (4): The row-level detail for your data.



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 (1)

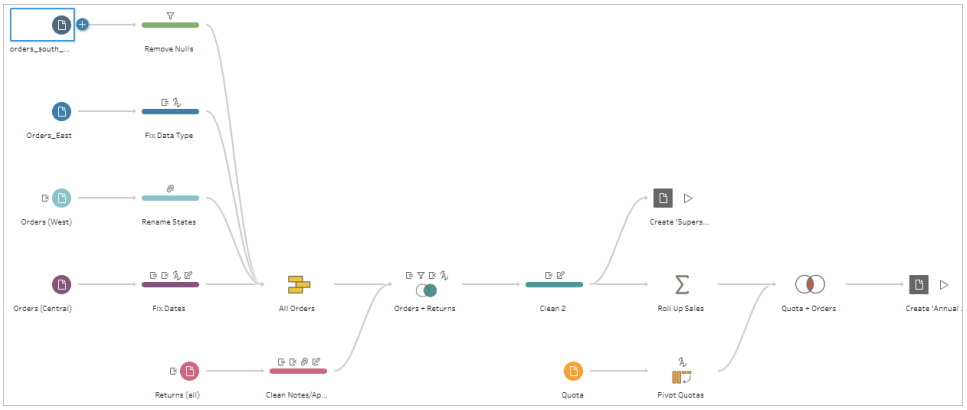
On the left side of the workspace is the **Connections** pane, which shows the databases and files you are connected to. Add connections to one or more data sources and then drag the tables you want to work with into the **Flow** pane. For more information see **Connect to Data** on page 75.



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

Flow pane (2)

At the top of the workspace is the **Flow** pane. This is where you'll build your flow. 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.



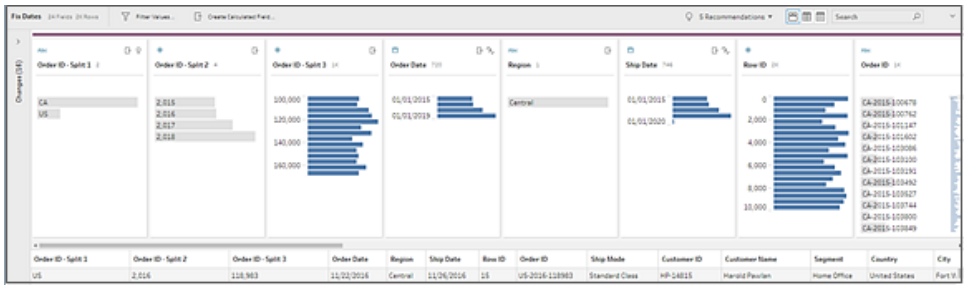
You start your flow by dragging tables into the **Flow** pane. Here you can add additional data sets; pivot your data, union or join data; create aggregations; and generate your flow output to a

file (.hyper, .csv, .xlsx), published data source that you can use in Tableau, database or to CRM Analytics. For more information about generating output files, see **Save and Share Your Work on page 383**.

> **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 Builder.

Profile pane (3)

In the centre 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.

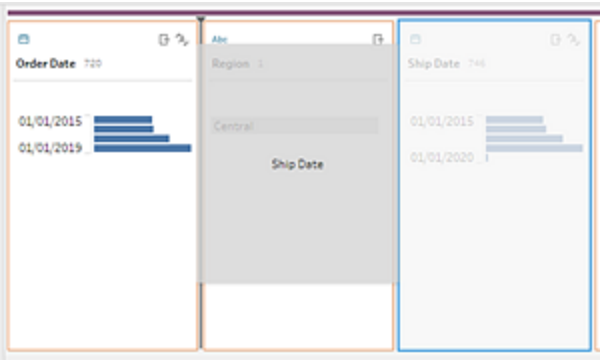


At the top of the **Profile** pane is a toolbar that shows you the cleaning operations that you can perform for each step in your flow. An options menu also appears on each card in the **Profile** pane where you can select the different operations that you can perform on the data.

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 values 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



Select one or more field values in a Profile card and right-click or Ctrl-click (MacOS) to see additional options to keep or exclude values, group selected values or replace values with Null.

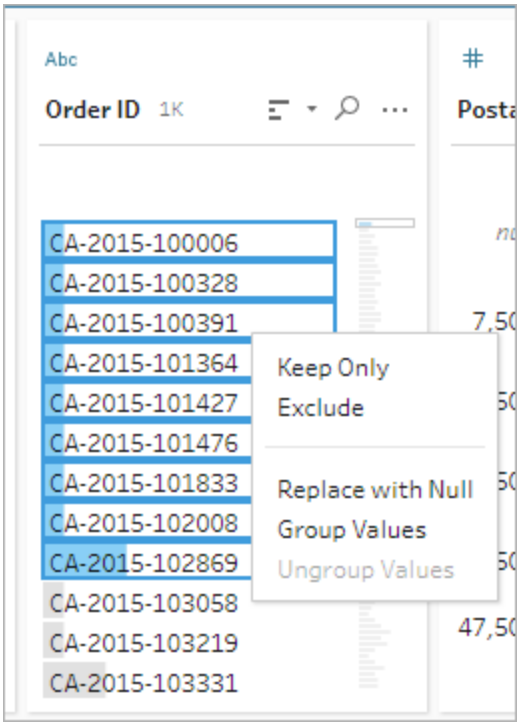
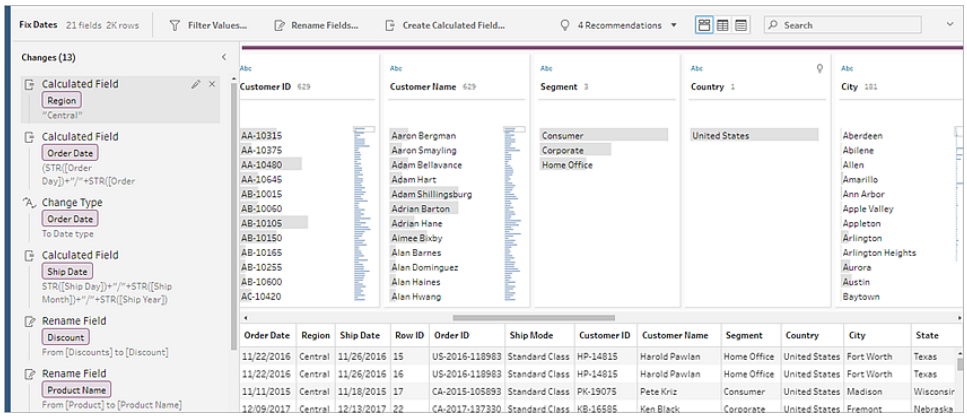


Tableau Prep keeps track of any changes you make, in the order you make them, so you can always go back and review or edit those changes if needed. Use drag and drop to re-order the operations in the list to experiment and apply changes in a different order.



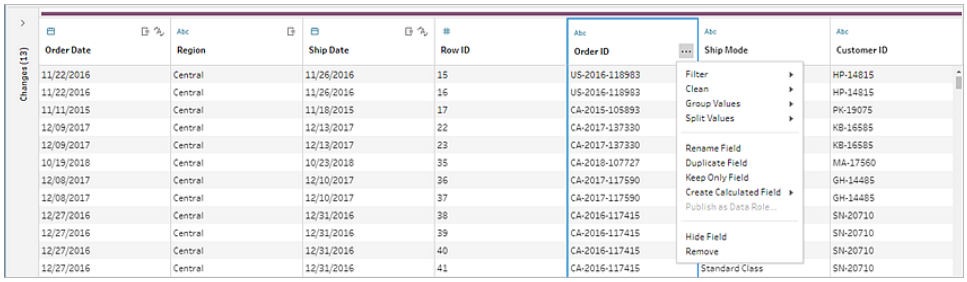
Click the arrow on the upper right of the pane to expand and collapse the **Changes** pane for more room to work with the data in the **Profile** pane.

For more information about applying cleaning operations to your data see **Clean and Shape Data** on page 227.

Data grid (4)

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 Prep BuilderXXXXX, where XXXXX represents a universally unique identifier (UUID). After you save the flow, the file is deleted. For more information about how Tableau Prep samples your data, see **Set your data sample size on page 121**.

Tableau Prep Builder 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 Values (on data entry)
- Calculations

Tableau Prep on the Web

Internet Explorer 11 on Windows and compatibility mode for Internet Explorer is not supported.

Starting in version 2020.4, Tableau Prep supports web authoring for flows. Now you can create flows to clean and prepare your data using Tableau Prep Builder, Tableau Server or Tableau Cloud. You can also manually run flows on the web and the Data Management is not required.

While most of the same Tableau Prep Builder functionality is also supported on the web, there are a few differences when creating and working with your flows.

Important: To create and edit flows on the web you must have a Creator licence. Data Management is only required if you want to run your flows on a schedule using Tableau Prep Conductor. For more information about configuring and using Tableau Prep Conductor, see Tableau Prep Conductor in the Tableau Server or Tableau Cloud help.

Installation and Deployment

To enable users to create and edit flows on the web, you'll need to configure several settings on your server. For more information about each of these settings, see Create and Interact with Flows on the Web.

- **Web Authoring**: Enabled by default, this option controls whether users can create and edit flows on Tableau Server or Tableau Cloud.

- **Run Now**: Controls whether users or only administrators can run flows manually using the **Run Now** option. The Data Management isn't required to run flows manually on the web.
- **Tableau Prep Conductor**: If Data Management is licensed, enable this option to let users schedule and track flows.
- **Tableau Prep Extensions** (version 2021.2.0 and later): Controls whether users can connect to Einstein Discovery to apply and run predictive models against data in their flow.
- **Autosave**: Enabled by default, this feature automatically saves a user's flow work every few seconds.

Sample data and processing limits

To maintain performance while working with flows on the web, limits are applied to the amount of data you can include in a flow.

The following limits apply:

- When connecting to files, the maximum file size is 1GB.
- The data sampling option to include all data is not available. The default sample data limit is 1 million rows.
- The maximum number of rows that a user can select when using large data sets is configured by the administrator. As a user, you can select the number of rows up to that limit.

For more information about setting your data sample, see Set your data sample size in the Tableau Prep help section.

Available features on the web

When you create and edit flows on the web you may notice a few differences in navigation and the availability of certain features. While most features are available across all platforms, some features are limited or not yet supported in Tableau Server or Tableau Cloud. The following table identifies features where differences might apply.

Feature area	Exceptions	Tableau Prep Builder	Tableau Server	Tableau Cloud
Connect to Data	Some connectors may not be supported on the web. Open the **Connect** pane on your server to see supported con-	✓	✓	✓

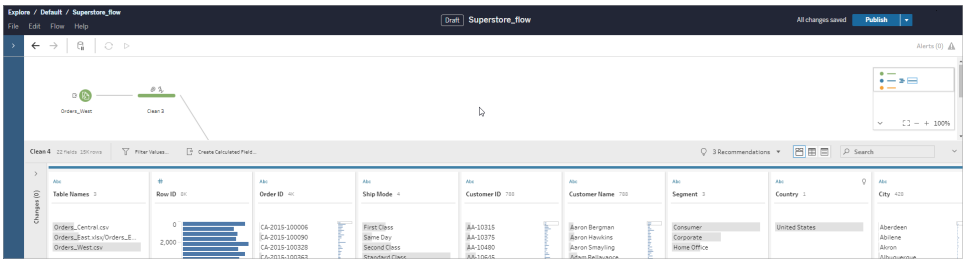
	nectors.			
Build and Organise your Flow		✓	✓	✓
Set your data sample size	In Tableau Server and Tableau Cloud, the data sample size is subject to limits set by your administrator	✓	✓	✓
Union files and database tables in the input step	Input unions can't be edited or created in Tableau Server or Tableau Cloud. Only in Tableau Prep Builder.	✓	✓	✓
Clean and Shape Data		✓	✓	✓
Copy data grid values	Available in Tableau Prep Builder and Tableau Server starting in version 2022.3 and Tableau Cloud starting in 2022.2 (August)	✓	✓	✓
Aggregate, Join or Union Data		✓	✓	✓
Use R and Python Scripts in your Flow	Script steps can't be added when creating or editing a flow in Tableau Cloud. This is currently supported only in Tableau Prep builder and Tableau Server.	✓	✓	⊘
Create reusable flow steps		✓	⊘	⊘
Automatically save your flows		Not Applicable	✓	✓

on the web				
Automatic file recovery		✓	Not Applicable	Not Applicable
View flow output in Tableau Desktop		✓	⊘	⊘
Create an extract to a file		✓	⊘	⊘
Create an extract to a Microsoft Excel worksheet		✓	⊘	⊘
Connect to a Custom SQL Query		✓	✓	✓
Create a published data source		✓	✓	✓
Save flow output to external databases		✓	✓	✓
Add Einstein Discovery Predictions to your Flow		✓	✓	✓

Autosave and working with drafts

When you create or edit flows on the server, your work is automatically saved as a draft every few seconds so that in the event of a crash, or when closing a tab by accident, you don't lose your work.

Drafts are saved to the server and project you are signed into. You can't save or publish a draft to another server, but you can save the flow to another project on that server using the **File** > **Publish As** menu option.



Draft content can only be seen by you until you publish it. If you publish changes and need to revert them, you can use the **Revision History** dialog to view and revert to a previously published version. For more information about saving flows on the web, see Automatically save your flows on the web.

Publishing flows on the web

Whether you create a flow from scratch on the web or edit an existing flow, before you can run the flow, you'll need to publish it.

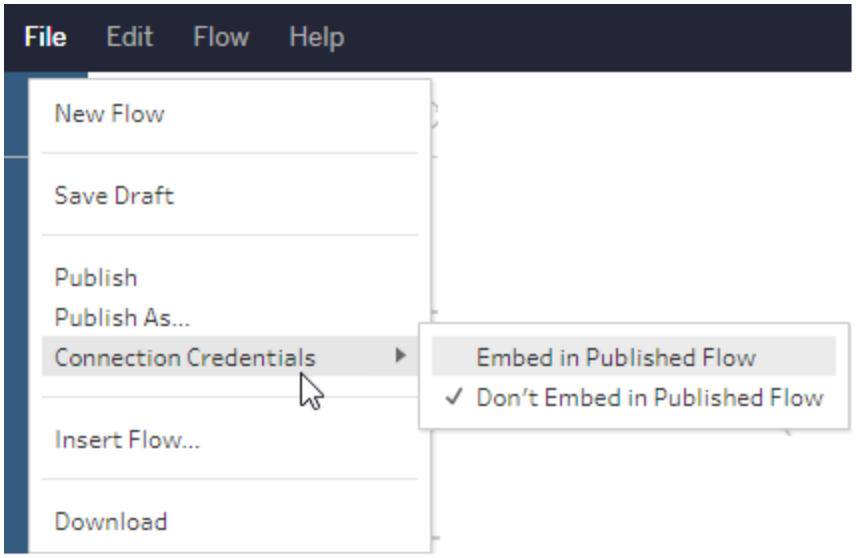
- You can only publish draft flows to the same server you are signed into.
- You can publish a draft to a different project using the **File** menu and selecting **Publish As**.
- You can embed credentials for your flow's database connections to enable the flow to run without having to manually enter the credentials when the flow runs. If you open the flow to edit it, you'll need to re-enter your credentials.

Embed credentials

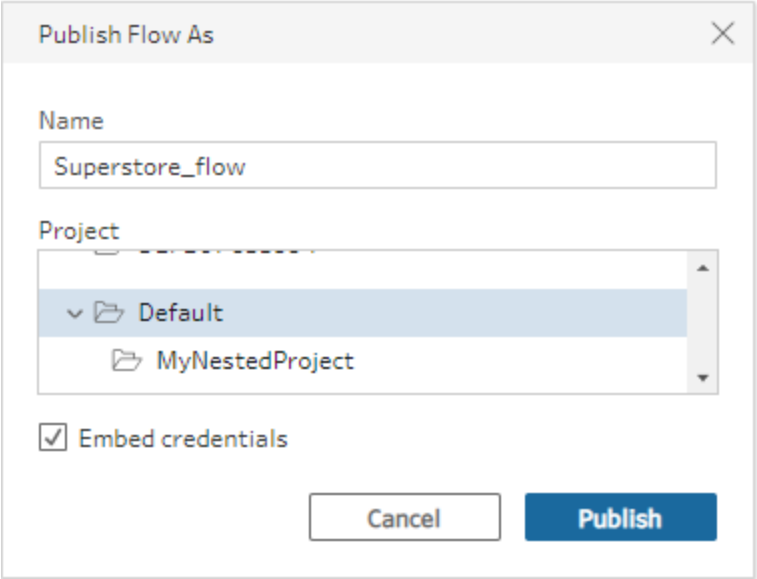
Embedding credentials only applies to running flows on your server. Currently, you will manually need to enter your credentials when editing a flow connected to a database. Embedding credentials can only be set at the flow level and not at the server or site level.

Do one of the following:

- From the top menu, select **File** > **Connection Credentials** > **Embed in Published Flow**.



- When publishing a flow, select the **Embed credentials** tick box. This option shows when you select **Publish As** to publish the flow to a new project for the first time or when you are editing a flow that was last published by someone else.



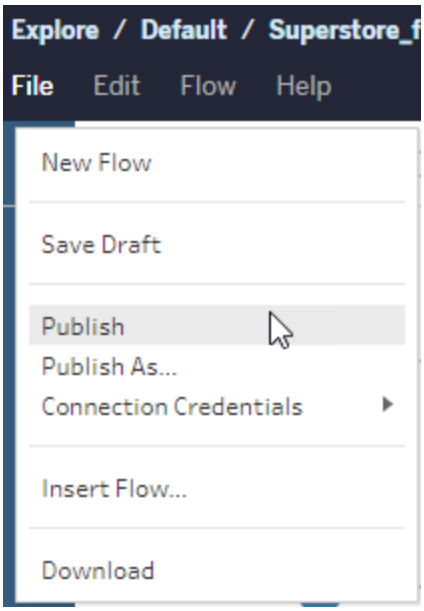
Publish a flow

When you publish your flow, it becomes the current version of the flow and can be run and seen by others who have access to your project. Flows that are never published or flow changes that

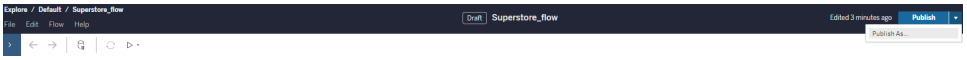
you make to a draft can only be seen by you until you publish the flow. For more information about flow statuses, see Automatically save your flows on the web.

To publish your flow, do one of the following:

- From the top menu, select **File** > **Publish** or **File** > **Publish As**



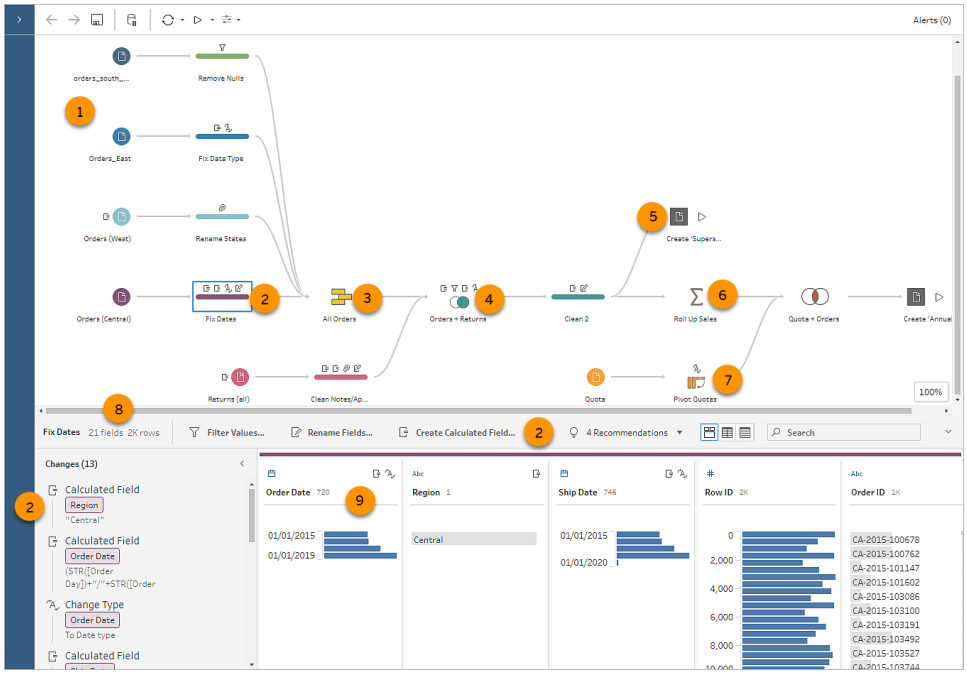
- From the top bar, click the **Publish** button or click the drop arrow to select **Publish As**.



Who can do this

- Server Administrator, Site Administrator Creator and Creator allow full connecting and publishing access.
- Creator can perform web authoring tasks.

Tableau Prep Visual Dictionary



1	**Input Step** Start your flow by dragging tables or files to the Flow pane to create an Input step. Icons show you the type of data source. Data Source Data Source with input union File (CSV, Excel, Extract, PDF, Spatial, etc) File with input union

2	**Clean Step, Changes Pane, Toolbar** Annotations help you keep track of changes you make to your data. While most changes can be made in a Clean step, you can also find these icons on other step types, in the Changes pane, and on the toolbar. Calculated Field Change Data Type Rename Field Group Values Filter, Exclude, or Keep Only Values Remove Fields Hidden Field (only visible in List view) Recommendations (suggested cleaning options) Merge Fields Show Profile Pane Show Data Grid Show List View Search
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 to run your flow, apply changes to your complete data set, and generate output data to use as a data source or share with others. File Published Data Source Database or CRM Analytics Run Flow
6	**Aggregate Step** Add an Aggregate step to group and aggregate your data, which changes the level of detail of your data. Aggregate Data
7	**Pivot Step** Add a Pivot step to change columns to rows. Pivot Data

8	**Profile Pane** See the exact row count of your data and know when your data is sampled. Sampled — Only shows when data is sampled 232K Rows 231660 — Hover over row summary to see exact row count
9	Profile Card Change data types and apply cleaning operations to a selected field in the Profile pane. Data Types Number (Decimal and Whole) Date Date and Time Text Cleaning Operations Calculated Field, Split, Duplicate, Clean Change Data Type Rename Field Group Values Filter, Exclude, or Keep Only Values Remove Fields Keep Only Field Hidden Field (only visible in List view) Merge Fields Search

Start or Open a Data Flow

To start preparing your data with Tableau Prep Builder, you can:

- Start a new flow
- Open an existing flow

> **Note**: Starting in version 2020.4.1, you can also create and edit flows in Tableau Server and Tableau Cloud. Information in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

You can open multiple Tableau Prep Builder workspaces to work on multiple flows at the same time. In Tableau Prep Builder version 2019.3.1 and earlier, if you select **File** > **Open**, Tableau Prep Builder replaces your current open flow with the new flow you select.

Start a new flow

Start a new flow by connecting to your data, just like in Tableau Desktop.

> **Notes**: 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.
> 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.

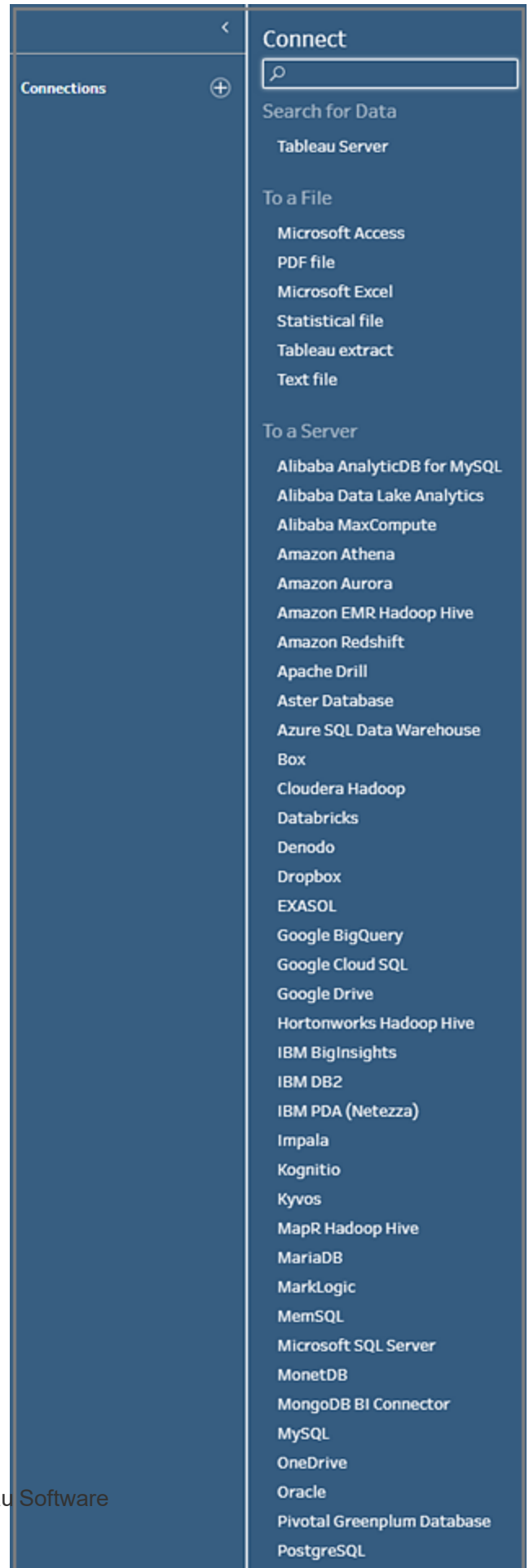
1. Open Tableau Prep Builder and click the **Add connection** button.

   In web authoring, from the **Home** page, click **Create** > **Flow** or from the **Explore** page, click **New** > **Flow**. Then click **Connect to Data**.

   Starting in version 2021.4, if you have the Data Management with Catalogue enabled, you can also click **New** > **Flow** from the **External Assets** page on the web to create a flow with a Catalogue-supported connection. For more information, see **Tableau Catalog** in the Tableau Server or Tableau Cloud help.

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.

> **Note**: In web authoring, the list of file connectors may differ.

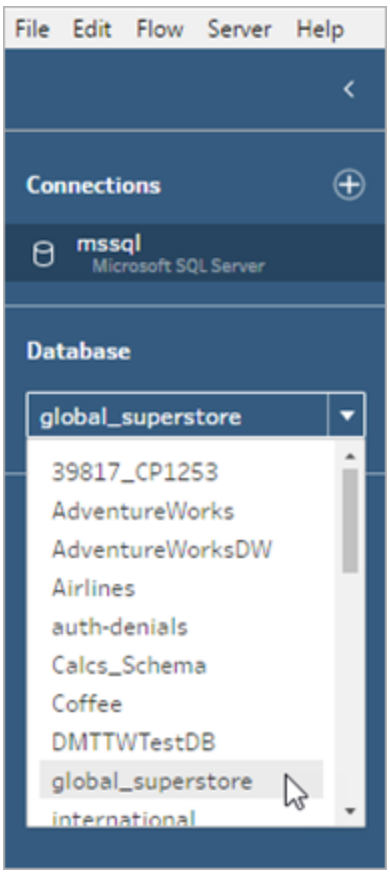

Connections
Connect
Search for Data
Tableau Server
To a File
Microsoft Access
PDF file
Microsoft Excel
Statistical file
Tableau extract
Text file
To a Server
Alibaba AnalyticDB for MySQL
Alibaba Data Lake Analytics
Alibaba MaxCompute
Amazon Athena
Amazon Aurora
Amazon EMR Hadoop Hive
Amazon Redshift
Apache Drill
Aster Database
Azure SQL Data Warehouse
Box
Cloudera Hadoop
Databricks
Denodo
Dropbox
EXASOL
Google BigQuery
Google Cloud SQL
Google Drive
Hortonworks Hadoop Hive
IBM BigInsights
IBM DB2
IBM PDA (Netezza)
Impala
Kognitio
Kyvos
MapR Hadoop Hive
MariaDB
MarkLogic
MemSQL
Microsoft SQL Server
MonetDB
MongoDB BI Connector
MySQL
OneDrive
Oracle
Pivotal Greenplum Database
PostgreSQL

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.

     > **Note**: In web authoring, for file connections, you can only download the files one at a time. Direct connections to a file network share isn't currently supported.

   - 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.



> **Note**: In Tableau Prep Builder, you can union multiple files or database tables from a single data source in the input step using a wildcard search. In

web authoring you can't create or edit input unions but they are supported in flows published from Tableau Prep Builder. For more information, see **Union files and database tables in the Input step** on page 130.

Open an existing flow

In Tableau Prep Builder, you can see and access your most recent flows right on the Start page, so it's easy to find your work in progress. When working with flows on the web, all your flows are conveniently listed on the **Explore** page under the **All Flows** menu. Open a flow

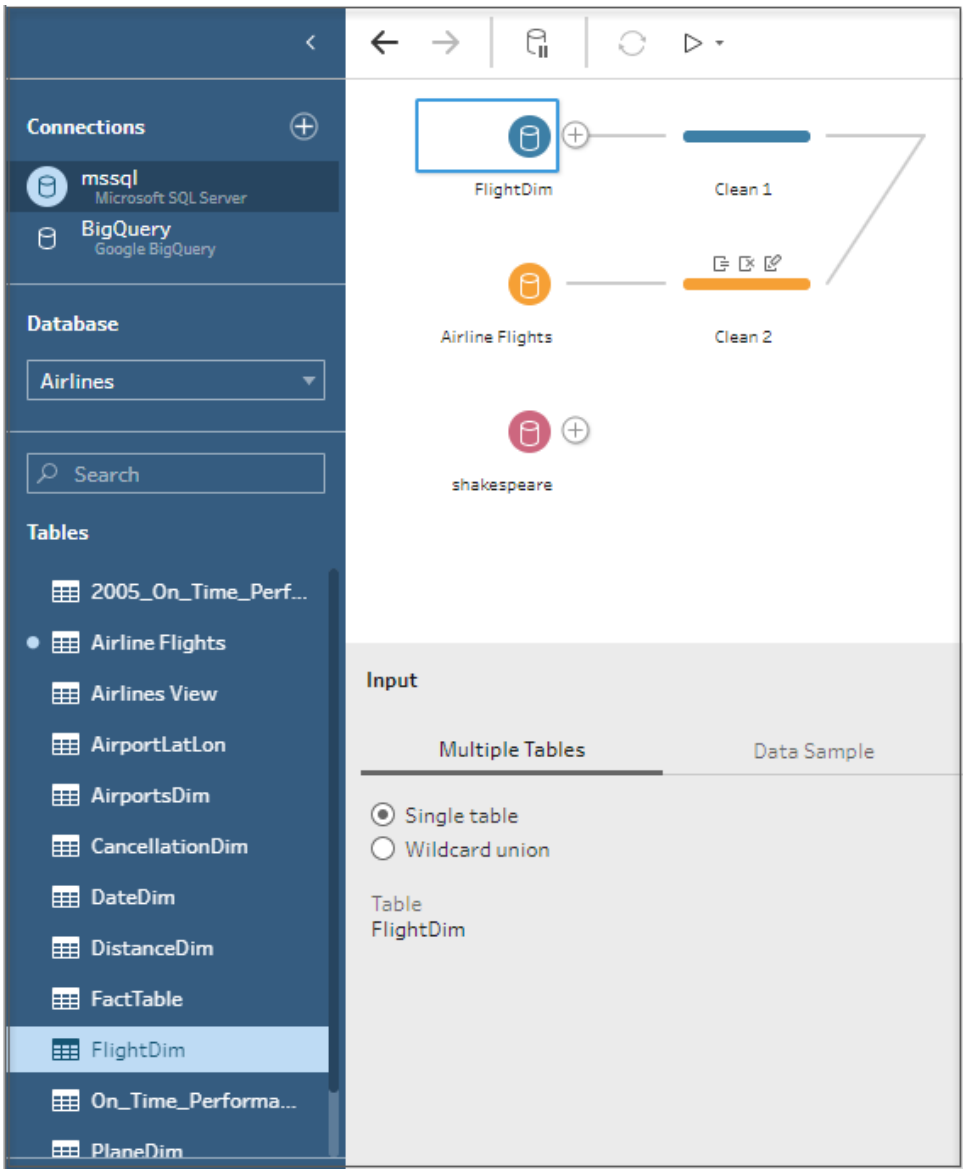
Open a flow in Tableau Prep Builder

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.

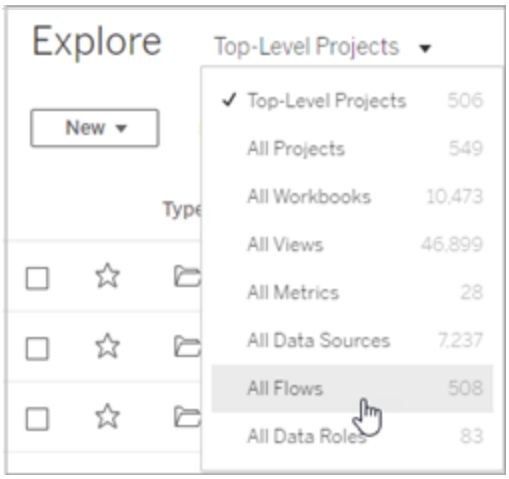
After you connect to your data, use the different options in the Input step to identify the data that you want to work with in your flow. Then you can add a cleaning step or other step type to examine, clean and shape your data.

When your flows include multiple data source connectors, Tableau Prep helps you easily see which connectors and tables are associated with your Input steps. When you click on the Input step, the associated connector and data table is highlighted in the Connections pane. This option was added in Tableau Prep Builder version 2020.1.1 and is also supported when editing flows on the web.

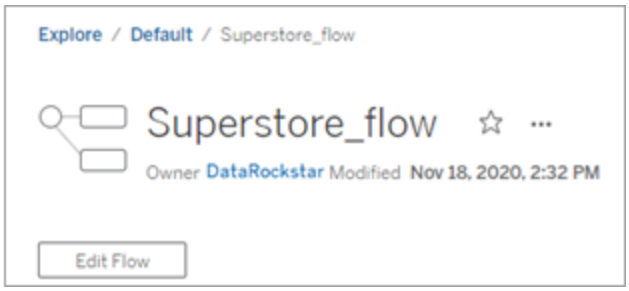


Open a flow in Tableau Prep on the web

1. To open and edit an existing flow, on the **Explore** page select **All Flows** from the top drop-down menu and select your flow from the list.



2. On the **Flow Overview** page, click **Edit** to edit your flow.



Your flow will open in a new tab. As soon as you start making changes, Tableau will automatically save your changes every few seconds and save your modified flow as a draft. Drafts are only visible to you and your administrator.

When you're finished, you can close your flow and continue making changes later or publish your flow to apply your changes, creating a new version of the flow.

Like other tools, flow publishing uses a first-in method. If another users modifies and republishes the flow before you, their changes are committed first. But you can track and revert to a previous version using the **Revision History** page. For more information, see Work with Content Revisions in the Tableau Desktop help.

Connect to Data

Tableau Prep helps you clean and shape your data for analysis. The first step in this process is to identify the data you'll work with.

> **Note**: Starting in version 2020.4.1, you can also create and edit flows in Tableau Server and Tableau Cloud. Information in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see Tableau Prep on the Web in the Tableau Server help.

You can connect to your data using any of the following:

- Built-in connectors for popular data types
- Custom connectors for other data types
- Published data sources
- Tableau data extracts
- Tableau Catalog

Connect via built-in connectors for popular data types

The most common way to connect to data is to use the built-in connectors in Tableau Prep Builder. These are available for most popular data types, and new connectors are added frequently with new versions of Tableau Prep Builder. For a list of available connectors, open

Tableau Prep Builder or start a flow on the web, then click the **Add connection** button to see available connectors listed under **Connect** in the left pane.

Most built-in connectors work the same across all of our platforms and are described in the Supported Connectors topic in the Tableau Desktop Help.

> **Note**: When connecting to a data source, don't use the reserved name ‘Number of Records’ for a field name or a calculated field. Using this reserved name will result in a permission-related error.

Considerations when using built-in connectors

- 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.
- When using a MySQL-based connector, the default behaviour is that the connection is secure when SSL is enabled. However, Tableau Prep Builder does not support custom certificate-based SSL connections for MySQL-based connectors.
- Some connectors, detailed in the sections below, have different requirements when using them with Tableau Prep Builder.

Connect to cloud data sources via Tableau Server or Tableau Cloud

You can connect to cloud data sources in Tableau Prep just like Tableau Desktop, but if you plan to publish flows that connect to cloud data sources and run them on a schedule in your server, you'll need to configure your credentials in Tableau Server or Tableau Cloud.

You set up your credentials in the **Settings** tab in the **My Account Settings** page and connect to your cloud connector input using these same credentials.

Tableau Prep Builder

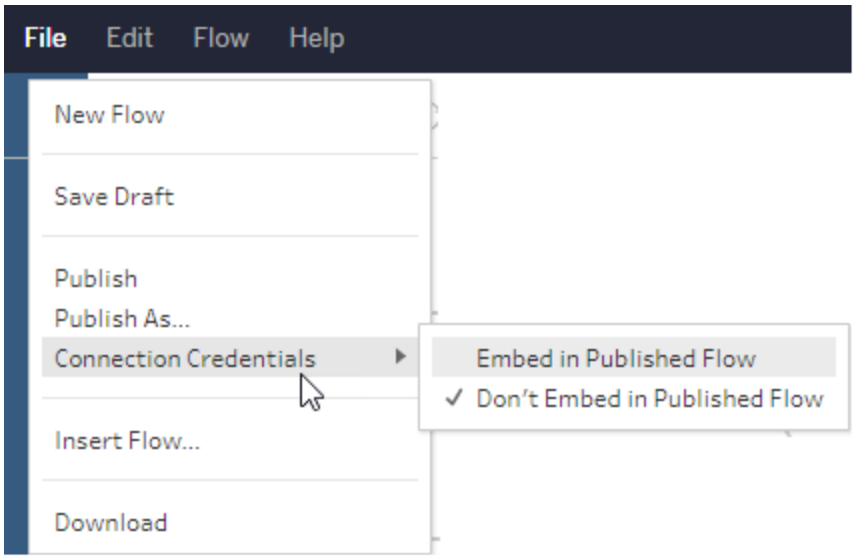
When publishing the flow, on the **Publish** dialog, click **Edit** to edit the connection, then in the **Authentication** drop-down, select **Embed** <your credentials>.

You can also add credentials right from the publish dialog (Tableau Prep Builder version 2020.1.1 and later) when publishing your flow and then automatically embed them in your flow when you publish. For more information, see **Publish a flow from Tableau Prep Builder** on page 467.

If you don't have saved credentials set up, and you select **Prompt user** in the **Authentication** drop-down, after you publish the flow, you must edit the connection and enter your credentials in the **Connections** tab in Tableau Server or Tableau Cloud or the flow will fail when run.

Tableau Prep on the web

In web authoring, you can embed credentials from the top menu under **File** > **Connection Credentials**. For more information, see Publishing flows in the Tableau Server help.



In Tableau Prep Builder version 2019.4.1, the following cloud connectors were added and are also available when creating or editing flows on the web:

- Box
- Dropbox
- Google Drive
- OneDrive

For more information about how to connect to your data using these connectors, see the connector-specific help topic in the Tableau Desktop help.

Connect to Salesforce data

Supported in Tableau Prep Builder version 2020.2.1 and later and when authoring flows on the web starting in Tableau Server and Tableau Cloud version 2020.4.

Tableau Prep Builder supports connecting to data using the Salesforce connector, just like Tableau Desktop, but with a few differences.

- Tableau Prep Builder supports any join type you want to do.
- Custom SQL can be created in Tableau Prep Builder 2022.1.1 or later. Flows that use custom SQL can be run and existing steps can be edited in 2020.2.1 or later.
- Using a standard connection to create your own custom connection isn't currently supported.
- You can't change the default data source name to be something unique or custom.

- If you plan to publish flows on Tableau Server and want to use saved credentials, the server administrator must configure Tableau Server with an OAuth client ID and secret on the connector. For more information, see Change Salesforce.com OAuth to Saved Credentials in the Tableau Server help.
- To run incremental refresh on flow inputs that use the Salesforce connector, you must be using Tableau Prep Builder version 2021.1.2 or later. For more information about using incremental refresh, see**Refresh Flow Data Using Incremental Refresh** on page 416.

Tableau Prep imports the data by creating an extract. Only extracts are currently supported for Salesforce. The initial extract may take some time to load, depending on the amount of data that is included. You will see a timer in the Input step while the data loads.

For general information about using the Salesforce connector, see Salesforce in the Tableau Desktop and Web Authoring help.

Connect to the Salesforce Data Cloud

Tableau Prep supports connecting to data using Salesforce Data Cloud. (Before 14 February 2023, Data Cloud was called Customer Data Platform.) The way you connect depends on your version:

Versions 2023.2 and later

Use the Salesforce Data Cloud connector. Compared to the earlier Customer Data Platform connector, the Salesforce Data Cloud connector is simpler to set up, recognises Data Spaces, presents clearer object labels and is powered by accelerated queries. See the steps below.

1. In the Connections pane select **Salesforce Data Cloud** from the Server connector list.
2. In the **Salesforce Data Cloud** dialog, click **Sign In**.
3. Sign in to Salesforce with your user name and password.
4. Select **Allow**.
5. Close the browser pane.
6. In Tableau Prep, select a Data Space to see tables.
7. Select a table.

Version 2021.4

See Tableau Desktop (version 2023.1 and earlier) in the Salesforce Data Cloud topic, except the required Salesforce_CDP.taco file is automatically installed already.

Versions 2021.1 - 2021.3

See Tableau Desktop (version 2023.1 and earlier) in the Salesforce Data Cloud topic.

Connect to Google BigQuery data

Tableau Prep Builder supports connecting to data using Google BigQuery, just like Tableau Desktop.

You must configure credentials to enable Tableau Prep to communicate with Google BigQuery. If you plan to publish flows to Tableau Server or Tableau Cloud, OAuth connections must also be configured for those applications.

> **Note**: Tableau Prep doesn't currently support using Google BigQuery customisation attributes.

- Set up OAuth for Google – Configuring OAuth connections for Tableau Server.
- OAuth Connections - Configure OAuth connections for Tableau Cloud.

Configure SSL to connect to Google BigQuery (MacOS only)

If you are using Tableau Prep Builder on Mac and you are using a proxy to connect to Big Query, you may need to modify the SSL configuration to connect to Google BigQuery

> **Note**: No extra steps are required for Windows users.

To configure SSL for OAuth connections to Google BigQuery, complete the following steps:

1. Export the SSL certificate for your proxy to a file, for example proxy.cer. You can find your certificate in `Applications > Utilities > Keychain Access >System > Certificates (under Category).`
2. Locate the version of java that you are using to run Tableau Prep Builder. For example: `/Applications/Tableau Prep Builder 2020.4.app/Plugins/jre/lib/security/cacerts`
3. Open the Terminal command prompt and run the following command for your Tableau Prep Builder version:

> **Note**: The keytool command must be run from the directory that contains the version of java that you are using to run Tableau Prep Builder. You may have to change directories before running this command. For example `cd /Users/tableau_user/Desktop/SSL.cer -keystore Tableau Prep Builder 2020.1.1/Plugins/jre/bin`. Then run the keytool command.

```
keytool -import -trustcacerts -file /Users/tableau_
user/Desktop/SSL.cer -keystore Tableau Prep Builder
<version>/Plugins/jre/lib/security/cacerts -storepass changeit
```

Example: `keytool -import -trustcacerts -file /Users/tableau_user/Desktop/SSL.cer -keystore Tableau Prep Builder 2020.4.1/Plugins/jre/lib/security/cacerts -storepass changeit`

If you get a **FileNotFoundException** (Access denied) when running the keytool command, try running the command with elevated permissions. For example: `sudo keytool -import -trustcacerts -file /Users/tableau_user/Desktop/SSL.cer -keystore Tableau Prep Builder 2020.4.1/Plugins/jre/lib/security/cacerts -storepass changeit`.

Set up and manage your Google BigQuery credentials

The credentials that you use to connect to Google BigQuery in your Input step must match the credentials that are set up in the **Settings** tab in the **My Account Settings** page for Google BigQuery in Tableau Server or Tableau Cloud.

If you select different credentials or no credentials in your authentication settings when publishing your flow, the flow will fail with an authentication error until you edit the connection for the flow in Tableau Server or Tableau Cloud.

To edit your credentials, do the following:

1. In Tableau Server or Tableau Cloud, on the **Connections** tab, on the Google BigQuery connection, click **More actions** ··· .
2. Select **Edit Connection**.
3. Select the saved credentials that are set up in the **Settings** tab in the **My Account Settings** page.

Sign In using Service Account (JSON) file

Supported in Tableau Prep Builder version 2021.3.1 and later. Service Account access is not available when authoring flows on the web.

1. Add a Service Account as a saved credential. For more information, see Change Google OAuth to Saved Credentials.
2. Sign in to Google BigQuery using your email or phone number, then select **Next**.
3. In Authentication, select **Sign In using Service Account** (JSON) file.
4. Enter the file path or use the **Browse** button to search for it.
5. Click **Sign In**.
6. Enter your password to continue.
7. Select **Accept** to allow Tableau to access your Google BigQuery data. You will be prompted to close the browser.

Sign In using OAuth

Supported in Tableau Prep Builder version 2020.2.1 and later and when authoring flows on the web starting in Tableau Server and Tableau Cloud version 2020.4.

1. Sign in to Google BigQuery using your email or phone number, and then select **Next**.
2. In Authentication, select **Sign In using OAuth**.
3. Click **Sign In**.
4. Enter your password to continue.
5. Select **Accept** to allow Tableau to access your Google BigQuery data. You will be prompted to close the browser.

For more information about setting and managing your credentials, see the following topics:

Manage Your Account Settings in the Tableau Desktop and Web Authoring help.

Publish a flow from Tableau Prep Builder on page 467 for information about setting authentication options when publishing a flow.

View and resolve errors for information about resolving connection errors in Tableau Server or Tableau Cloud.

Connect to SAP HANA data

Supported in Tableau Prep Builder version 2019.2.1 and later and when authoring flows on the web starting in Tableau Server and Tableau Cloud version 2020.4.

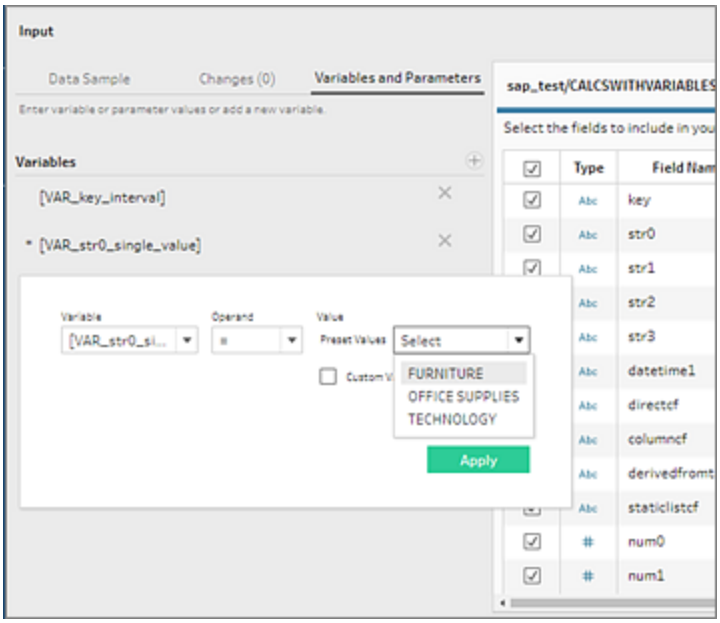
Tableau Prep Builder supports connecting to data using SAP HANA just like Tableau Desktop but with a few differences.

Connect to the database using the same procedure you would use in Tableau Desktop. For more information see SAP HANA. After you connect and search for your table, drag the table to the canvas to begin building your flow.

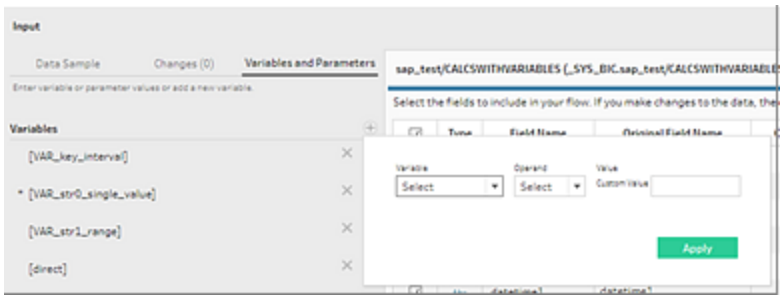
Prompting for variables and parameters when opening a flow isn't supported in Tableau Prep. Instead, in the Input pane, click the **Variables and Parameters** tab and select the variables and operands you want to use, then select from a list of preset values or enter custom values to query your database and return the values you need.

Tableau Prep Builder and Tableau Prep Conductor don't support variables and parameters when connecting to a SAP HANA published data source.

> **Note**: Starting in Tableau Prep Builder version 2019.2.2 and on the web starting in version 2020.4.1, you can use Initial SQL to query your connection. If you have multiple values for a variable, you can select the value you need from a drop-down list.



You can also add additional variables. Click the plus button ⊕ in the **Variables** section and select a variable and operand, then enter a custom value.



Note: This connector requires Tableau Server version 2019.2 and later to run the flow on a schedule. If you are using an earlier server version, you can refresh the flow data using the command line interface. For more information about running flows from the command line see **Refresh flow output files from the command line** on page 425. For more information about version compatibility, see **Version Compatibility with Tableau Prep** on page 444.

Connect to Spatial files and databases

Supported in Tableau Prep Builder version 2020.4.1 and later and when authoring flows on the web starting in Tableau Server and Tableau Cloud version 2020.4.

You can connect to spatial files and spatial data sources in Tableau Prep Builder or when creating or editing flows on the web.

Tableau Prep supports the following connection types:

- Spatial File formats
  - Tableau Prep Builder: Esri Shapefiles, Esri File Geodatabases, KML, TopoJSON, GeoJSON, extracts, MapInfo MID/MIF, TAB files and zipped shapefiles.
  - Tableau Server and Tableau Cloud: Zipped shapefiles, KML, TopoJSON, GeoJSON, Esri File Geodatabases and extracts.
- Spatial databases (Amazon Redshift, Microsoft SQL Server, Oracle and PostgreSQL).

You can also combine spatial tables with non-spatial tables using a standard join and output spatial data to an extract (.hyper) file. Spatial functions, spatial joins through intersections and visualising spatial data on a map view in Tableau Prep is not currently supported.

Supported cleaning operations

When working with shape file data, some cleaning operations are not supported. Only the following cleaning operations are available in Tableau Prep when working with shape file data.

- Filters: Only to remove Null or unknown values
- Rename Field
- Duplicate Field
- Keep Only Field
- Remove Field
- Create Calculated Field

Before you connect

Before connecting to spatial files, makes sure the following files are in the same directory:

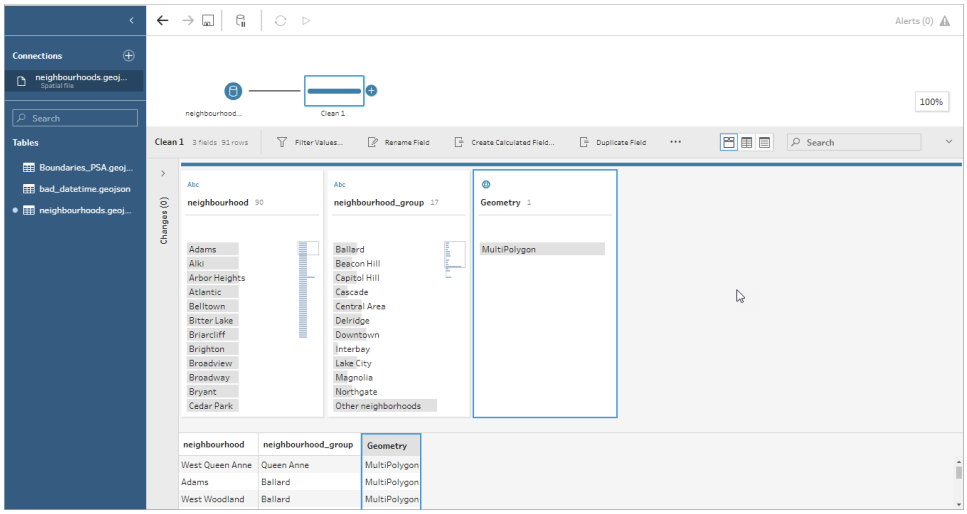
- **Esri shapefiles**: The folder must contain .shp, .shx, .dbf and .prj files as well as .zip files of the Esri shapefile.
- **Esri File Geodatabases**: The folder must contain the File Geodatabase's .gdb or the .zip of the File Geodatabases's .gdb.
- **KML files**: The folder must contain the .kml file. (No other files are required.)
- **GeoJSON files**: The folder must contain the .geojson file.(No other files are required.)
- **TopoJSON files**: The folder must contain the .json or .topojson file. (No other files are required.)

Connect to spatial files

1. Do one of the following:
   - Open Tableau Prep Builder and click the **Add connection** button.
   - Open Tableau Server or Tableau Cloud. From the **Explore** menu, click **New** > **Flow**.
2. From the list of connectors, select **Spatial file**.

   Spatial fields are assigned spatial data type and cannot be changed. If the fields come from a spatial file, the field is assigned a default field name of "Geometry". If the fields come from a spatial database, the database field names are shown. If Tableau can't

determine the type of data, the field shows as "Null".



Connect using ODBC

Supported in Tableau Prep Builder version 2019.2.2 and later. This connector type is not yet supported when authoring flows on the web.

If you need to connect to data sources that aren't listed in the **Connections** pane, you can connect to any data source using the **Other Databases (ODBC)** connector that supports the SQL standard and implements the ODBC API. Connecting to data using the **Other Databases (ODBC)** connector works similarly to how you might use it in Tableau Desktop, however there are a few differences:

- You can only connect using the DSN (data source name) option.
- To publish and run your flow in Tableau Server, the server must be configured using a matching DSN.

  > **Note**: Running flows from the command line that include the Other Databases (ODBC) connector isn't currently supported.

- There is a single connection experience for both Windows and MacOS. Prompting for connection attributes for ODBC drivers (Windows) isn't supported.
- Only 64-bit drivers are supported by Tableau Prep Builder.

Before you connect

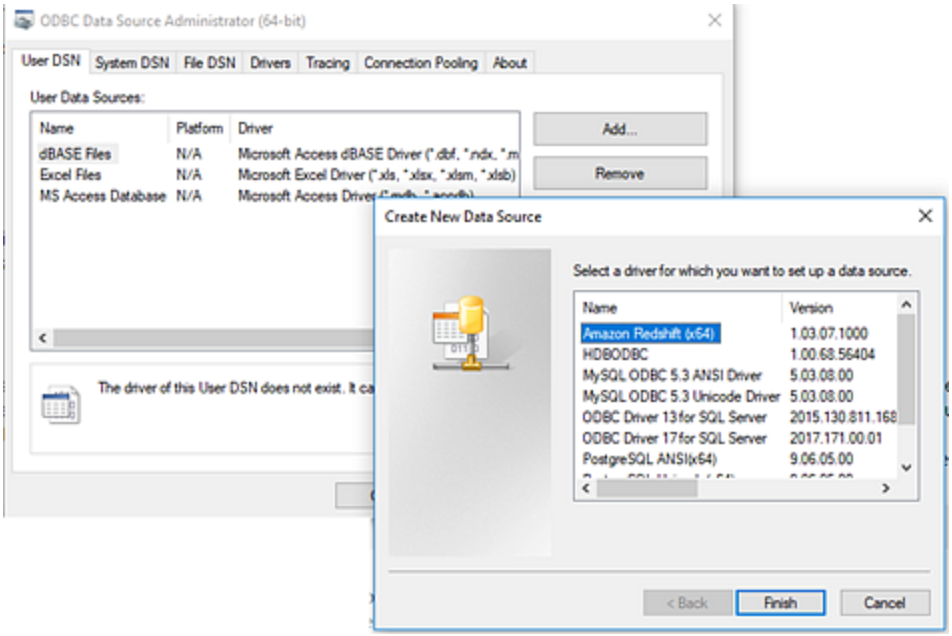
To connect to your data using the **Other Databases (ODBC)** connector, you must install the database driver and set up and configure your DSN (data source name). To publish and run flows to Tableau Server, the server must also be configured with a matching DSN.

Important: Tableau Prep Builder only supports 64-bit drivers. If you have a 32-bit driver already set up and configured, you may need to uninstall it and then install the 64-bit version if the driver doesn't allow both versions to be installed at the same time.

1. Create a DSN using either the **ODBC Data Source Administrator (64-bit)** (Windows) or the using an ODBC Manager utility (MacOS).

   If you don't have the utility installed on your Mac, you can download one from (www.odbcmanager.net for example) or you can manually edit the odbc.ini file.

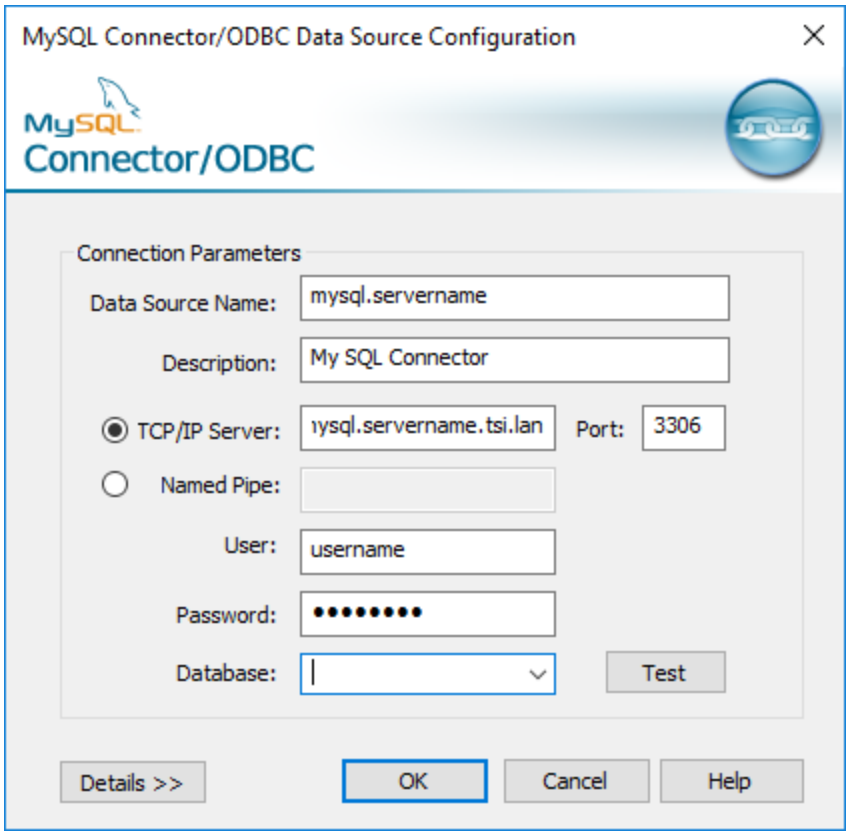
2. In the **ODBC Data Source Administrator (64-bit)** (Windows) or the ODBC Manager utility (MacOS), add a new data source then select the driver for the data source then click **Finish**.



3. In the **ODBC Driver Setup** dialog, enter the configuration information such as server name, port, user name and password. Click **Test** (if your dialog has that option) to verify that your connection is set up correctly, then save your configuration.

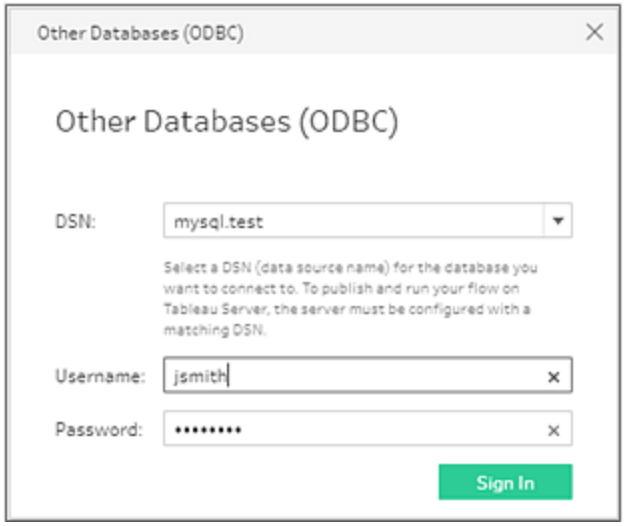
> **Note**: Tableau Prep Builder doesn't support prompting for connection attributes so you must set this information when configuring the DNS.

This example shows the configuration dialog for a MySQL Connector.

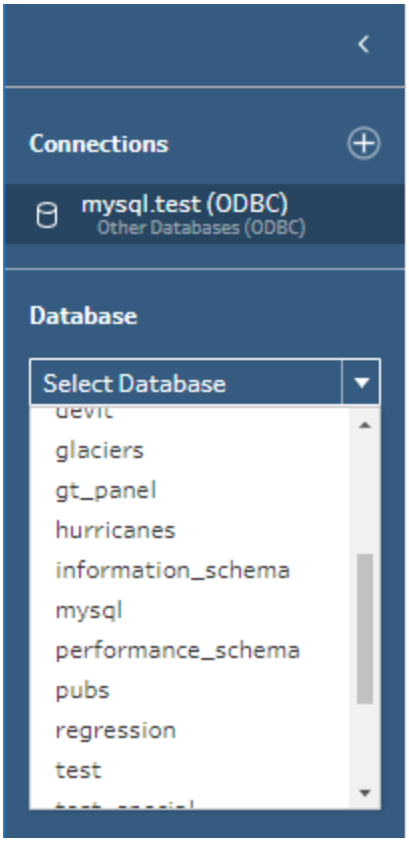


Connect using Other Databases (ODBC)

1. Open Tableau Prep Builder and click the **Add connection** button.
2. From the list of connectors, select **Other Databases (ODBC)**.
3. In the **Other Databases (ODBC)** dialog, select a DSN from the drop-down list and enter the user name and password. Then click **Sign In**.



4. From the **Connections** pane, select your database from the drop-down list.



Connect to Microsoft Excel data and clean with Data Interpreter

Supported for direct Microsoft Excel connections only. Data Interpreter isn't currently available for Excel files stored in cloud drives.

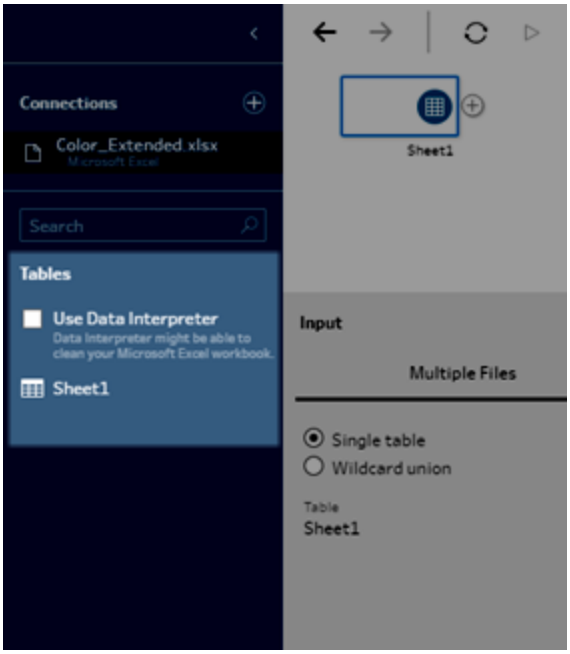
When working with Microsoft Excel files, you can use Data Interpreter to detect sub-tables in your data as well as remove extraneous information to help prepare your data for analysis. When you turn on Data Interpreter, it detects these sub-tables and lists them as new tables in the **Tables** section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

If you 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.

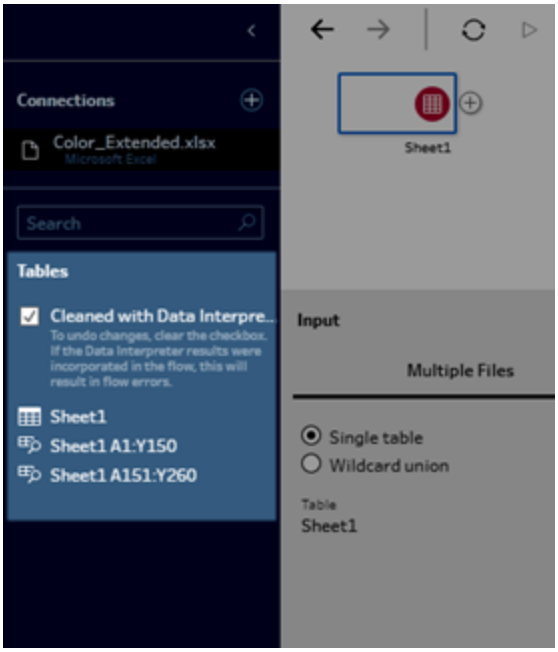
> **Note**: Currently, Data Interpreter only detects sub-tables in your Excel spreadsheets and doesn't support specifying the starting row for text files and spreadsheets. Also, tables that Data Interpreter detected are not included in the Wildcard Union search results.

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. Tick the **Use Data Interpreter** box.
4. Drag the new table to the **Flow** pane to include it in your flow. To remove the old table, right-click on the Input step for the old table and select **Remove**.

Connect using custom connectors

When Tableau Prep doesn't provide a built-in connector for your ODBC- and JDBC-based data, you can use a custom connector. You can:

- Use a partner-built connector. For more information about connectors in the exchange, see **Use partner-built connectors** below.
- Use a custom connector built with the Tableau Connector SDK. The Connector SDK provides tools to build a customised connector for ODBC- or JDBC-based data. For more information, see Connectors Built with the Tableau Connector SDK in Tableau Desktop help.

Custom connectors for ODBC- and JDBC-based data are supported in Tableau Prep Builder version 2020.4.1 and later.

For connector developers, see Run Your Connector for more information.

Some custom connectors require the installation of an additional driver. If prompted during the connection process, follow the prompts to download and install the required driver. Custom connector currently cannot be used with Tableau Cloud.

Use partner-built connectors

Partner-built or other custom connectors are available from the Connect pane. These connectors are listed under Additional Connections and are also available from the Tableau Exchange connectors page.

1. Click **Connections** in the left pane.
2. From the **Additional Connectors** section in the Connect pane, click on the connector you want to use.
3. Click **Install and Restart Tableau**.

   After the connector is installed, it appears in the **To a Server** section of the Connect pane.

Note: If you receive a warning that the connectors can't load, install the .taco file you need from the Tableau Exchange connectors page. If you are prompted to install the drivers, go to Tableau Exchange for driver download instructions and locations.

Connect to published data sources

Published data sources are those you can share with others. When you want to make a data source available to other users, you can publish it from Tableau Prep Builder (version 2019.3.1 and later) to Tableau Server or Tableau Cloud, or as output from your flow.

You can use a published data source as an input data source for your flow, whether you are working in Tableau Prep Builder or on the web.

Note: When you publish a flow that includes a published data source as an input, the publisher is assigned as the default flow owner. When the flow runs, it uses the flow owner for the **Run As** account. For more information about the Run As account, see Run As Service Account. Only the Site or Server Administrator can change the flow owner in Tableau Server or Tableau Cloud and only to themselves.

Tableau Prep Builder supports:

- Published data sources with user filters or functions starting in Tableau Prep Builder version 2021.1.3.
- Connections to a single server and site. Logging in to a different server or the same server and different site isn't supported. You must use the same server or site connection to do the following:
  - Connect to the published data source.
  - Publish flow output to Tableau Server or Tableau Cloud.
  - Schedule the flow to run on Tableau Server or Tableau Cloud.

  If your flow uses published data sources and you sign out of the server, this breaks the flow connection. The flow will be in an error state and you won't be able to see the data from the published data source in the profile pane or data grid.

Note: Tableau Prep Builder does not support published data sources that include multi-dimensional (cube) data, multi-server connections or published data sources with related tables.

Tableau Server and Tableau Cloud supports:

- Published data sources with user filters or functions starting in Tableau Server and Tableau Cloud version 2021.2.
- Creating or editing a flow on the web using a published data source (Tableau Server or Tableau Cloud version 2020.4 and later)
- Connecting to published data sources (Tableau Server and Tableau Cloud version 2019.3 and later)

  Note: Earlier versions of Tableau Server may not support all features of the published data source.

About credentials and permissions:

- You must be assigned a role of Explorer or higher in the server site where you are signed in to connect to published data sources. Only Creators can create or edit flows on the web. For more information about site roles, see Set User's Site Roles in the Tableau Server help.
- In Tableau Prep Builder, data source access is authorised based on the identity of the user signed into the server. You will see only the data to which you have access.
- In Prep web authoring (Tableau Server and Tableau Cloud), data source access is also authorised based on the identity of the user signed in to the server. You will see only the data to which you have access.

  However, when you run the flow manually or using a schedule, data source access is authorised based on the identity of the flow owner. The last user to publish a flow becomes the new flow owner.
- Site and Server Administrators can change the flow owner, but only to themselves.
- Credentials must be embedded to connect to the published data source.

  Tip: If credentials aren't embedded for the data source, update the data source to include the embedded credentials.

Using published data sources in your flow

To connect to a published data source and use it in your flow, follow the instructions for your Tableau Prep version:

Tableau Prep Builder version 2020.2.2 and later and on the web

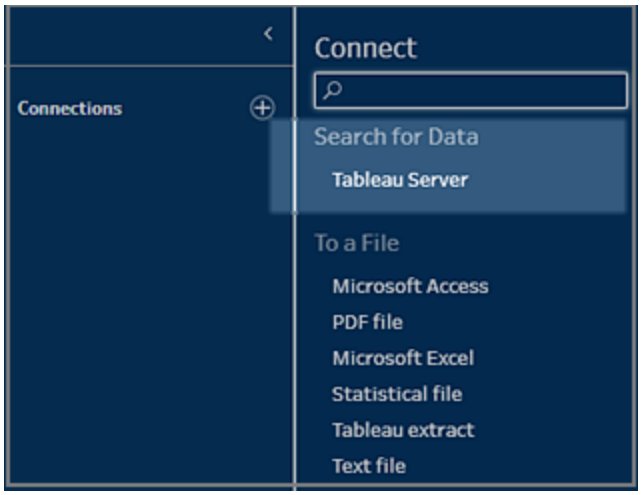
You can connect to published data sources and more that are stored on Tableau Server or Tableau Cloud directly from the Connect pane. If you have the Data Management with Tableau Catalog enabled, you can also search for and connect to databases and tables and view or filter by meta data about the data sources, such as descriptions, data quality warnings and certifications.

For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Cloud Help.

1. Open Tableau Prep Builder and click the **Add connection** button.

   In web authoring, from the **Home** page, click **Create** > **Flow** or from the **Explore** page, click **New** > **Flow**. Then click **Connect to Data**.

2. On the **Connect** pane, under **Search for Data**, select **Tableau Server**.



3. Sign in to connect to your server or site.

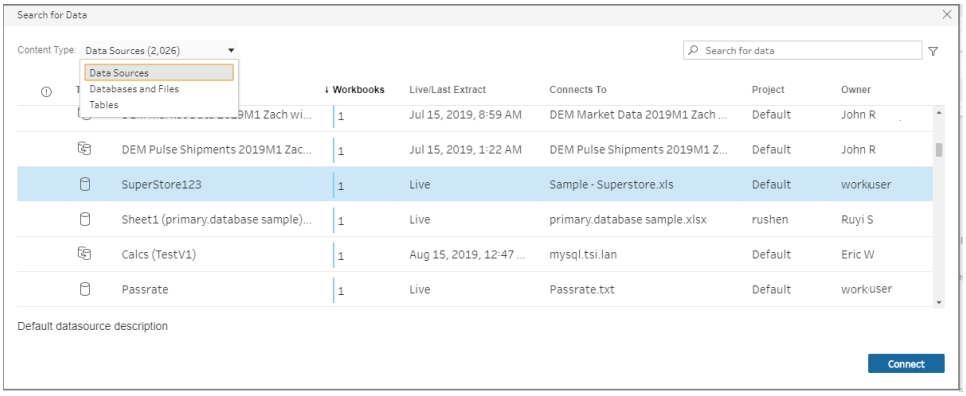
   In web authoring, the **Search for data** dialog opens for the sever you are signed into.

4. In the **Search for Data** dialog, search from a list of available published data sources. Use the filter option to filter by connection type and certified data sources.

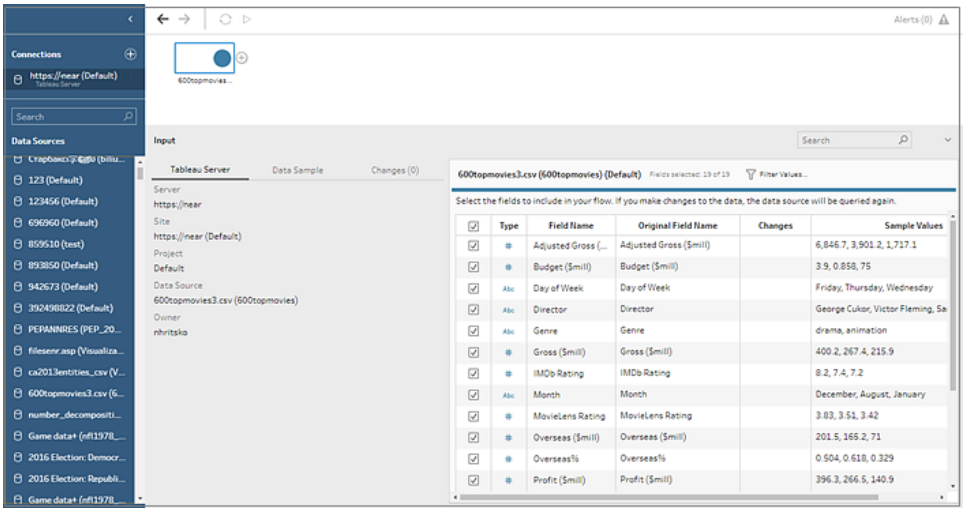
5. Select the data source you want to use, then click **Connect**.

   If you don't have permission to connect to a data source, the row and the **Connect** button is greyed out.

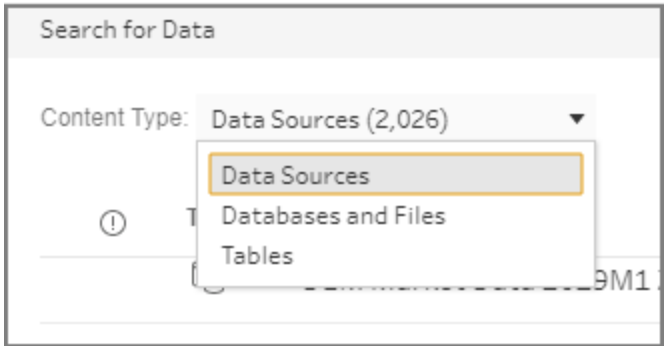
> **Note**: The **Content Type** drop-down isn't shown if you don't have Data Management with Tableau Catalog enabled. Only published data sources are shown in the list.



6. The data source is added to the Flow pane. In the **Connections** pane, you can select additional data sources or use the search option to find your data source and drag it to the flow pane to build your flow. The **Tableau Server** tab in the Input pane shows details about the published data source.



7. (Optional) If you have Data Management with Tableau Catalog enabled, use the **Content Type** drop-down to search for databases and tables.



You can use the filter option in the top right corner to filter your results by connection type, data quality warnings and certifications.

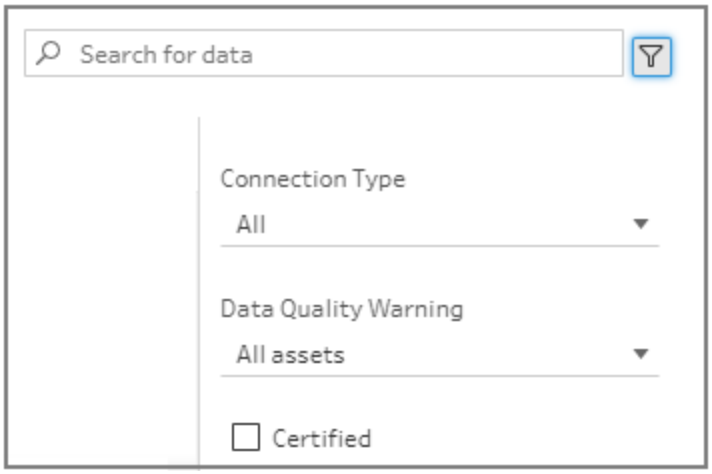
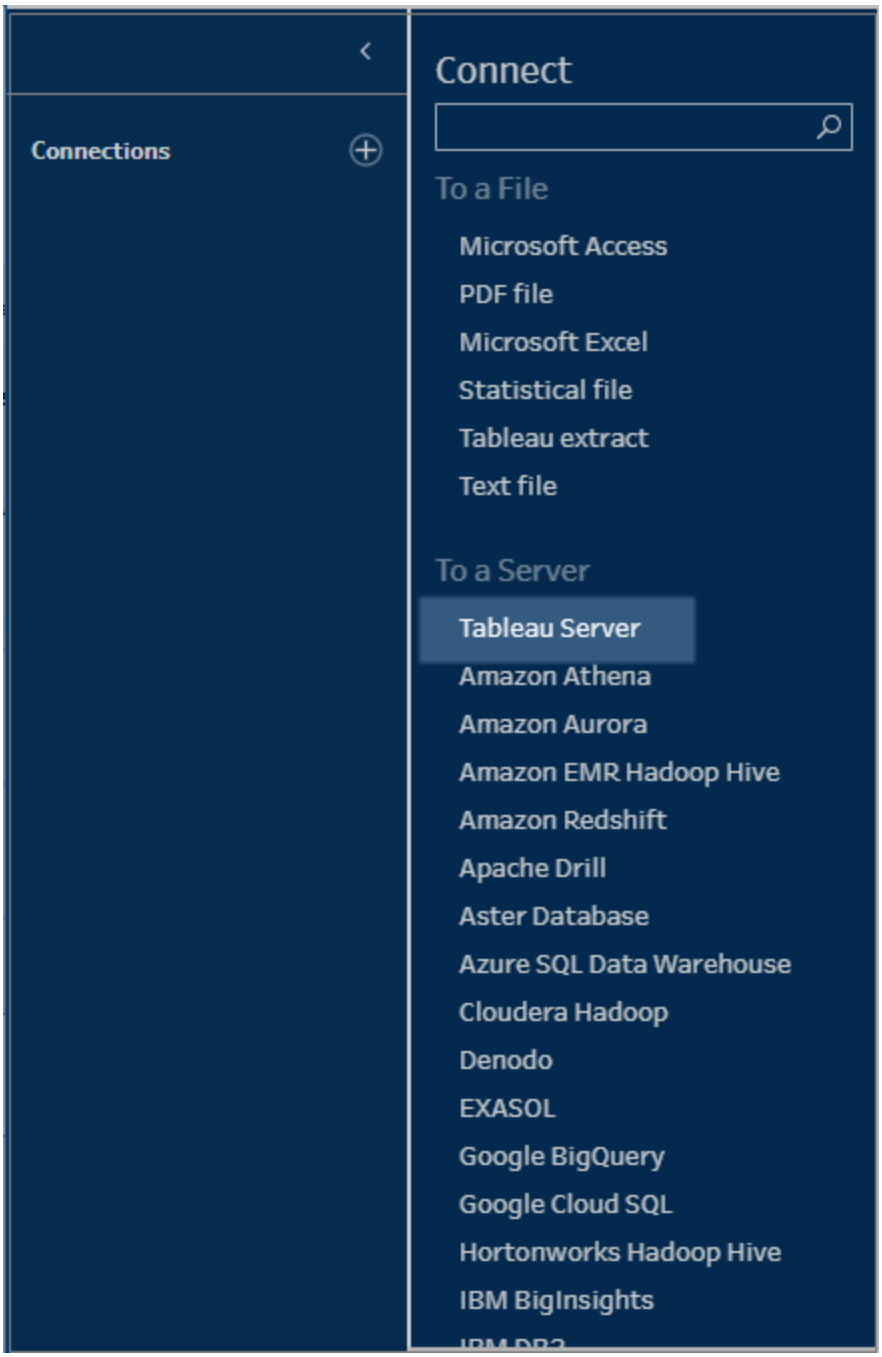
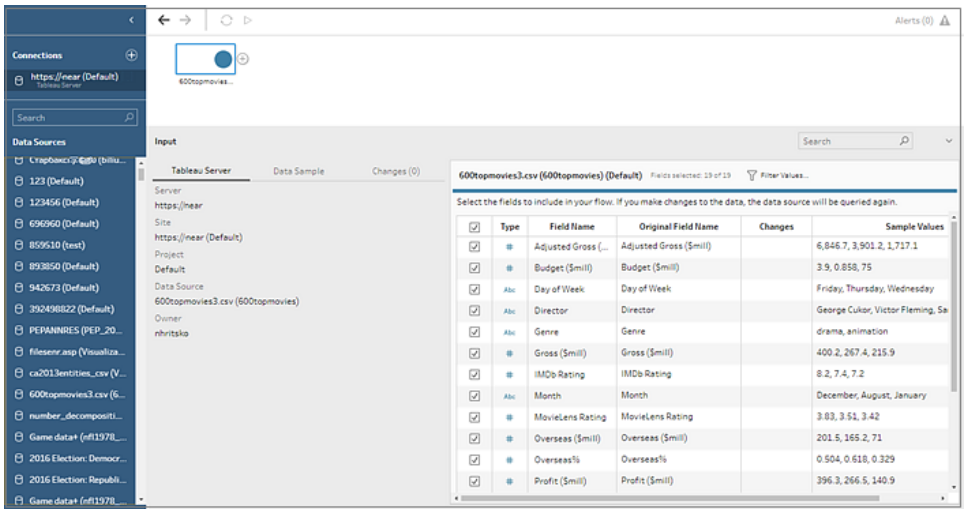


Tableau Prep Builder version 2020.2.1 and earlier

1. Open Tableau Prep Builder and click the **Add connection** button.
2. From the list of connectors, select **Tableau Server**.



3. Sign in to connect to your server or site.
4. Select your data source or use the search option to find your data source and drag it to the flow pane to start your flow. The **Tableau Server** tab in the Input pane shows details about the published data source.



Connect to Virtual Connections

Supported in Tableau Prep Builder version 2021.4.1 and later and in Tableau Server and Tableau Cloud version 2021.4 and later. Data Management is required to use this feature.

You can connect to data using virtual connections for your flows. Virtual connections are a sharable resource that provides a central access point to data.

Considerations when connecting to virtual connections:

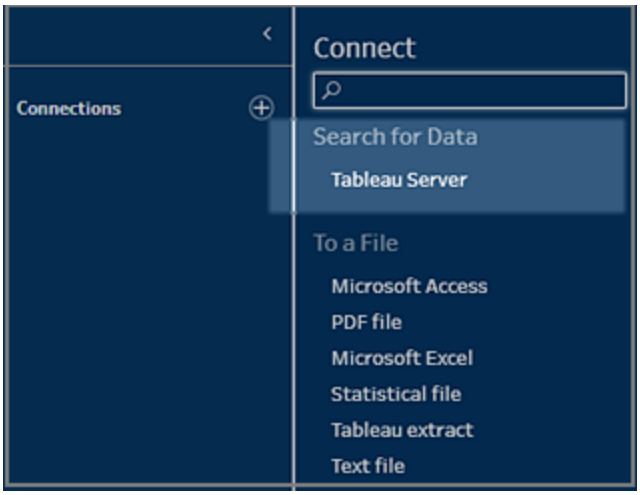
- Database credentials are embedded in the virtual connection. You only need to sign into your server to access the tables in the virtual connection.
- Data policies that apply row-level security can be included in the virtual connection. Only tables, fields and values you have access to are shown when working with and running your flows.
- Row-level security in virtual connections does not apply to flow output. All users with access to the flow output see the same data.
- Custom SQL and Initial SQL are not supported.
- Parameters are not supported. For more information about using parameters in your flow, see **Create and Use Parameters in Flows** on page 205.

For more information about virtual connections and data policies, see the Tableau Server or Tableau Cloud help.

1. Open Tableau Prep Builder and click the **Add connection** button.

   In web authoring, from the **Home** page, click **Create** > **Flow** or from the **Explore** page, click **New** > **Flow**. Then click **Connect to Data**.

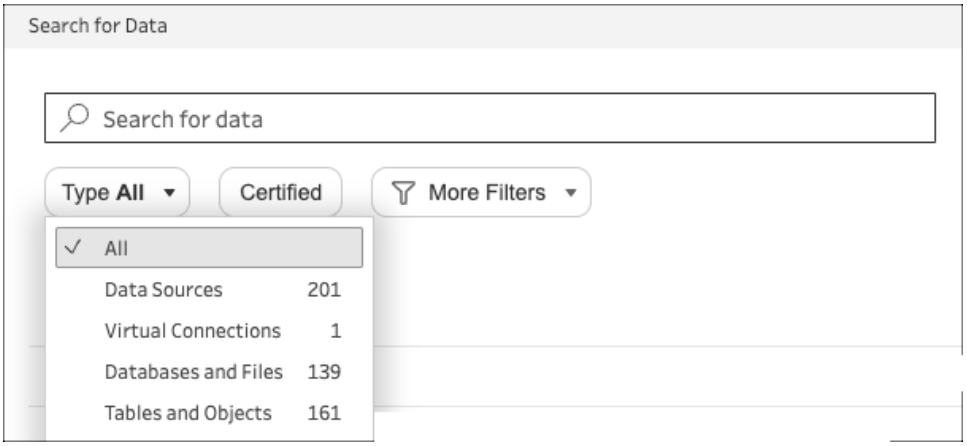
2. On the **Connect** pane, under **Search for Data**, select **Tableau Server**.



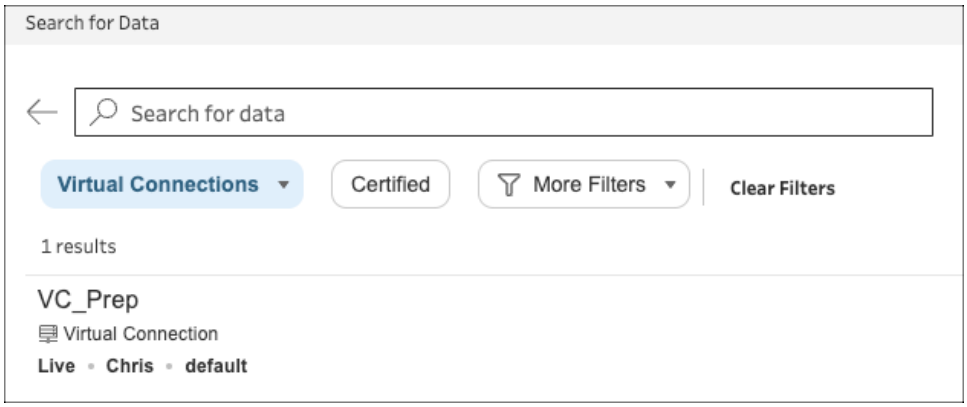
3. Sign in to connect to your server or site.

   In web authoring, the **Search for data** dialog opens for the sever you are signed into.

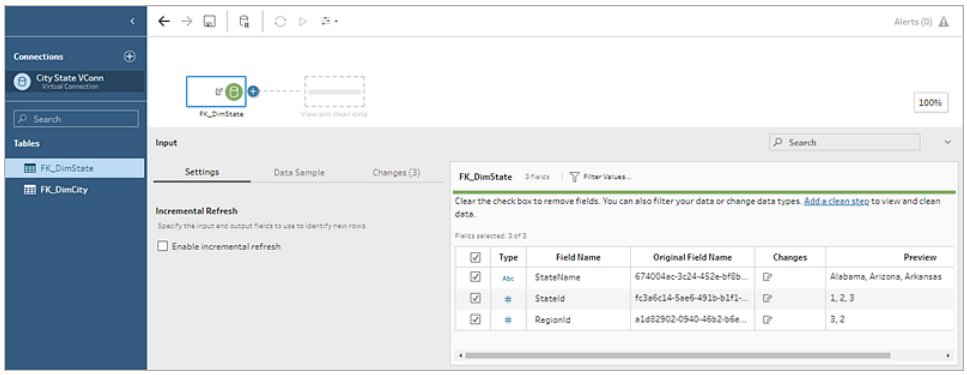
4. Click **Type All**.



5. Select **Virtual Connections**.



6. Select the data source you want to use, then click **Connect**.
7. The data source is added to the Flow pane. In the **Connections** pane, you can select from the list of tables included in the virtual connection and drag them to the flow pane to begin your flow.



Note: If you see **Rename** operations in the **Changes** pane when connecting to a virtual connection, do not remove them. Tableau Prep auto-generates these operations to map to and display the field's user-friendly name.

Connect to Tableau data extracts

You can connect to a data extract as input to your data flow. Extracts are saved subsets of data that you can create by using filters and configuring other limits. Extracts are saved as .hyper files.

For more information on using extracts with Tableau Prep Builder, see **Save and Share Your Work** on page 383.

Connect to data via Tableau Catalog

If you have the Data Management with Tableau Catalog enabled, you can also search for and connect to databases, files and tables stored on Tableau Server or Tableau Cloud.

For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Cloud Help.

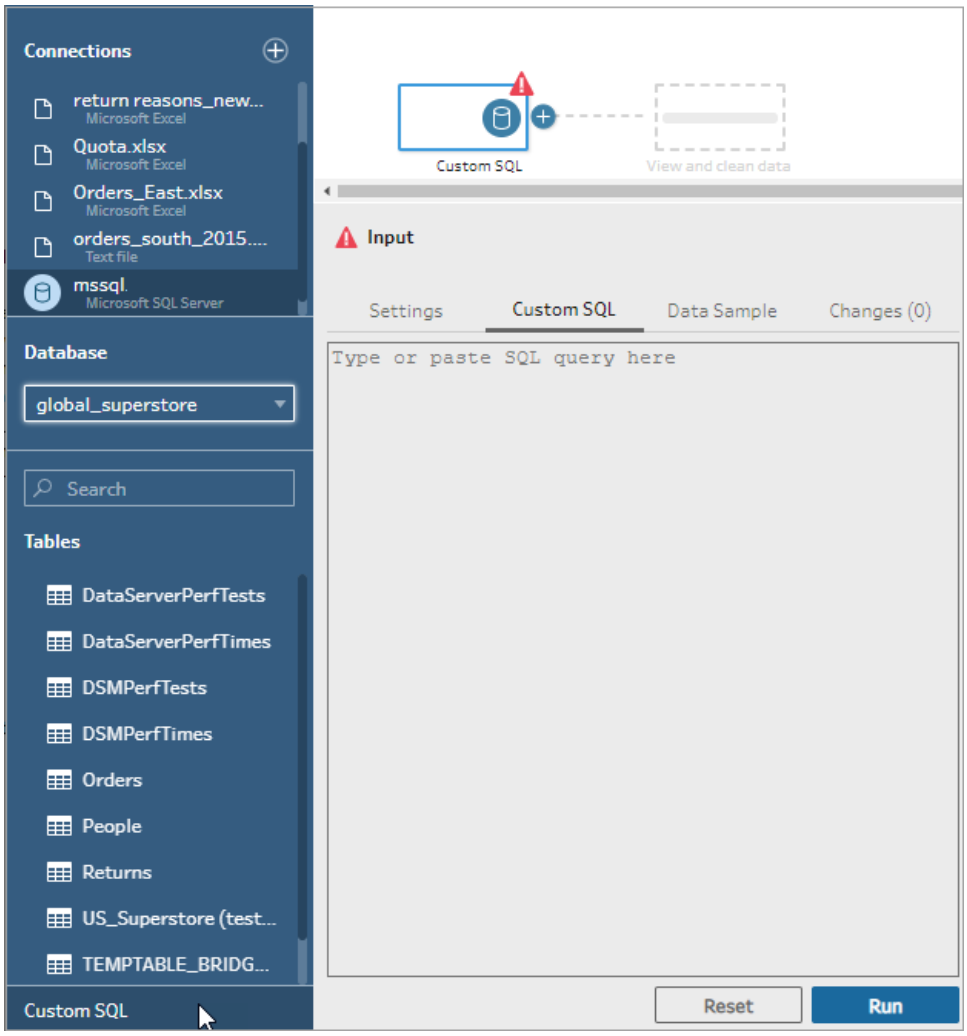
Other connection options

When you connect, you may also see the following options, depending on which connection you choose.

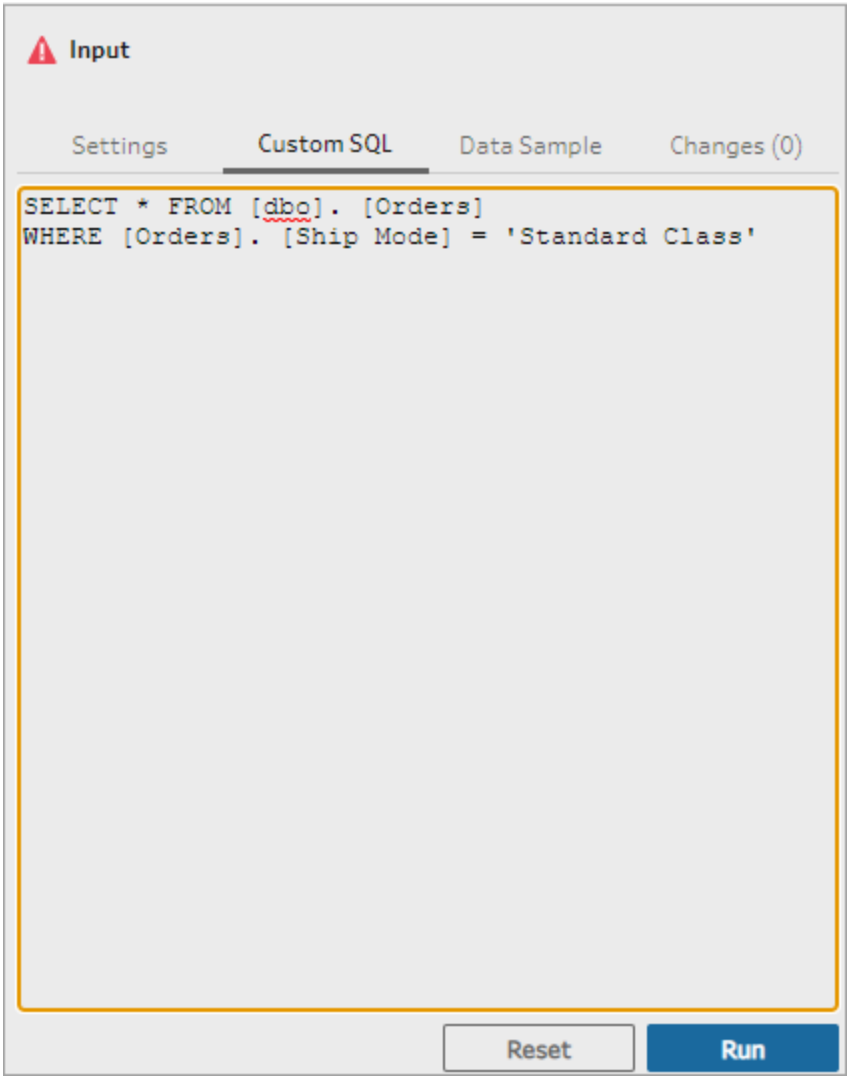
Use Custom SQL to connect to data

If you know exactly the information you need from a database and understand how to write SQL queries, you can use custom SQL queries to connect to data just like you can in Tableau Desktop. You can use custom SQL to union your data across tables, recast fields to perform cross-database joins, restructure or reduce the size of your data for analysis, and so on.

1. Connect to your data source, and in the Connections pane, in the Database field, select a database.
2. Click the **Custom SQL** link to open the Custom SQL tab.



3. Type or paste the query into the text box and then click Run to run your query.



4. Add a clean step in the flow pane to see that only relevant fields from the custom SQL query are added to your flow.

Use Initial SQL to query your connections

Supported in Tableau Prep Builder version 2019.2.2 and later and when authoring flows on the web starting in version 2020.4.1

You can specify an Initial SQL command that will run when a connection is made to a database that supports it. For example, when connecting to Amazon Redshift, you can enter a SQL statement to apply a filter when connecting to the database just like adding filters in the Input step. The SQL command will apply before data is sampled and loaded into Tableau Prep.

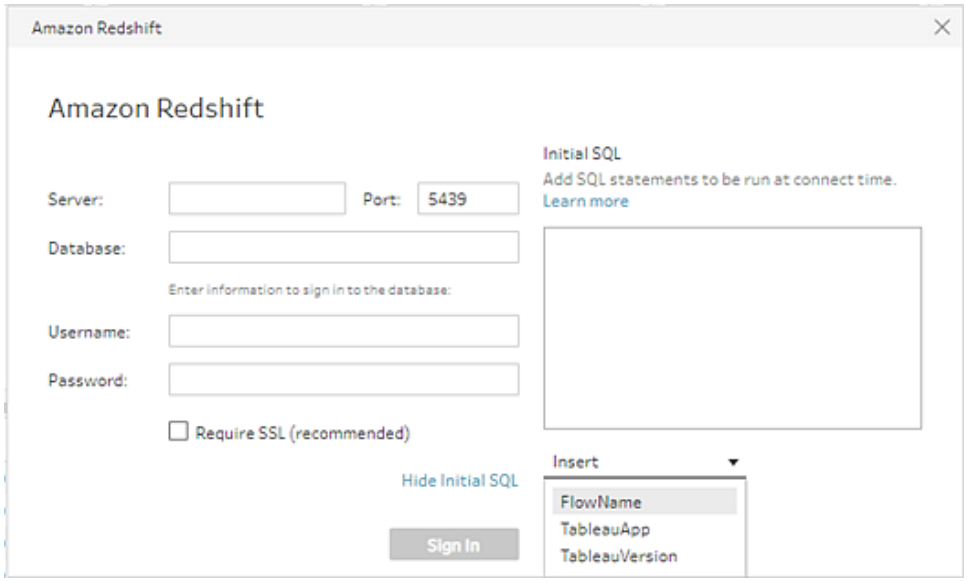
Starting in Tableau Prep Builder (version 2020.1.3) and on the web, you can also include parameters to pass application name, version and flow name data to include tracking data when you query your data source.

Run Initial SQL

To refresh your data and run the Initial SQL command do one of the following:

- Change the Initial SQL command and refresh the Input step by re-establishing the connection.
- Run the flow. The Initial SQL command is run before processing all of the data.
- Run the flow on Tableau Server or Tableau Cloud. The Initial SQL is run every time that the flow is run as part of the data loading experience

Note: Data Management is required to run your flow on a schedule on Tableau Server or Tableau Cloud. For more information about the Data Management, see About Data Management.



1. In the Connections pane, select a connector in the list that supports Initial SQL.
2. Click the **Show Initial SQL** link to expand the dialog and enter your SQL statements.

Include parameters in your Initial SQL statement

Supported in Tableau Prep Builder version 2020.1.3 and later and when authoring flows on the web starting in version 2020.4.1.

You can pass the following parameters to your data source to add additional detail about your Tableau Prep application, version and flow name. The **TableauServerUser** and **TableauServerUserFull** parameters are not currently supported.

Parameter	Description	Returned value
TableauApp	The application being used to access your data source.	Prep Builder Prep Conductor
TableauVersion	The application version number.	Tableau Prep Builder: Returns the exact version. For example 2020.4.1 Tableau Prep Conductor: Returns the major server version where Tableau Prep Conductor is enabled. For example 2020.4
FlowName	The name of the .tfl file in Tableau Prep Builder	Example: Entertainment Data_Cleaned

Configure your Data Set

Note: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

To determine how much of your data set to work with 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.

The Input step is where you can decide what and how much data to include in your flow. This is always the first step in the flow.

If you're connected to an Excel or text file, you can also refresh the data from the Input step. For more information, see **Add More Data in the Input Step** on page 126.

In the Input step, you can:

- Right-click or Cmd-click (MacOS) on the Input step in the flow pane to rename or remove it.
- Union multiple files in the same parent or child directory. For more information, see **Union files and database tables in the Input step** on page 130.
- (version 2023.1 and later) Include automatically generated row numbers based on the original sort order of your data set. See **Include row numbers from your data set** below.
- Search for fields.
- See a preview of field values.
- Configure the field properties by changing the field name or configure the text settings for text files.

  > **Note**: Field values that include square brackets are automatically converted to parentheses.

- Configure the data sample ingested into your flow. See **Set your data sample size** on page 121.
- Remove fields you don't need. You can always go back to the input step and include them later.
- Hide fields that you don't need to clean, but still want to include in your flow output. You can unhide them at any time if you need them.
- Apply filters to selected fields.
- Change the field data type for the data connections that support it.
- (version 2023.3 and later) You can set the header and start row for CSV files.
- (version 2024.1 and later) You can set the header and start row for Excel files.

Include row numbers from your data set

Supported in Tableau Prep Builder version 2023.1 and later and on the web for Microsoft Excel and text (.csv) files.

> **Note**: This option isn’t currently supported for files included in an input union.

Starting in version 2023.1, Tableau Prep automatically generates row numbers based on the original sort order of your data that you can include as a new field in your flow. This is available for Microsoft Excel or Text (.csv) file types only.

In previous releases, if you wanted to include these row numbers, you had to manually add them to the source before adding the data set to your flow.

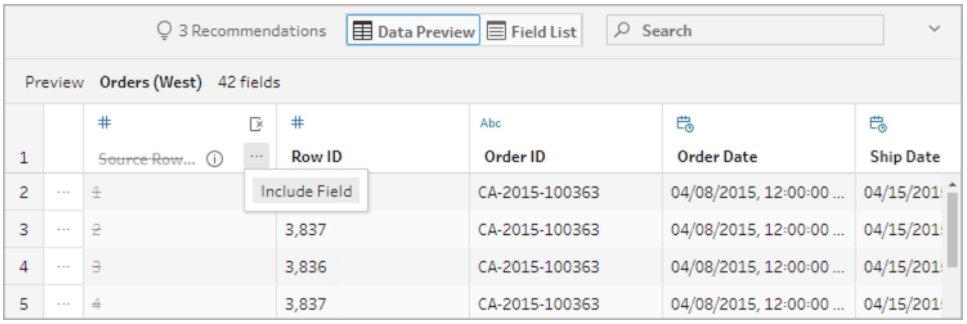
This field is generated in the Input step when you connect to your data. By default, it's excluded from the flow, but you can include it in one click. If you choose to include it, it behaves like any other field and can be used in your flow operations and calculated fields.

Tableau Prep also supports the ROW_NUMBER function for calculated fields. This function is useful when there are fields in your data set that can define the sort, such as Row ID or Timestamp. For more information about using this function, see **Create Level of Detail, Rank and Tile Calculations** on page 275.

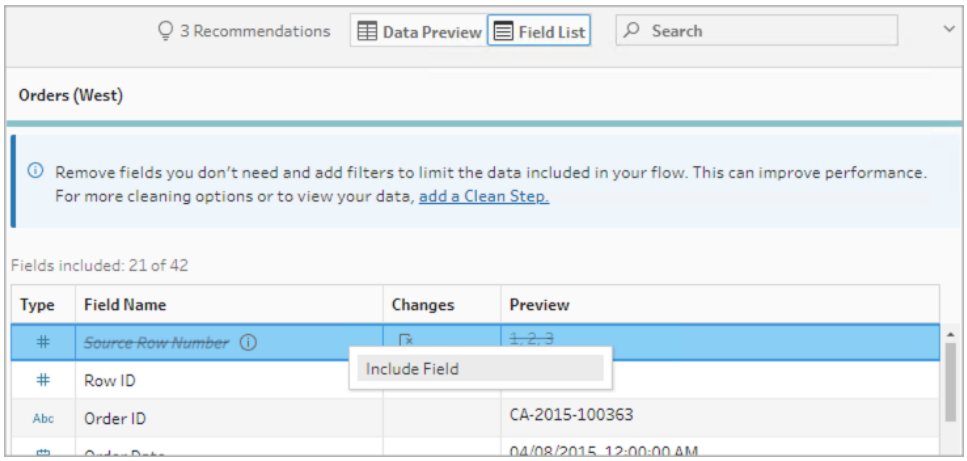
Add the Source Row Number field to your flow

1. Right-click or Cmd-click (MacOS) on the field, or click the **More options** ··· menu and select **Include Field**.

   Data Preview:



   Field List:



The change list is cleared, the field is now part of the flow data and you can see the generated row numbers in subsequent flow steps.

Source Row Number details

When you include the **Source Row Number** in your data set, the following options and considerations apply.

- The data source row numbers are applied before any data sampling or filters.
- This creates a new field called **Source Row Number** that persists throughout the flow. This field name isn't localised but can be renamed at any time.
- If a field with this name already exists, the new field name is incremented by 1. For example, **Source Row Number-1**, **Source Row Number-2** and so on.
- You can change the field's data type in subsequent steps.
- You can use this field in flow operations and calculations.
- This value is regenerated for the whole data set each time the input data is refreshed or the flow is run.
- This field isn't available for input unions.

Set header and data start row

Supported in Tableau Prep Builder version 2023.3 and later and on the web for text (.csv) files and in version 2024.1 and later for Excel (.xls) files.

You can set a specific row as a field header row and the row on which the data starts for Excel and text (.csv) files.

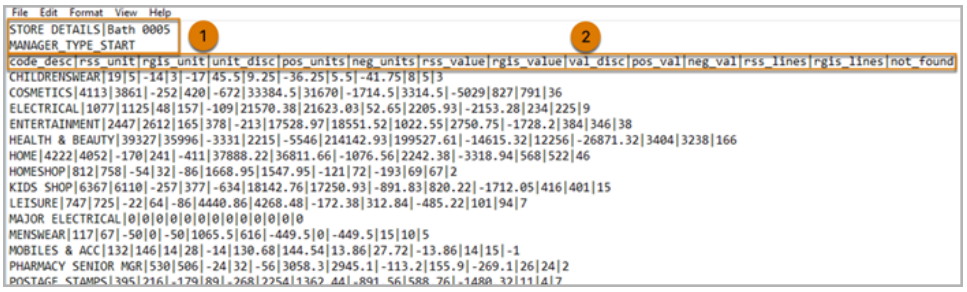
When connecting to Excel or text files, a common scenario is that the files are formatted with meta-information in the first few rows to make them human readable. By default, Tableau Prep

interprets the first rows of CSV files as the field header row. Excel files are interpreted based on field types and empty rows. Tableau Prep may select a row as the header or not include a header row.

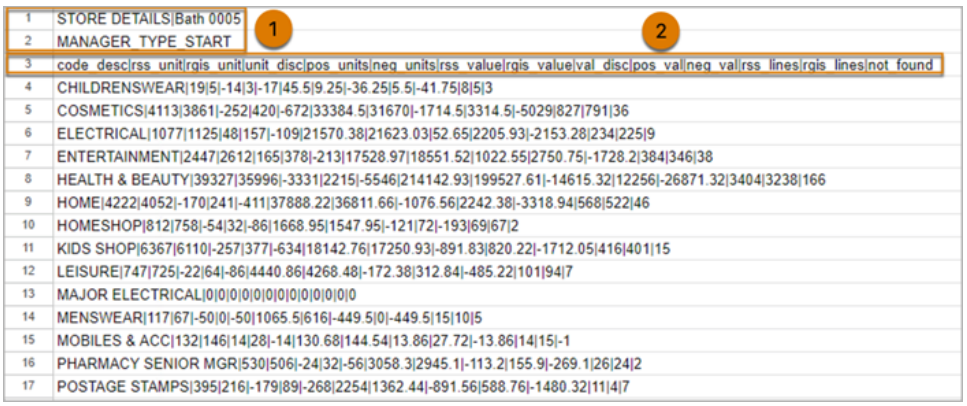
For example, from the following files, `STORE DETAILS` is interpreted as the header row.

You can exclude the metadata information (1) and provide the correct schema structure for your data by setting row 3 as the header (2) and row 4 as the row for the data to start.

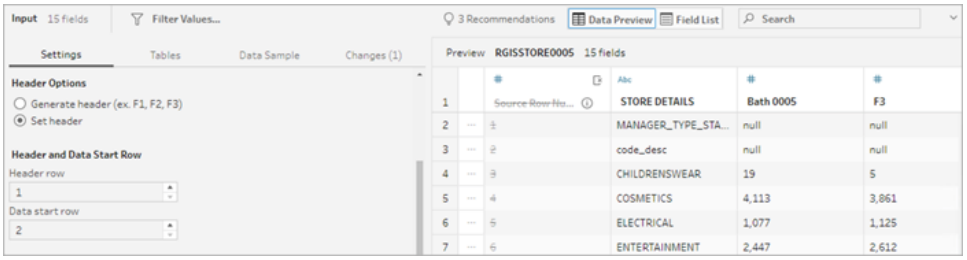
CSV files:



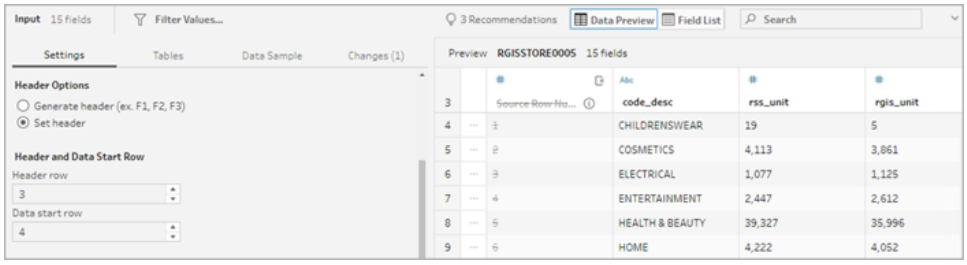
Excel files:



For example, the following shows the default settings for the row header and start row:



The following shows the data with the metadata excluded:



Note: The Data Preview doesn’t reflect changes to data sample settings.

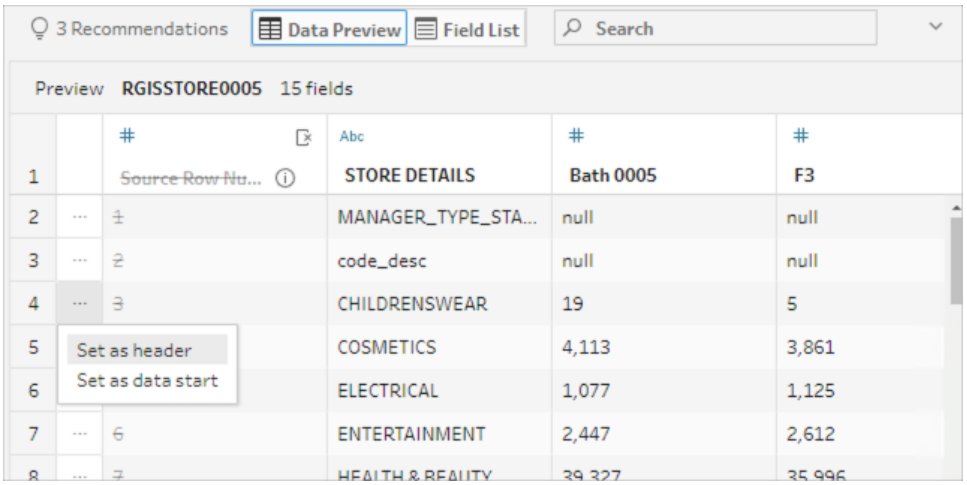
Configure the header and start row

Using the Data Preview input view, you can visually inspect the schema structure of your data and set header and start rows to exclude metadata from the input source data.

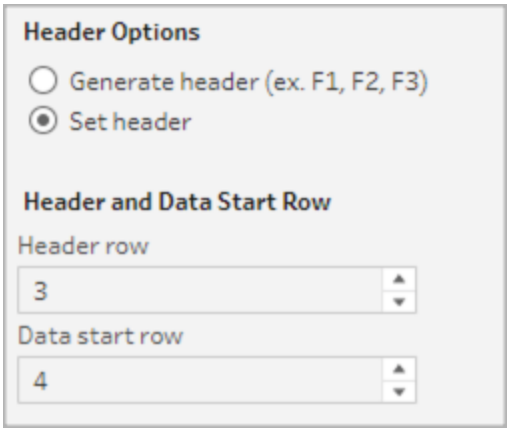
You can set the data start row to any value that’s higher than the header row value. By default, Tableau Prep sets the data start row to the next consecutive number after the header row. Any rows between the header row and data start row are ignored.

Note: Data Preview and Data interpreter are mutually exclusive. Data Interpreter only detects sub-tables in your Excel spreadsheets and doesn't support specifying the starting row for text files and spreadsheets.

1. Select the input step.
2. From the toolbar, click the **Data Preview** input view.
3. On the row that you want to set as the header, click the **More options** ··· menu and select **Set as header**.
4. On the row that you want to set as the data start row, click the **More options** ··· menu and select **Set as data start**. By default, the data start row is set to the next consecutive row number.

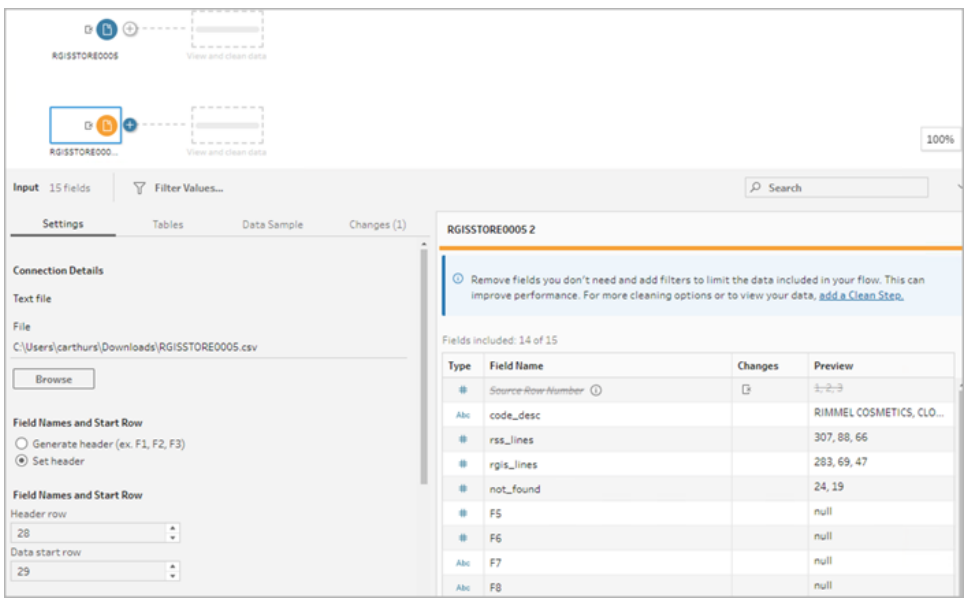


The Header Options menu displays the header row and the row number for the Data start row. Optionally, you can set the header and start row directly in the Header Options dialog.



Multiple schema in a single file

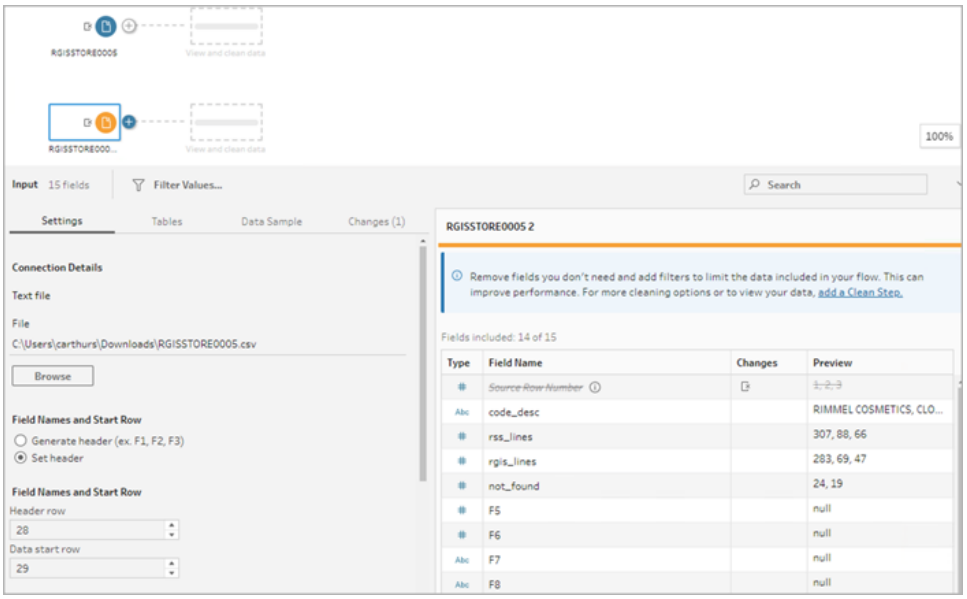
If a single file includes multiple data sources, you can create an additional input step by connecting to the same data source, then set the header and data start rows for the second data source. For example, the following file contains a data source starting on line number 3 (1) with a different, separate second schema starting on row number 28 (2).



For this type of data source, follow these steps.

1. Select the first input step.
2. From the toolbar, click the **Data Preview** input view.
3. On the row that you want to set as the header, click the **More options** ··· menu and select **Set as header**.

4. On the row that you want to set as the data start row, click the **More options** ··· menu and select **Set as data start**. By default, the data start row is set to the next consecutive row number.
5. Select the next input step.
6. Repeat the steps above for setting the header and start row for additional data sources.



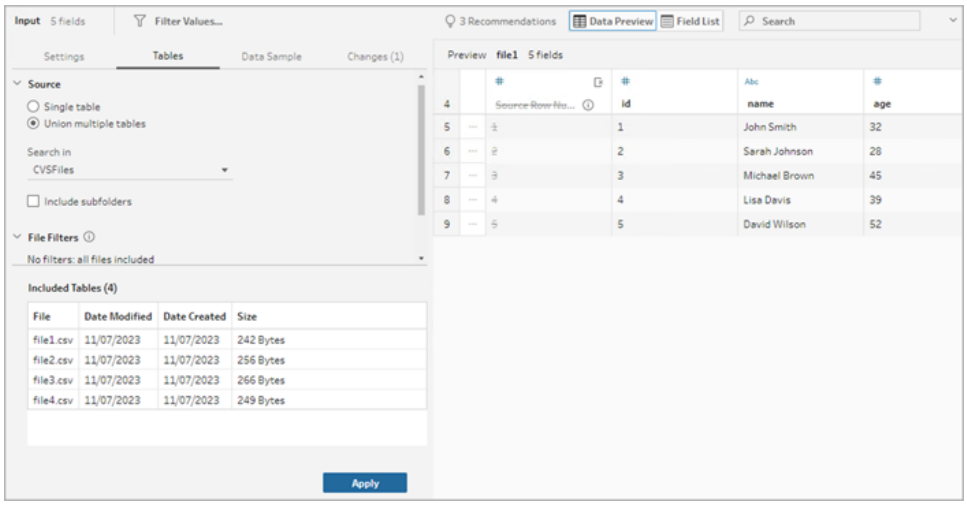
Any rows between the header row and data start row are ignored.

Union multiple tables

Supported in Tableau Prep Builder version 2024.1 and later and on the web for text (.csv) files.

You can union multiple tables from data sources that have the same schema structure and metadata row.

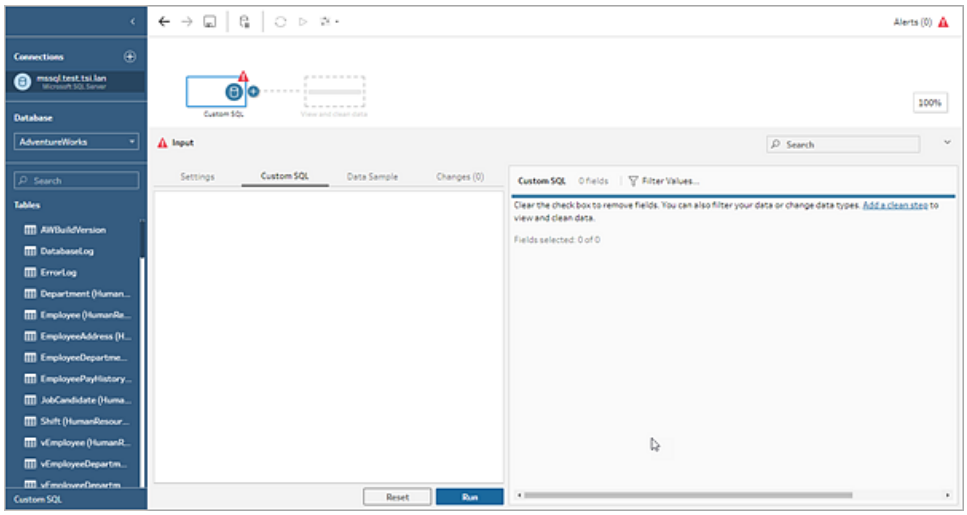
1. Connect to the files and select the first input step.
2. From the toolbar, click the **Data Preview** input view.
3. On the row that you want to set as the header, click the **More options** ··· menu and select **Set as header**.
4. On the row that you want to set as the data start row, click the **More options** ··· menu and select **Set as data start**.
5. Click the **Tables** tab and select **Union multiple tables**.



6. Click **Apply** to union the files and maintain the header and row selection for all the files in the input union. This assumes that the file structure and schema across input unioned files is the same.

Connect to a custom SQL query

If your database supports using custom SQL, you 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 **Use Custom SQL to connect to data** on page 100.

Apply cleaning operations in an input step

Only some cleaning operations are available in an Input step. You can make any of the following changes in the Input field list. Your changes are tracked in the **Changes** pane and annotations are added to the left of the Input step in the **Flow** pane and in the Input field list.

- **Hide Field**: Hide fields instead of removing them to reduce clutter in your flow. You can always unhide them if you need them. Hidden fields will still be included in your output when you run your flow.
- **Filter**: Use the calculation editor to filter values or starting in version 2023.1, you can also use the **Relative Date Filter** dialog to quickly specify date ranges for any date or date & time fields.
- **Rename Field**: In the **Field Name** field, double-click or Ctrl-click (MacOS) on the field name and enter a new field name.
- **Change Data Type**: Click the data type for the field and select a new data type from the menu. This option is currently supported for Microsoft Excel, text and PDF files, Box, Dropbox, Google Drive and OneDrive data sources. All other data sources can be changed in a clean step.

Select fields to include in the flow

> **Note**: Starting in version 2023.1 you can select multiple fields to hide, unhide, remove or include. In previous releases, you can work with one field at a time and select or clear the tick boxes to include or remove fields.

The **Input** pane shows you a list of fields in your data set. By default, all fields are included except the auto-generated field, **Source Row Number**. Use either the Data Preview or List view to manage your fields.

- **Search**: Find fields.
- **Hide Fields**: Hide fields that you want to include in your flow output but don't need to clean.
    - In the Field List, click the eye icon or select **Hide Fields** from the **More options** ··· menu.

- In the Data Preview, select **Hide Fields** from the **More options** ··· menu.

Fields are processed by the flow during run time. You can also **Unhide** fields at any time if you need them. For more information, see Hide fields.

- **Include fields**: Add fields to your flow that have been marked as removed.
  - In the Field List, select one or more rows and right-click, Cmd-click (MacOS) or click the **More options** ··· menu and select **Include Fields** to add back fields that are marked as removed.
  - In the Data Preview, click the **More options** ··· menu on the field you want to include in your flow and select **Include Field**.
- **Remove Fields**:
  - In the Field List, select one or more rows and right-click, Cmd-click (MacOS), click the "X" or click the **More options** ··· menu and select **Remove Fields** to remove fields that you don't want to include in the flow.
  - In the Data Preview, click the **More options** ··· menu on the field that you want to remove and select **Remove Field**.

Apply filters to fields in the Input step

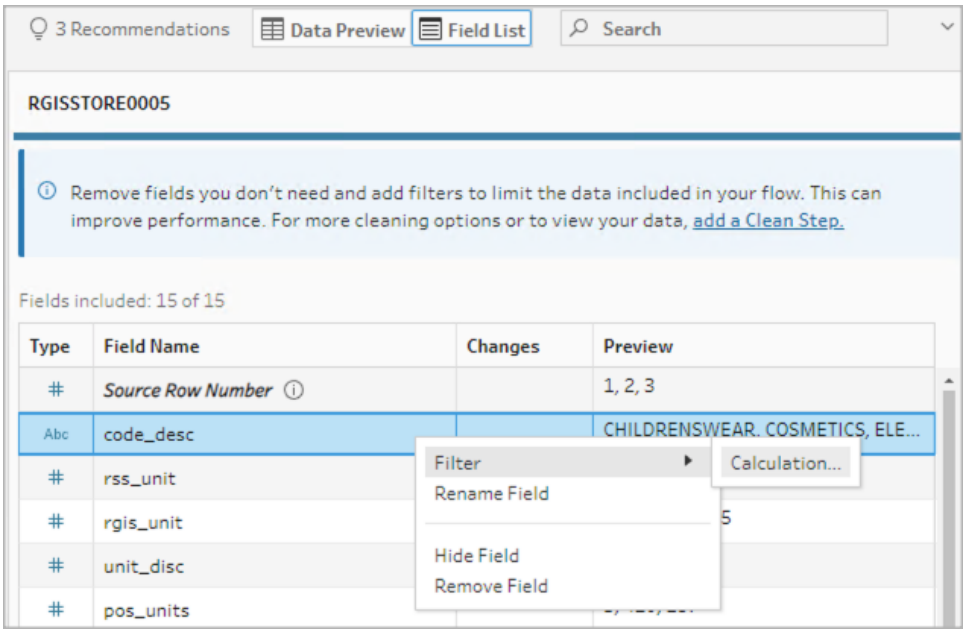
Apply filters in the input step to reduce the amount of data that you ingest from your data sources. You can gain interactive performance efficiency and a more useful data sample by eliminating the data you don't want to process when you run the flow.

In the input step, you can apply filters using the Calculation Editor. Starting in version 2023.1, you can also use the **Relative Date Filter** dialog to specify an exact date range of values to include for date and date & time field types. For more information, see "Relative Date filter" in **Filter Your Data** on page 175.

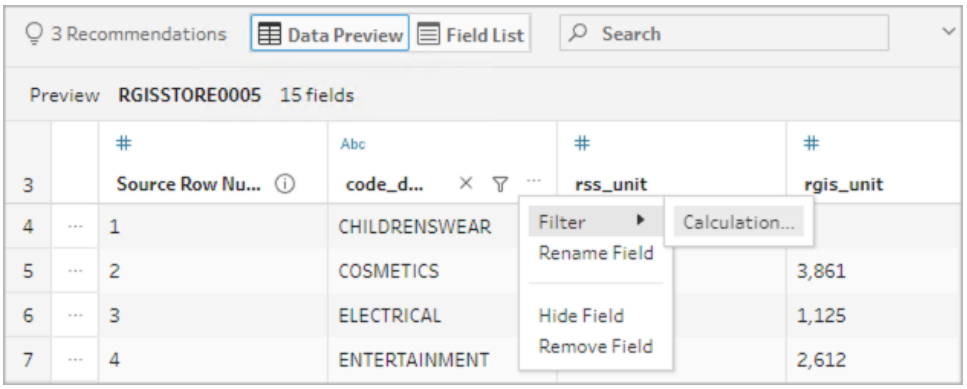
You can use other filter options in the Clean step or other step types. For more information, see **Filter Your Data** on page 175

Apply a calculation filter

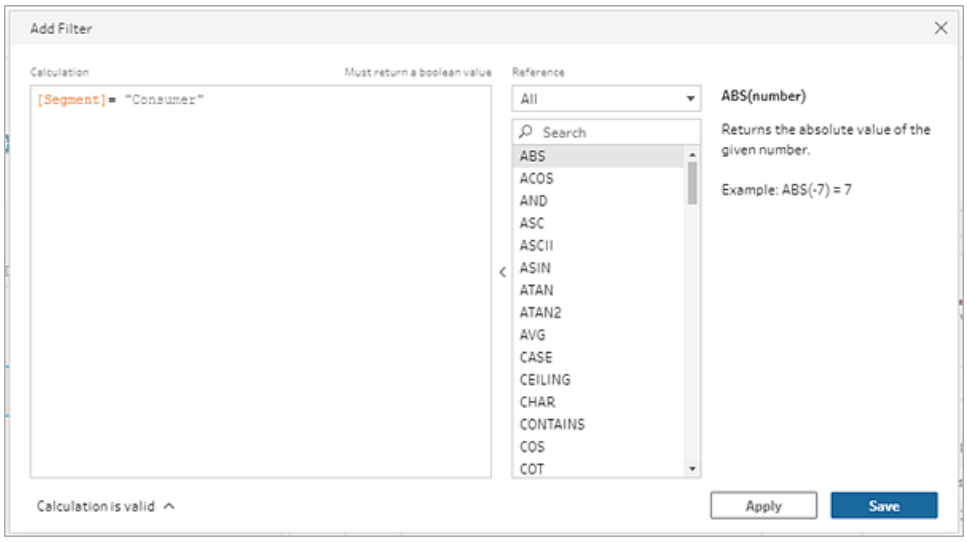
1. In the toolbar click **Filter Values**. Use one of the following methods to filter your data:
   - In the Field List, click the **More options** ··· menu from the name of the field and select **Filter** > **Calculation ...**.



- In the Data Preview, click the **More options** ··· menu from the name of the field and select **Filter** > **Calculation ...**.

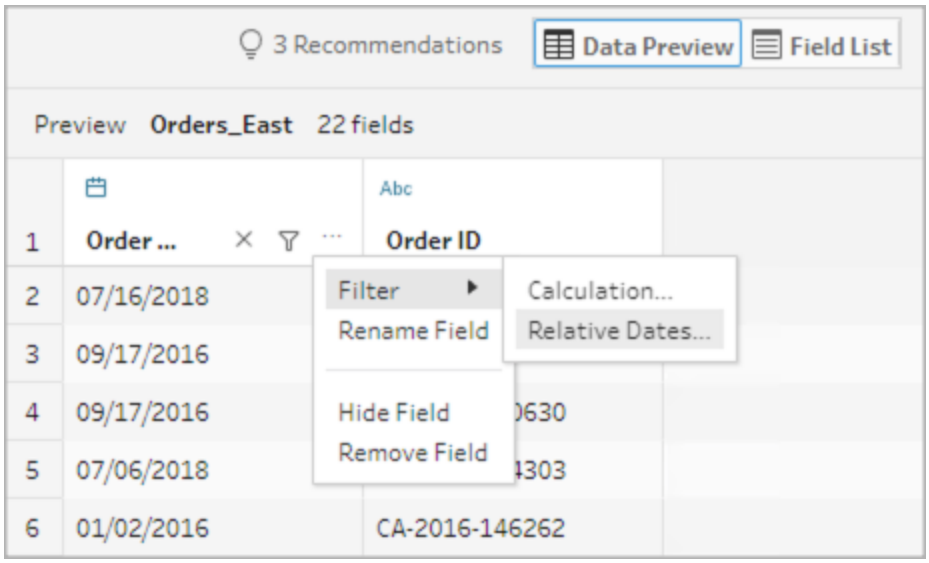


2. Enter your filter criteria in the calculation editor.



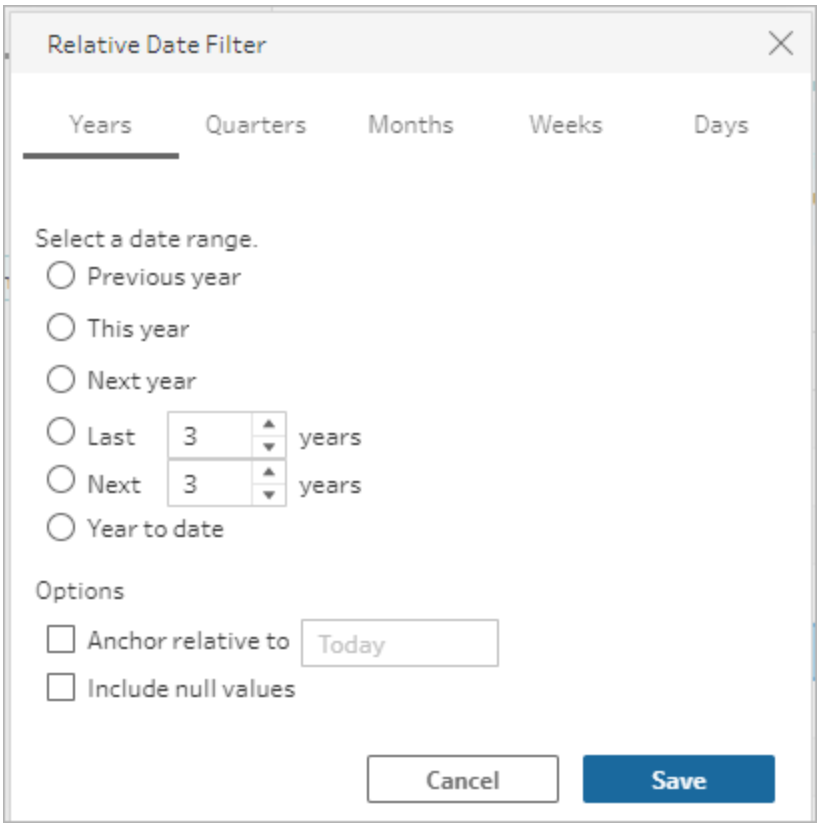
Apply a relative date filter

1. Select a field with a data type of Date or Date & Time and use one of the following methods to apply a relative date filter.
   - In the Field List, right-click, Cmd-click (MacOS) or click the **More options** ··· menu from the Field Name column and select **Filter** > **Relative Dates**.
   - In the Data Preview, click the **More options** ··· menu from the field, and select **Filter** > **Relative Dates**.



2. In the Relative Date Filter dialog, specify the exact range of years, quarters, months, weeks or days that you want to include in your flow. You can also configure an anchor relative to a specific date and include null values.

> **Note**: By default, the filter operates relative to the date that the flow is run or previewed within the authoring experience.

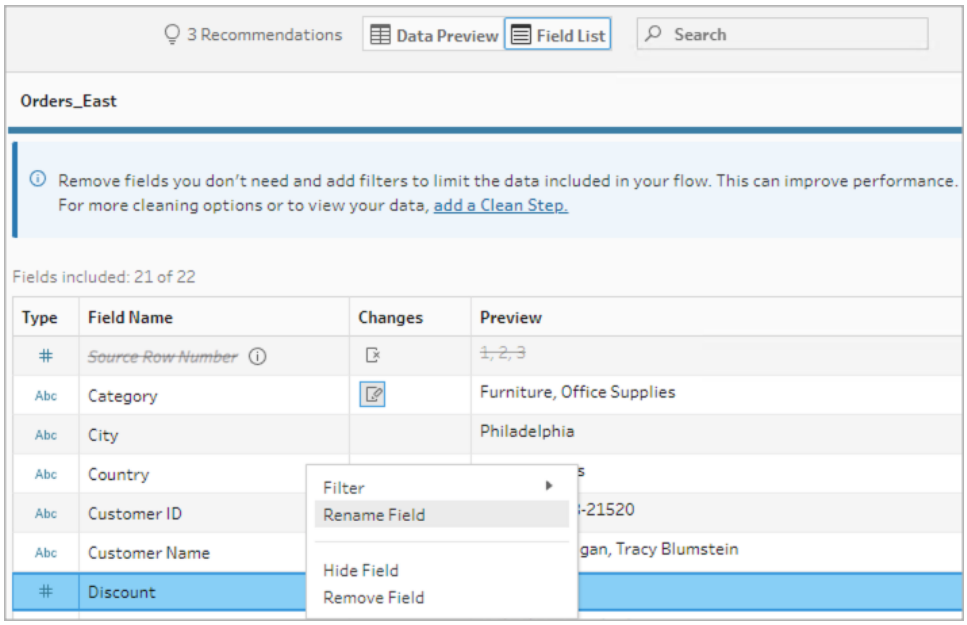


Change field names

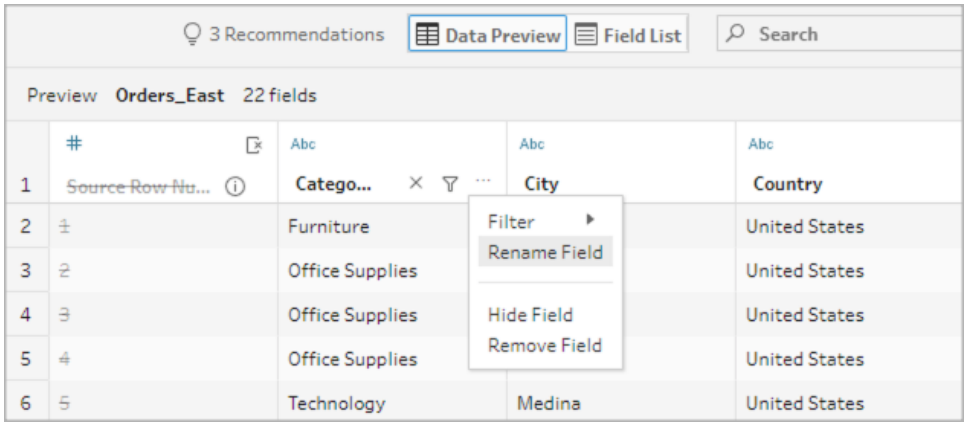
Use one of the following methods to change the name of a field.

An annotation is added in the field grid and in the flow pane to the left of the Input step. Your changes are also tracked in the Changes pane.

- In the Field List, select a field from the Field Name column and click **Rename Field**. Type the new name in the field.



- In the Data Preview, select a field and click **Rename Field**. Type the new name in the field.



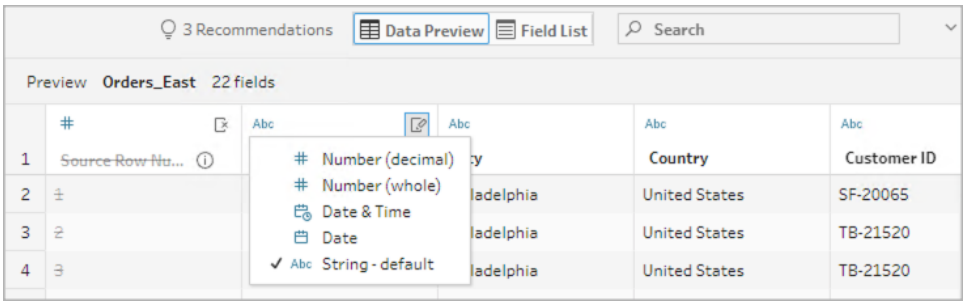
Change data types

Currently supported for Microsoft Excel, text and PDF files, Box, Dropbox, Google Drive and OneDrive data sources. All other data sources can be changed in a clean step.

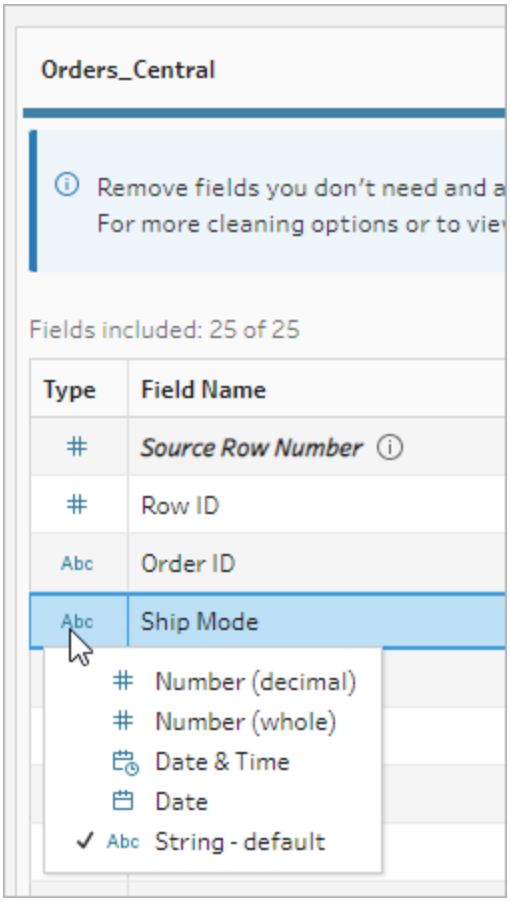
> **Note**: The data type for Source Row Number (version 2023.1 and later) can only be changed in a Clean step or other step type.

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

1. Click the data type for the field.
2. Select the new data type from the menu.
   - Data Preview:



   - Field View:



You can also change the data type for fields in other step types in the flow or assign data roles to help validate your field values. For more information about changing your data

type or using data roles, see **Review the data types assigned to your data** on page 163 and **Use Data Roles to Validate your Data** on page 190.

Configure field properties

When you work with text files, you see a **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 or configure incremental refresh settings. For more information about setting up incremental refresh for your flow, see **Refresh Flow Data Using Incremental Refresh** on page 416.

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 Builder 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.

Set your data sample size

Stratified row selection is supported in Tableau Prep Builder version 2023.3 and later.

Tableau Prep determines the maximum number of rows required to effectively explore and prepare data for a representative sample of your data set by default. Based on the Tableau Prep sample algorithm, the more fields that are in your input data, the smaller the number of rows

allowed. When data is sampled, the resulting sample may include all the rows you need – or not, depending on how the sample was calculated and returned. For example, by default, Tableau Prep uses the Quick select method to sample data. Using this method, the top rows are loaded and if your data set is large and the data is structured chronologically, you might see your earliest data sampled, but not a complete representation of all the data. If you don't see the data that you expect, you can change the data sample settings to run the query again.

When creating or editing flows using Web Authoring, the maximum number of rows that a user can select when using large data sets is configured by the administrator. For more information, see *Sample data and processing limits* in the Tableau Server or Tableau Cloud help.

Preparing your data for sampling

If you know that certain values aren't required for your analysis, remove the fields in the Input step so that the data isn't included when you author or run your flow.

If you have a large data set that triggers sampling, removing fields in the Input step increases the number of rows that Tableau Prep loads. When sampling isn't applied, removing fields in the Input step reduces the data volume that Tableau Prep loads.

After you've removed unnecessary fields and values from the data set, you can change the amount of data loaded for sampling or the sample method.

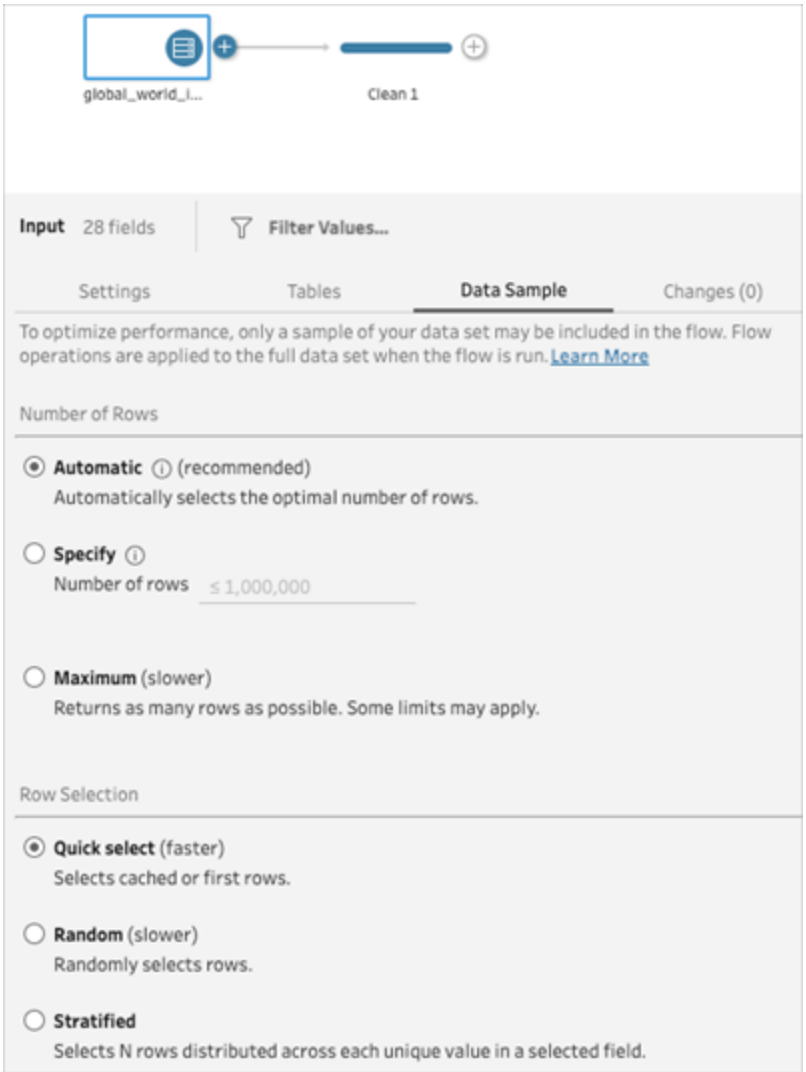
Changing the data sample settings

Sample data helps with the interactive experience and makes editing the flow more efficient compared to profiling all the data and applying changes to larger data sets as you're working. All data is used when you run the flow. Any changes you make in the sample section apply to the current flow.

To validate your data after cleaning and shaping, run the flow and view the output in Tableau Desktop.

> **Note**: Run the full flow rather than "view sample in Tableau Desktop" so you can see the entirety of your data. If you see unexpected or incorrect values that weren't in the sample, you can go back into Tableau Prep to address this.

1. Remove unnecessary fields and values from the data set.
2. Select an Input step, then click the **Data Sample** tab.



3. Select the number of rows that you want to load for data sampling. The number of rows you choose has an effect on performance.
   - **Automatic**: (default) loads data quickly and automatically calculates the number of rows so that there's enough data for a sample. The number of rows loaded is equal to or less than 393,216.
   - **Specify**: typically used to load a small number of rows so that you can understand the structure of the data and have fast load times. Specify a number of rows less than 1 million.

> Note: In Web authoring: The maximum number of rows that a user can select when using large data sets is configured by the administrator. As a user, you can select the number of rows up to that limit.

- **Maximum**: loads as much data as possible for row selection that is equal to or less than 1,048,576. Ensure that you have the High-Performance Requirements for large data sets.

4. Select the method to use for the number of rows returned for sampling. Performance can be affected when Random or Stratified are chosen.

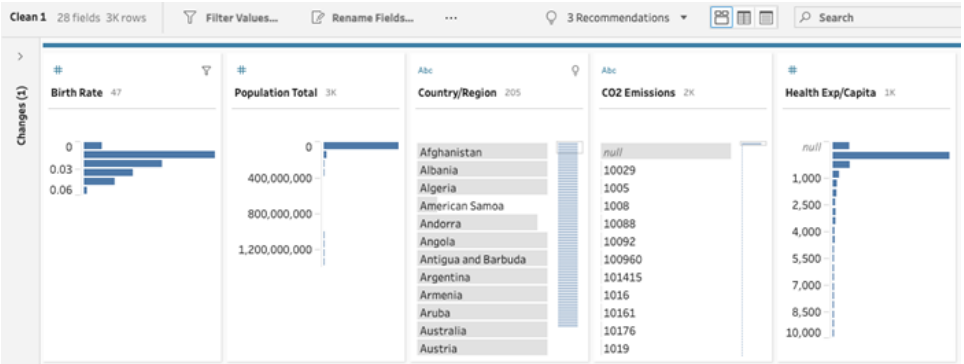
> **Note**: Row selection is only supported if your input data source supports random sampling. If your data source doesn't support random sampling, the default method of Quick Select is used.

- **Quick Select**: (default) samples data based on performance as the rows are returned as quickly as possible. Some rows may not be included in the sample. The rows used for sampling might be the first N number of rows or the rows that the database had cached in memory from a previous query. While this is almost always a faster result than random sampling, it can return a biased sample (such as data for only one year rather than all years present in the data, if the records are sorted chronologically).
- **Random**: allows you to sample a large data set and return a general representation of the entire selection of rows. Tableau Prep returns random rows based on all of the selected rows loaded. This option can affect performance when the data is first retrieved.
- **Stratified**: allows you to group by a specified field and then sample data within each subgroup. Prep returns the requested number of rows distributed across the selected field for grouping as equally as possible. In some cases, based on the data source, this can result in some values for the field having more rows than others.

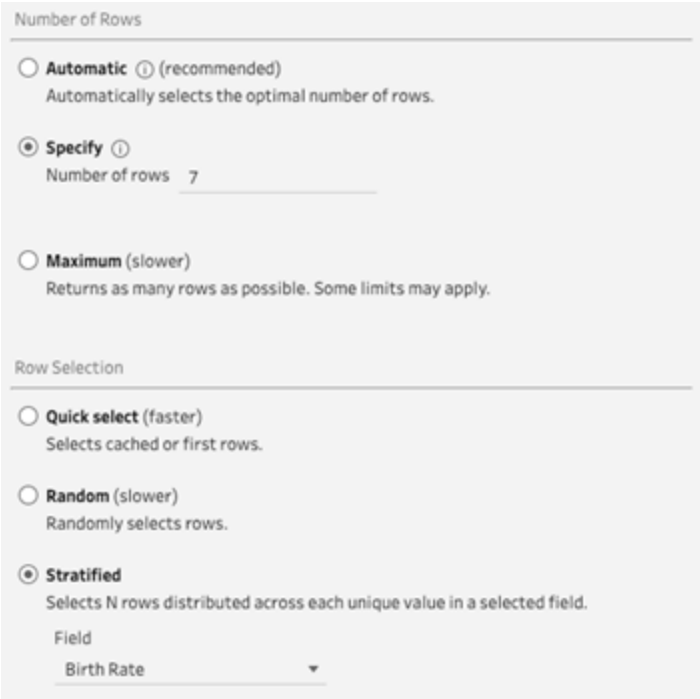
Examples

These examples are based on the global world indicators data set included with Tableau Prep. The first sample uses **Automatic** for the number of rows and **Random** for the sample method or the number of rows to return for sampling.

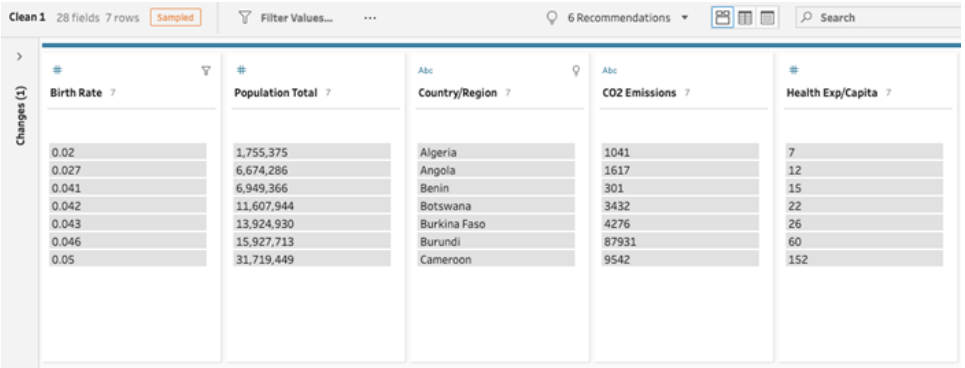
With these values selected, 3 K rows are selected at random and used to represent the entire data set.



The second sample uses **Specify** for the number of rows and **Stratified** for the sample method. The specified number of rows is set to a value of 7 and the *Birth Rate* field used for grouping.



The new sample values show a unified distribution of 7 rows of unique values across all fields.



Add More Data in the Input Step

Note: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

After you connect to your data sources and begin to build your flow you may want to refresh your data connections as new data comes in. You can also join or union data sets in the input step to make working with larger data sources more efficient.

Refresh input step data or change your connection

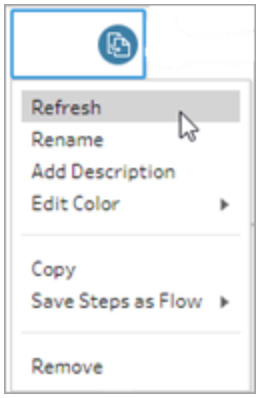
If data changes in your input files or tables after you begin working with your flow, you can refresh the Input step to bring in the new data or you can easily change and update individual input step connections without breaking your flow.

Refresh your data source

Applies to file types. Not yet supported on the web.

Do one of the following:

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



- 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.

Replace your data source

Applies to file types, data sources and extracts in Tableau Prep Builder and on the web.

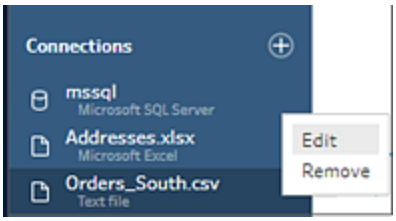
Refresh your data source by editing individual input connections or replacing individual flow data sources with a different data source.

Edit the connection

Use this option to easily refresh your credentials or replace the data source with the same data source type.

> **Note**: To maintain performance, Tableau Prep samples large data sets. If your data is sampled, you may or may not see your new data in the profile pane. You can change the settings for how your data is sampled in the **Data Sample** tab in the Input step, but it may impact performance. For more information about setting your data sample size, see **Set your data sample size** on page 121.

1. In the **Connections** pane, right-click or Ctrl-click (MacOS) on the data source and select **Edit**.



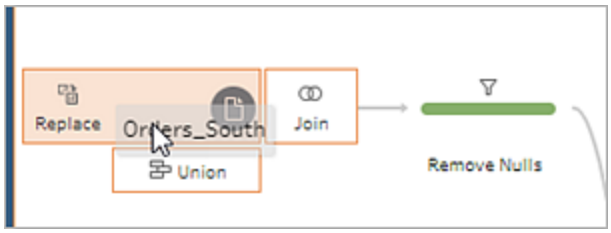
2. Re-establish your connection by signing in to the database or re-selecting the file or Tableau extract.

Replace the input connection

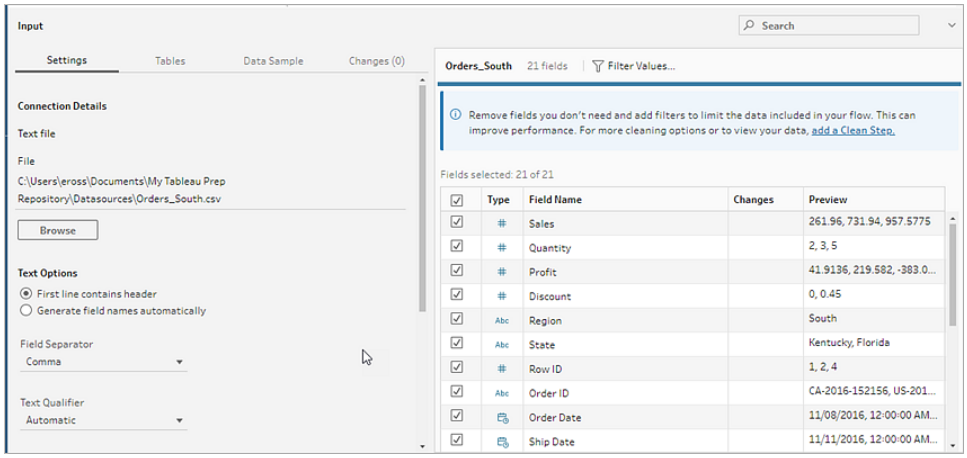
Easily replace an existing data source in your flow with any new data source without breaking the flow connection. Depending on your Tableau Prep version, you can drag and drop a new data source over your old data source or manually disconnect and reconnect your data source.

Drag and drop to replace the input connection (version 2022.4 and later)

1. From the **Connections** pane, drag the new table to the flow pane on top of the input step you want to replace and drop it on the **Replace** option.

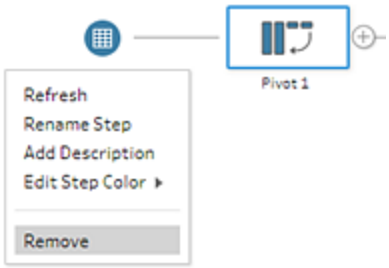


2. Reconfigure any settings and fix any error as needed.



Manually disconnect and reconnect an input data source (version 2022.3 and earlier)

1. In the flow pane, right-click on 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 new or updated data source.
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.

Union files and database tables in the Input step

Input unions can only be edited and created in Tableau Prep Builder but can be scheduled to run on the web.

When working with multiple files or database tables from a single data source, you can apply filters to search for files or use a wildcard search to find tables and then union the data to include all the file or table data in the Input step. To union files, the files must be in the same directory or sub-directory.

> **Note**: This option is not available for Tableau extracts.

New files that are added to the same folder that match the filter criteria are automatically included in the union the next time you open the flow or run it from the command line.

Packaged flow files (.tflx) won't automatically pick up new files because the files are already packaged with the flow. To include new files for packaged flows, open the flow file (.tfl) in Tableau Prep Builder to pick up the new files, then repackage the flow to include the new file data.

To union database tables, the tables must be in the same database and the database connection must support using a wildcard search. The following databases support this type of union:

- Amazon Redshift
- Microsoft SQL Server
- MySQL

- Oracle
- PostgreSQL

If you add or remove files or tables after you create the union you can refresh the Input step to update your flow with the new or changed data.

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 **Union your data** on page 367.

Union files

By default, Tableau Prep Builder 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, you can specify additional filter criteria to find the files or sheets that you want to include in the union.

Core filter criteria

In Tableau Prep Builder version 2022.1.1 and earlier, you can select from the following criteria:

- **Search in**: Select the directory to use to search for files. Tick the **Include subfolders** 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 order*, all files that include "order" in the file name are returned. Leave this field blank to include all of the files in the specified directory.

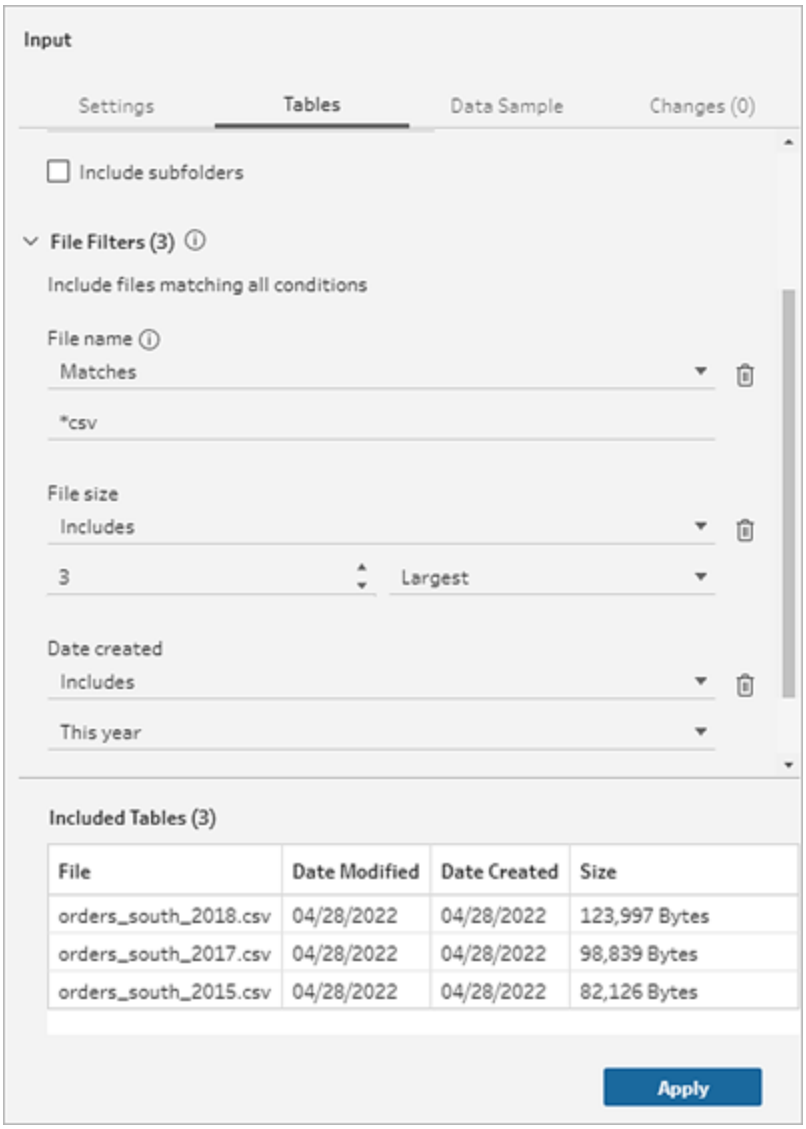
Additional filters

Supported in Tableau Prep Builder version 2022.2.1 and later and for flows published to Tableau Cloud.

> **Note**: If you use additional filters in your flow, flow scheduling is currently only available using Tableau Cloud. You can run the flow manually in Tableau Prep Builder or through the command line interface. This feature isn't compatible with Tableau Server version 2022.1 and earlier.

Starting in Tableau Prep Builder version 2022.2.1 and later, the filtering options when searching for files to union have changed. While you still specify a directory and sub-directory to search in, you can now set multiple filters to perform a more granular search.

These filtering options apply to Text, Microsoft Excel and Statistical file types. You can select multiple filters. Each filter is applied separately, in the order that you select them, top to bottom. Filters can't currently be moved around once added, but you can delete and add filters as needed.



Select from the following filters:

Filter	Description
Filename	Select **Match** or **Don't match** for a file name pattern. For example "orders*".
File size	Filter files by selecting a **Range of sizes** or **Ranked by size**. **Range of sizes**: Select from the following options: • Specify a range of values. • Select an operator of **Less than**, **Less than or equal to**, **Greater than or equal to** or **Greater than** and apply it to a single value. **Ranked by size**: Include or exclude the N largest or smallest files.
Date created	Filter files by selecting a **Range of dates**, **Relative date** or **Ranked by date**. **Range of dates**: Select from the following options: • Specify a date and time range. • Select an operator of **Before**, **Before or equal to**, **After or equal to** or **After** and apply it to a single value. **Relative date**: Include or exclude an exact range of years, quarters, months, weeks or days. You can also configure an anchor relative to a specific date. **Note**: “Last” date periods include the complete current unit of time, even if some dates haven't occurred yet. For example, if you select the last month and the current date is 7th January, Tableau will display dates for 1st January to 31st January. **Ranked by date**: Include or exclude the N newest or oldest files.
Date modified	Filter files by selecting a **Range of dates**, **Relative date** or **Ranked by date**. **Range of dates**: Select from the following options: • Specify a date and time range. • Select an operator of **Before**, **Before or equal to**, **After or equal to** or **After** and apply it to a single value. **Relative date**: Include or exclude an exact range of years, quarters, months, weeks or days. You can also configure an anchor relative to a specific date.

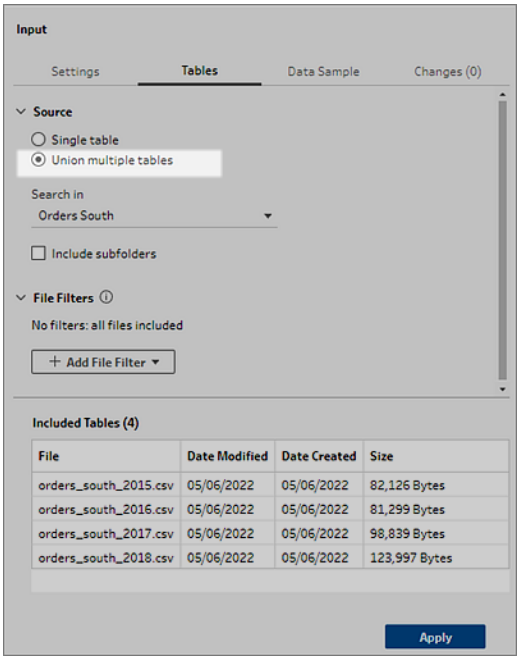
	Note: “Last” date periods include the complete current unit of time, even if some dates haven't occurred yet. For example, if you select the last month and the current date is 7th January, Tableau will display dates for 1st January to 31st January. **Ranked by date**: Include or exclude the N newest or oldest files.

Note: The instructions below vary based on your Tableau Prep Builder version.

Create an input union

Applies to Tableau Prep Builder version 2022.2.1 and later

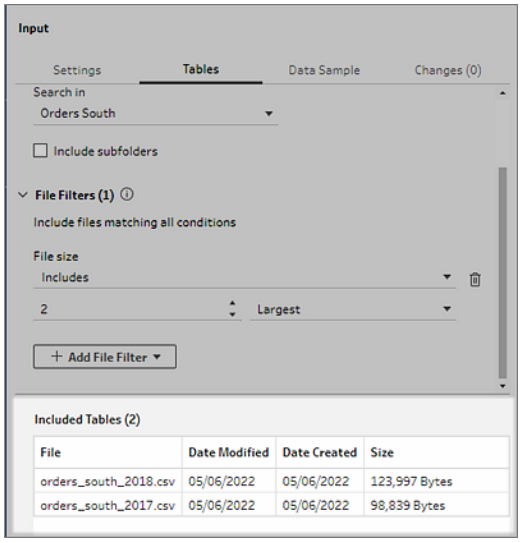
1. Click the **Add connection** button and under **Connect**, click **Text File** for .csv files, **Microsoft Excel** for Excel files or **Statistical file** for Statistical files, then select a file to open.
2. In the **Input** pane, select the **Tables** tab, and then select **Union multiple tables**.



3. Select a folder to search in. You can also include all sub-folders listed under a given directory to expand your search.

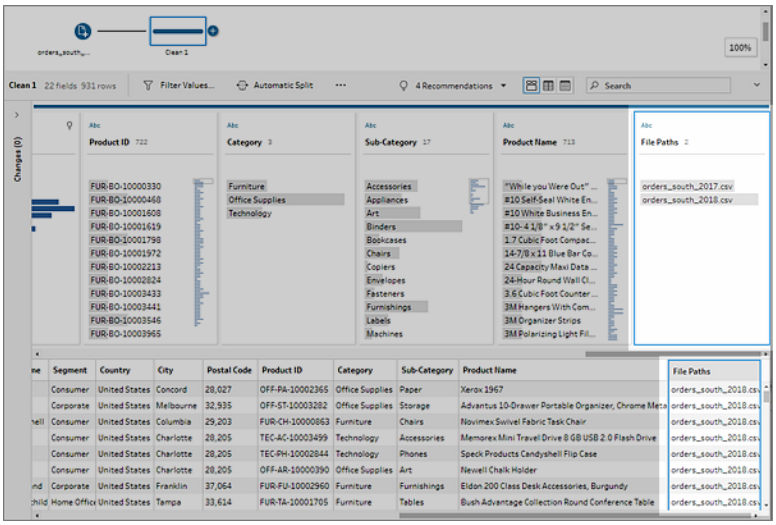
4. Click **Add File Filter** and select from the following options:
    - **File name**: Enter a name pattern to search on.
    - **File size**: Search by range of size or ranked by size.
    - **Date created**: Search by range of dates, relative date or ranked by date.
    - **Date modified**: Search by range of dates, relative date or ranked by date.
5. Click **Add File Filter** again to add more filters.

    Filter results are shown in the **Included Tables** section.



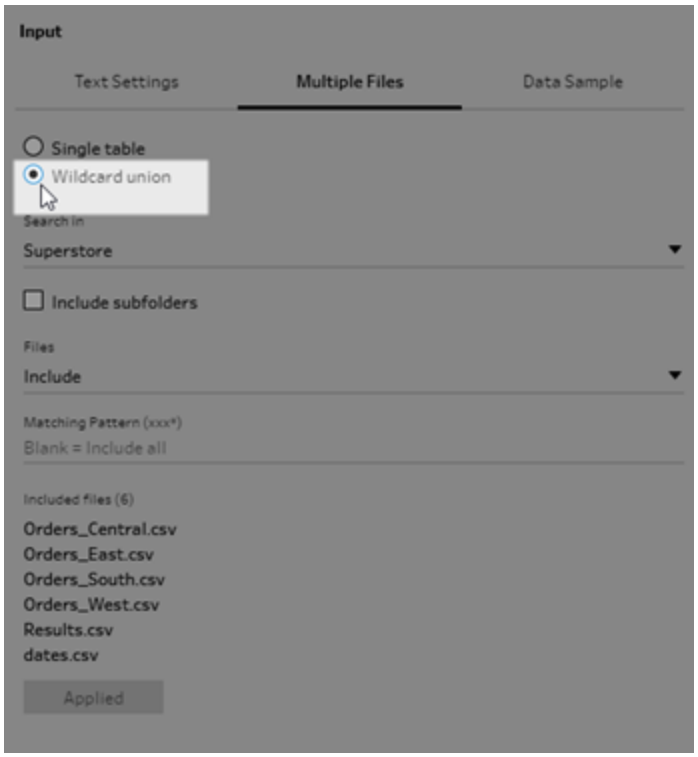
6. Click **Apply** to union the files.

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. This field is added automatically.

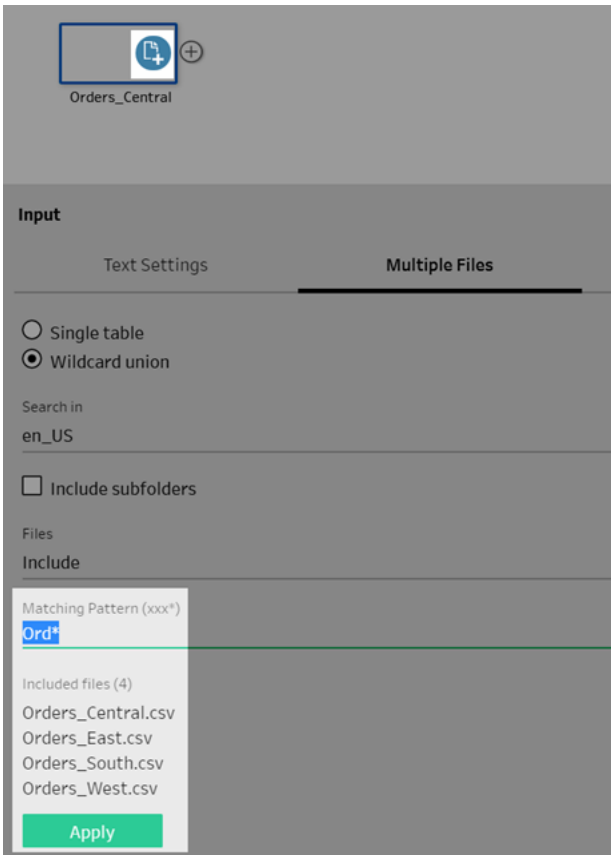


Create an input Union (version 2022.1.1 and earlier)

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**.

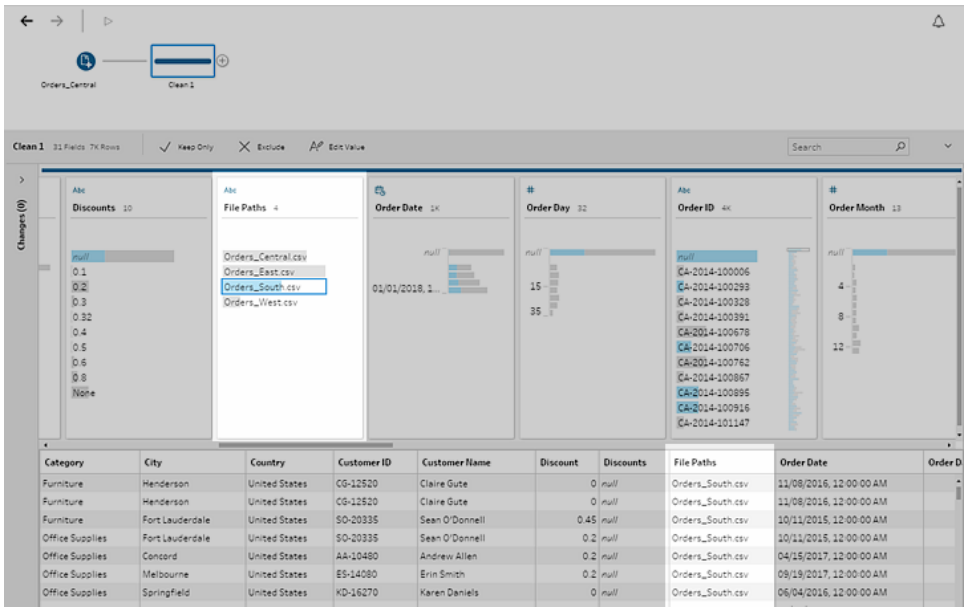


The example below shows an input union using a matching pattern. The plus sign on the file icon on the Orders_Central Input step in the **Flow** pane indicates that this step includes an input union. The files in the union are listed under **Included files**.



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. This field is added automatically.



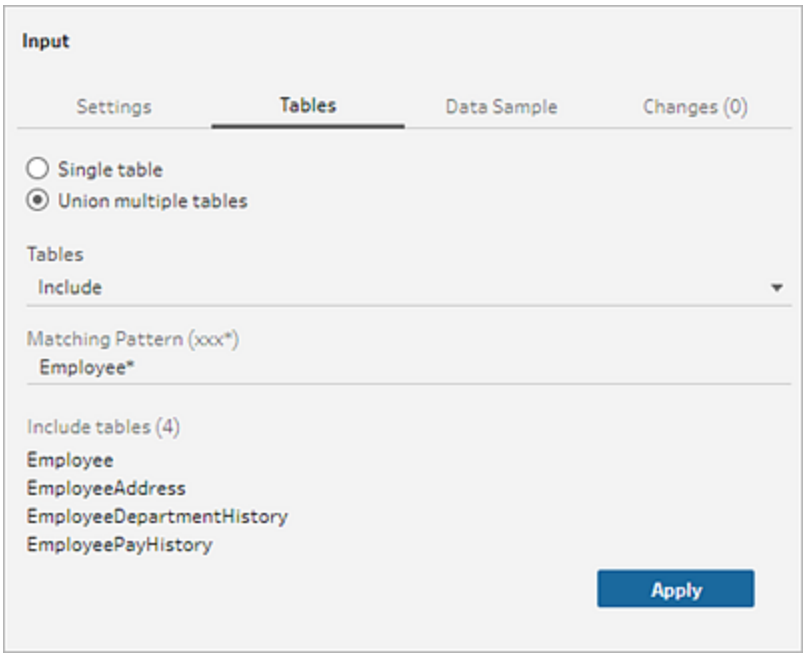
Union database tables

Supported in Tableau Prep Builder version 2018.3.1 and later

> **Note**: The input union interface for database tables has been updated in Tableau Prep Builder version 2022.2.1. Your options might look different depending on your version.

1. Click the **Add connection** button and under **Connect**, connect to a database that supports input unions.
2. Drag a table to the flow pane.
3. In the **Input** pane, select the **Tables** tab, and then select **Union multiple tables**.

   In prior versions, select the **Multiple Tables** *tab, and then select* **Wildcard union**.



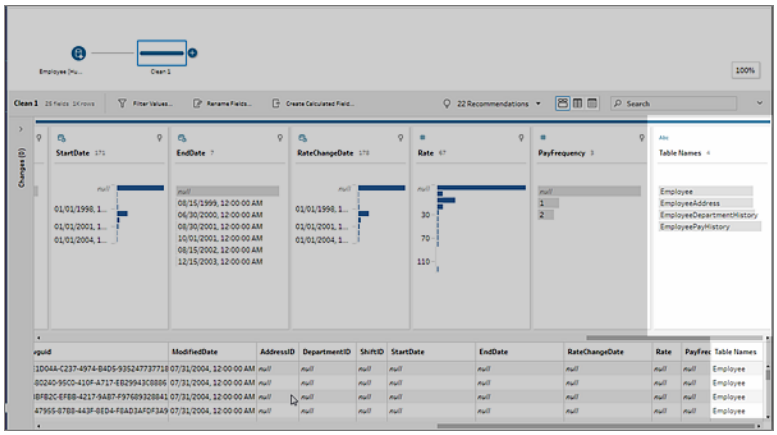
4. In the **Tables** field, select **Include** or **Exclude** from the drop-down option, then enter a matching pattern to find the tables that you want to union.

   In prior versions use search, **Tables** *and* **Matching Pattern** *options.*

   Only tables that display in the **Connections** pane in the **Tables** section can be included in the union. The input union search doesn't search across schemas or across the database connection to find tables.

5. Click **Apply** to union the table data.

   When you add a new step to the flow, you can see all the tables added to the data set in the **Table Names** field in the Profile pane. This field is added automatically.



Merge fields after a union

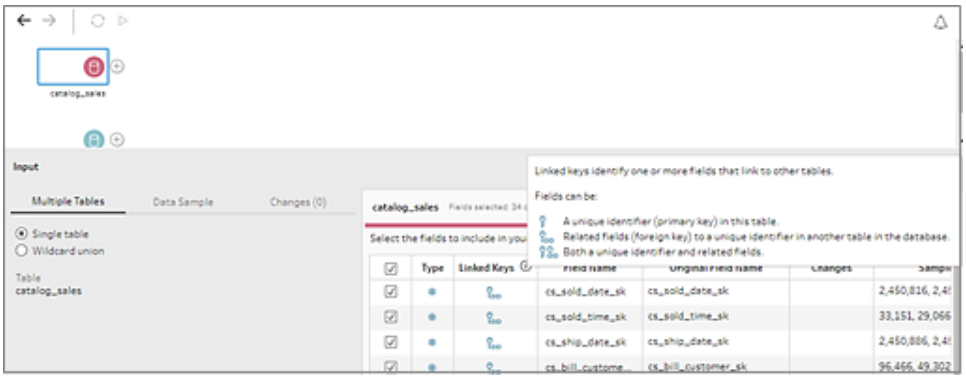
After you create a union in the input step, you might want to merge fields. You can do this in any subsequent step, except for the Input or Output steps. For more information, see **Additional merge field options** on page 373.

Join data in the Input step

In Tableau Prep Builder (version 2019.3.1 and later), and on the web, when you connect to databases that include tables with relationship data, Tableau Prep can detect and show which fields in a table are identified as the unique identifier and which fields are identified as a related field as well as show the related table names for these fields.

A new column called **Linked Keys** shows in the Input pane and shows the following relationships if they exist:

- Unique identifier. This field uniquely identifies each row in the table. There can be multiple unique identifiers in a table. The values in the fields must be unique and cannot be blank or null.
- Related field. This field relates the table to another table in the database. There can be multiple related fields in a table.
- Both Unique Identifier and related field. The field is a unique identifier in this table and also relates the table to another table in the database.



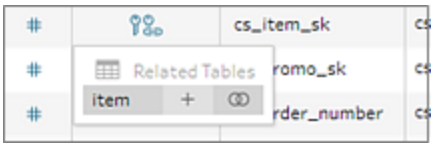
You can leverage these relationships to quickly find and add the related tables to your flow or create joins from the Input step. This feature is available for any supported database connector where table relationships are defined.

1. Connect to a database (such as Microsoft SQL Server) that contains relationship data for fields, such as unique identifiers or related fields (foreign key).

2. In the Input pane, click on a field that is marked as a related field or as both a unique identifier and related field.

   A dialog opens that shows a list of related tables.

3. Hover on the table that you want to add or join and click the plus button to add the table to your flow, or click the join button to create a join with the selected table.

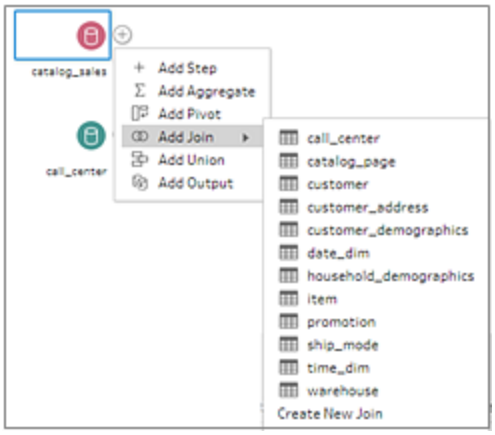


   If you create a join, Tableau Prep uses the defined field relationship to join the tables and shows you a preview of the join clauses that it will use to create the join.

4. Alternatively, you can join related tables from the menu in the Flow pane. Click the plus icon, then select **Add Join** to see a list of related tables. Tableau Prep creates the join

based on the fields that make up the relationship between the two tables.

> **Note**: If your table doesn't have table relationships defined, this option is not available.



For more information about working with joins, see **Join your data** on page 360.

Build and Organise your Flow

> **Note**: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

After you connect to the data that you want to include in your flow, you can begin cleaning and shaping your data by adding new steps to the flow or inserting steps in between existing steps.

To organise your flow, you can change the default step colours, add descriptions to provide context for your steps or cleaning actions or reorganise your flow layout to make complex flows easier to follow.

Add or insert steps

As you build out your flow, you can add different step types to perform the actions that you need. For example add a **Clean Step** to do things like split fields, apply filters, or perform a variety of other operations to clean dirty data. Use a **Join** or **Union** step type to combine data tables or add a **Script** step type to incorporate R or Python scripts into your flow.

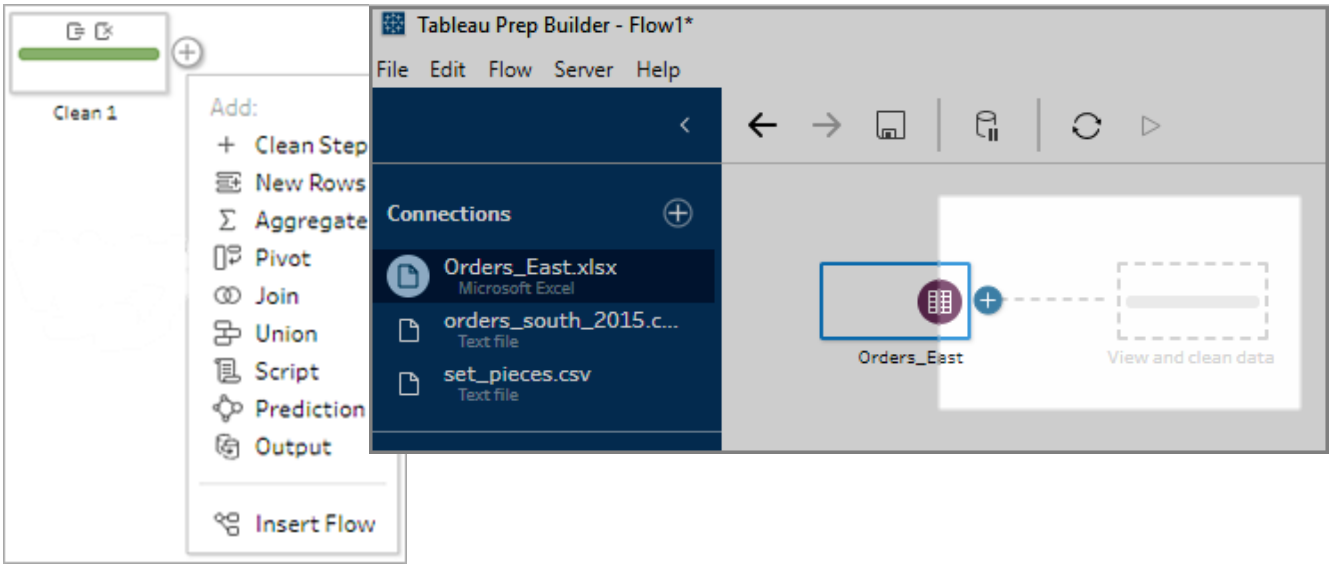
As your flow begins to takes shape, you may need to go back to earlier steps in your flow and insert different step types to perform various actions like adding an additional cleaning step or aggregating your data to use the same level of detail as a later step.

> **Note**: The menu options that you see will vary depending on your Tableau Prep Builder version and whether you are adding a step to build out the next step in the flow versus inserting a step between existing steps. If you are using Tableau Prep Builder version 2019.3.1 or earlier, refer to that section to see your menu options.

You can't add input steps using these menus. Instead you'll need to drag tables from the **Connections** pane to the Flow pane. For more information, see **Connect to Data** on page 75.

Add steps

After you connect to your data and drag a table onto the canvas, click the plus button to select a step type from the menu, or click on the suggested clean step (Tableau Prep Builder version 2020.3.3 and later and on the web) to automatically add a cleaning step to your flow.



Select a step type:

- **Clean Step**: Add a cleaning step to perform a variety of cleaning actions. For more information about the different cleaning actions that are available, see **Clean and Shape Data** on page 227.

  > **Note**: In Tableau Prep Builder version 2019.4.2, the **Add Branch** option was replaced with the **Clean Step** option. To split your flow into different branches, click the plus button between two existing steps and select a step type from the **Add** menu.

- **New Rows**: Generate new rows to fill gaps in your sequential data set. For more information, see **Fill Gaps in Sequential Data** on page 272.

Aggregate: Create an **Aggregation** step to select fields and change their level of detail. For more information, see **Aggregate and group values** on page 359.

- **Pivot**: Create a **Pivot** step to perform a variety of pivot options such as converting column data to rows, or row data to columns. You can also set up a wildcard pivot to automatically add new data to your pivot. For more information, see **Pivot Your Data** on page 332.
- **Join**: Create a **Join** step to combine data tables. When you create a join from the menu option, you must manually add the other input to the join and add your join clauses. As an alternative, you can drag and drop a step (shown below) to join files automatically. For more information about creating a join, see **Join your data** on page 360.

If you connect to databases that include tables with relationship data, you can also create a join from the menu in the Flow pane. For more information about joining tables using this method, see **Join data in the Input step** on page 141.

- **Union**: Create a Union step. Add tables to the union by dragging them to the step and dropping them on the **Add** option that displays. As an alternative, you can drag and drop one step onto another step to union files. For more information about creating a union, see **Union your data** on page 367.
- **Script** (Tableau Prep Builder version 2019.3.1 and later and on the web): Create a Script step to include R and Python scripts in your flow. Script steps are not currently supported in Tableau Cloud. For more information, see **Use R and Python scripts in your flow** on page 341.
- **Prediction**: Use Einstein Discovery-powered models to bulk score predictions for the data in your flow. For more information, see **Add Einstein Discovery Predictions to your flow** on page 374.
- **Output**: Create an Output step to save the output to an extract file (.hyper), a .csv file, publish the output as a data source to a server or write your flow output to a database.

  Saving Output steps to a file is not currently supported on the web. For more information about output types, see **Save and Share Your Work** on page 383.

- **Paste**: Add copied steps from the same flow. For more information about copying and pasting steps in the same flow, see **Clean and Shape Data** on page 227.
- **Insert Flow** (Tableau Prep Builder version 2019.3.2 and later and on the web): Add flow steps that were saved from another flow into your current flow. You can add them to the end of a step or insert them between existing steps. For more information about using saved flow steps in your flow, see **Create reusable flow steps** on page 269

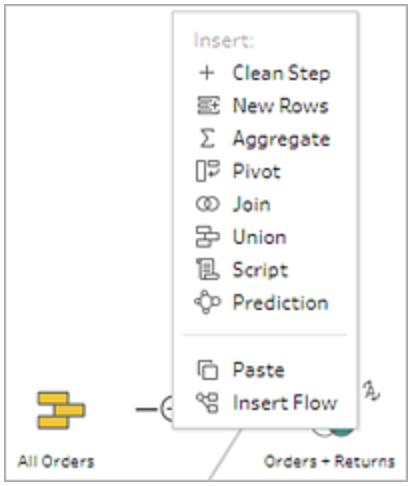
> **Note**: This option was added to this menu in Tableau Prep Builder version 2019.4.2. In prior versions, you could add flow steps using right-click or Ctrl-click (MacOS) in the white space of the flow pane.

Insert steps

Insert a step between existing steps. Input and Output step types aren't available from this menu. The options vary depending on your product version. If you are using an earlier version of Tableau Prep Builder, refer to the **Version 2019.3.1 and earlier** section below.

1. Hover in the middle of the flow line where you want to insert a step until the plus icon appears. Then click the icon and select a step type.

> **Note**: Your options may look different depending on your product version. For example **Insert Flow** was added to this menu in Tableau Prep Builder version 2019.4.2.



2. Select a step type:

   - **Clean Step**: Insert a cleaning step between existing steps to perform a variety of cleaning actions. For more information about the various cleaning actions you can use, see **Clean and Shape Data** on page 227.
   - **New Rows**: Generate new rows to fill gaps in your sequential data set. For more information, see **Fill Gaps in Sequential Data** on page 272.
   - **Aggregate**: Insert an **Aggregation** step between existing steps to select fields and change their level of detail. For more information, see **Aggregate and group values** on page 359.
   - **Pivot**: Insert a **Pivot** step between existing steps to perform a variety of pivot options, such as converting column data to rows or row data to columns. You can also set up a wildcard pivot to automatically add new data to your pivot. For more information, see **Pivot Your Data** on page 332.
   - **Join**: Insert a **Join** step between existing steps . When you create a join from the menu option, you must manually add the other input to the join and add your join clauses. As an alternative, you can drag and drop a step (shown below) to join files automatically.

     For more information about creating a join, see **Join your data** on page 360.

     If you connect to databases that include tables with relationship data you can also create a join from the menu in the Flow pane. For more information about joining tables using this method, see **Join data in the Input step** on page 141.
   - **Union**: Insert a Union step. Add tables to the union by dragging them to the step and dropping them on the **Add** option that displays. As an alternative, you can drag and drop one step onto another step to union files. For more information about creating a union, see **Union your data** on page 367.
   - **Script** (Tableau Prep Builder version 2019.3.1 and later and on the web): Insert a Script step to include R and Python scripts in your flow. Script steps are not

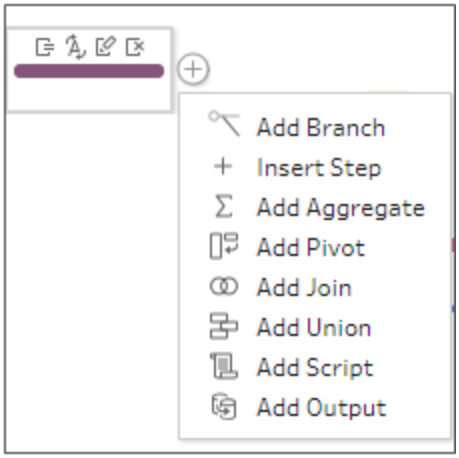
currently supported in Tableau Cloud. For more information, see **Use R and Python scripts in your flow** on page 341.

- **Prediction**: Use Einstein Discovery-powered models to bulk score predictions for the data in your flow. For more information, see **Add Einstein Discovery Predictions to your flow** on page 374.
- **Paste**: Insert copied steps from the same flow between existing steps. For more information about copying and pasting steps in the same flow, see **Clean and Shape Data** on page 227.
- **Insert Flow** (Tableau Prep Builder version 2019.3.2 and later and on the web): Insert flow steps that were saved from another flow into your current flow. You can add them to the end of a step or insert them between existing steps. For more information about using saved flow steps in your flow, see **Create reusable flow steps** on page 269.

> **Note**: This option was added to this menu in Tableau Prep Builder version 2019.4.2. In prior versions, you could insert flow steps using right-click or Ctrl-click (MacOS) in the white space of the flow pane.

Version 2019.3.1 and earlier

1. Hover over a step until the plus ⊕ icon appears, then click the icon and select a step type. **Insert Step** inserts a cleaning step between steps. All other options will create a branch from the flow.



2. Select from the following options:

   - **Add Branch**: Split your flow into different branches.
   - **Insert Step**: Insert a cleaning step between existing steps to perform a variety of cleaning actions. For more information about the various cleaning actions you can use, see **Clean and Shape Data** on page 227.
   - **Add Aggregate**: Create an **Aggregation** step where you can select the fields that you want to aggregate or group. For more information, see **Aggregate and group values** on page 359.
   - **Add Pivot**: Create a **Pivot** step where you can perform a variety of pivot options to convert column data to rows, or row data to columns. For more information, see **Pivot Your Data** on page 332.
   - **Add Join**: Create a **Join** step where you can manually add the other input to the join and add the join clauses. As an alternative, you can drag and drop a step to join files. The following example shows dragging the Orders_Central Input step and dropping it on **Join**:

   For more information about creating a join, see **Join your data** on page 360.

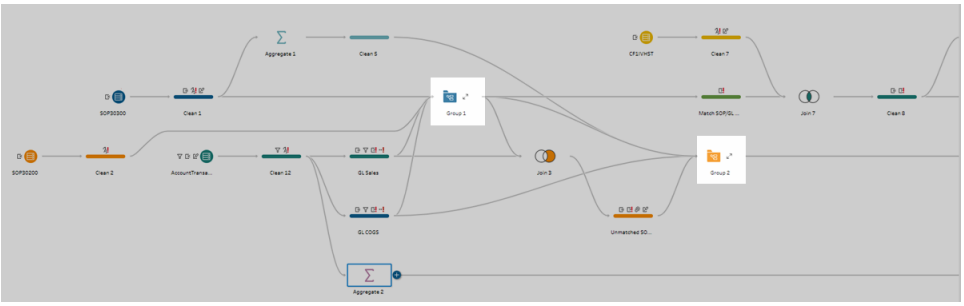
In Tableau Prep Builder version 2019.1.3 and later, if you connect to databases that include tables with relationship data, you can also create a join from the menu in the Flow pane. For more information about joining tables using this method, see **Join data in the Input step** on page 141.

- **Add Union**: Create a Union step. Add tables to the union by dragging them to the step and dropping them on the **Add** option that displays. As an alternative, you can drag and drop one step onto another step to union files. For more information about creating a union, see **Union your data** on page 367.
- **Add Script**(version 2019.3.1 and later): Create a Script step to include R and Python scripts in your flow. For more information, see **Use R and Python scripts in your flow** on page 341.
- **Add Output**: Select this option to save the output to an extract file (.hyper), a .csv file or publish the output as a data source to a server.

Group steps

Supported in Tableau Prep Builder version 2020.3.3 and later and on Tableau Server or Tableau Cloud starting in version 2020.4.

Use the Group option to compartmentalise sections of large complex flows into folders to make it easier to follow, troubleshoot or share your flow with others. You can change the colour of the group, add a description, copy and paste the grouped steps to other areas of your flow or, in Tableau Prep Builder, even save the grouped steps to a file on your server to reuse them in other flows.



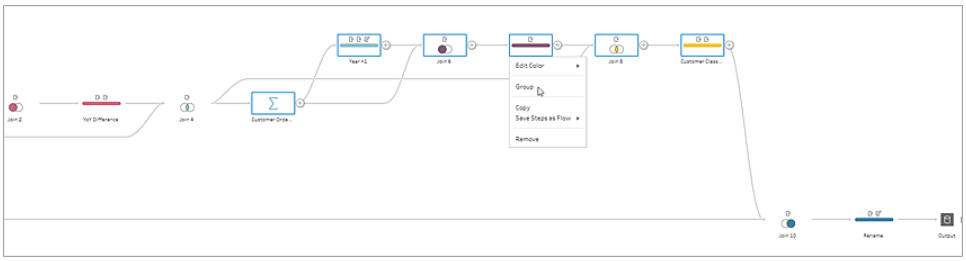
Requirements to group steps

- Steps must be directly connected with a flow line.
- Steps can only be included in one group at a time.

- Groups can't be nested.
- You can add or remove steps from a group at any time, as long as you maintain the flow line connections between steps in the group. This also applies to removing steps from a flow that are already included in a group. In that scenario, the group is automatically ungrouped.

Create a group

Select a set of connected steps in your flow (you can also drag to select multiple steps in one click), then right-click or Ctrl-click (MacOS) on the selected steps and select **Group** from the menu.

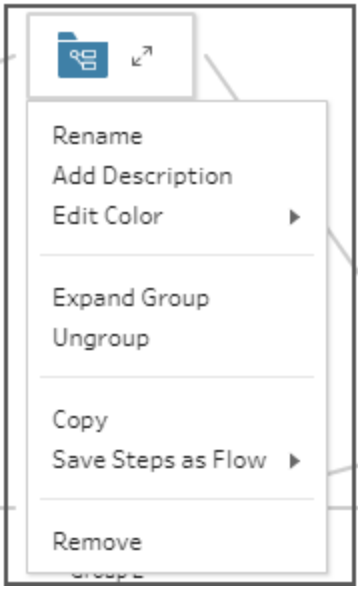


After you create the group, you can do any of the following:

- Click the double arrows to expand or collapse the group at any time.
- Add more steps to the group by dragging a connected step and dropping it onto the collapsed folder.
- Remove steps from the group. In the expanded state, right-click or Ctrl-click (Mac OS) on a step and select **Remove from Group**.

  > **Note**: This option isn't available if you try to remove a step that breaks the continuity of the group.

- In the collapsed state, right-click or Ctrl-click (MacOS) to open the menu and select from the following options:



- **Rename**: Change the group name.
- **Add Description**: Enter a description for the group
- **Edit Colour**: Change the colour of the group folder. This won't change the colour of the individual steps in the group.
- **Expand Group**: Show all the steps in the group. You can also click the double arrows to expand the group.
- **Ungroup**: Remove all the steps from the group and delete the group.
- **Copy**: Copies the group and all of the steps in the group to your clipboard to paste elsewhere in your flow. For more information about using copy and paste, see **Copy and paste steps** on page 263.
- **Save Steps as Flow** (Tableau Prep Builder only): Save your grouped steps locally to a file on your computer or publish it to Tableau Server or Tableau Cloud to share with others or use it in other flows. For more information about saving steps for reuse, see **Create reusable flow steps** on page 269.
- **Remove**: Removes the group and all the steps in the group from the flow.

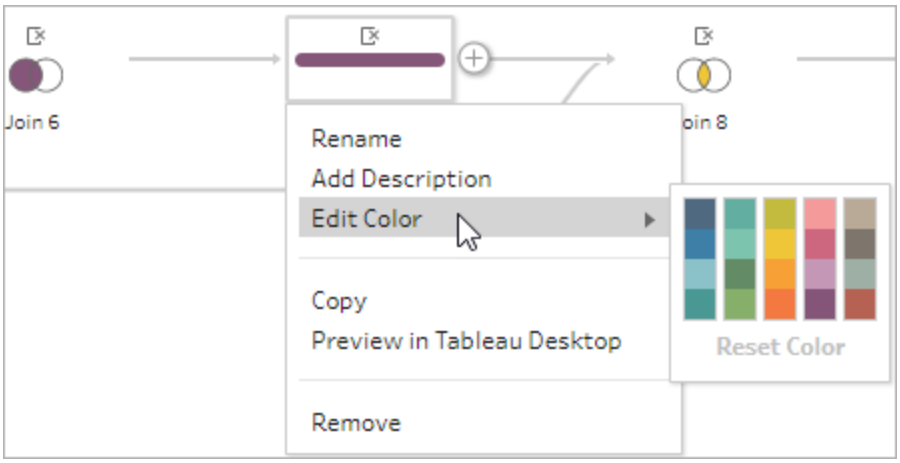
- (version 2021.1.2 and later) In the expanded state, right-click or Cntrl-click (MacOS) in the expanded group area to open the menu to collapse the group or ungroup the steps.

Change the flow colour scheme

Tableau Prep assigns each step in your flow a colour by default. This colour 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 colour scheme for your steps do the following:

1. Select one or more steps.
2. Right-click or Ctrl-click (MacOS) on a selected step and select **Edit Colour**.



3. Click on a colour in the colour palette to apply it.

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

- Click **Undo** from the top menu.
- Ctrl+Z or Command-Shift-Z (MacOS).
- Select the steps you changed, right-click on a selected step and select **Edit Colour**, then select **Reset Colour** from the bottom of the colour palette.

Remove steps from the flow

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

> **Note**: You can't remove flow lines coming into or out of a collapsed step group. You must either expand the group or ungroup the steps first.

- To remove a step or flow line, select the step or line you want to remove, right-click on 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 or Ctrl-click (Mac OS) on one of the selected steps and select **Remove**.
  - Press Ctrl+A or Cmd+A (MacOS) to select all elements in the flow, or press Ctrl+click or Cmd+Click (MacOS) to select specific elements, and then press the **Delete** key.

Add descriptions to flow steps and cleaning actions

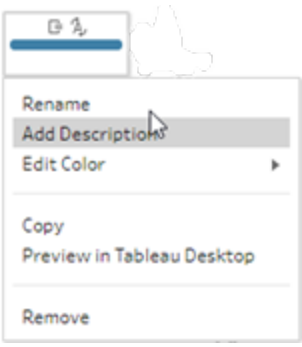
As you build your flow and perform various cleaning operations, you might want to add a description to help others who might later look at or work with your flow to better understand your steps. You can add a description to any individual step in your flow directly on the Flow pane, to any step group or to any cleaning action in the **Changes** pane to provide additional context for your changes. The description can be up to 200 characters long.

For more information about viewing changes in the Changes pane, see **View your changes** on page 241.

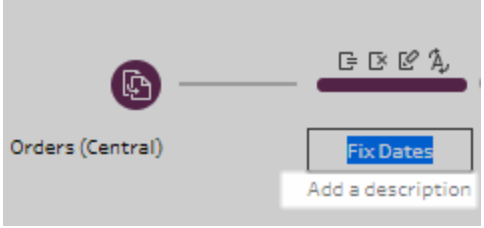
Add a description to flow steps

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.

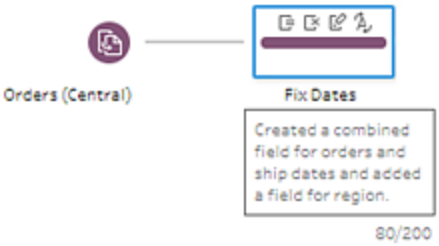
1. In the Flow pane, select a step.
2. Do one of the following:
   - Right-click or Ctrl-click (MacOS) on the step and select **Add Description** from the menu.



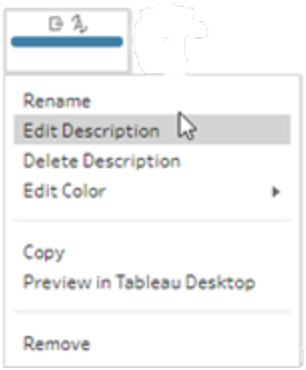
   - Double-click in the name field for the step, then click on **Add a description**.



3. Type your description in the text box.



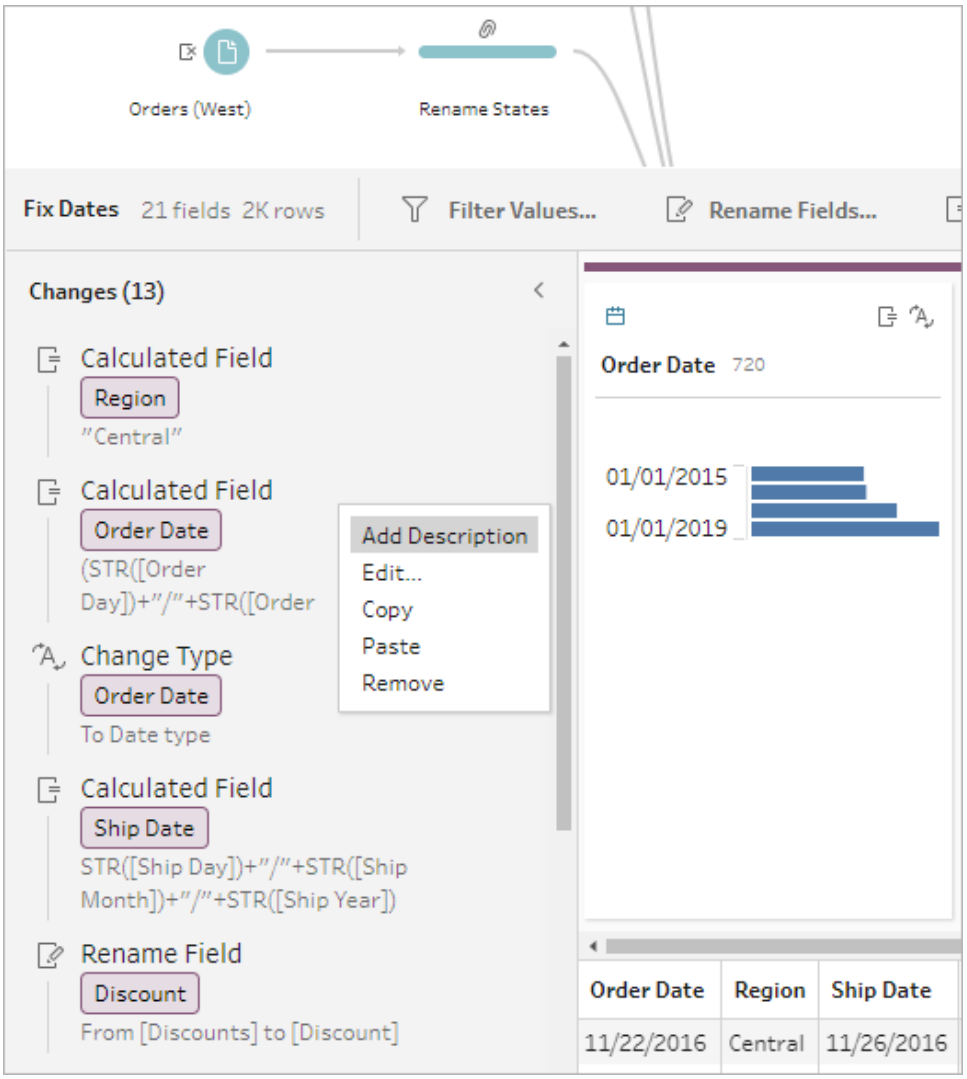
4. Click outside the text box or press Enter to apply your changes. By default, the description displays underneath the step. To hide the description click the message icon.
5. To edit or delete the description, right-click or Ctrl-click (MacOS) on the step or description. Then from the menu, select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



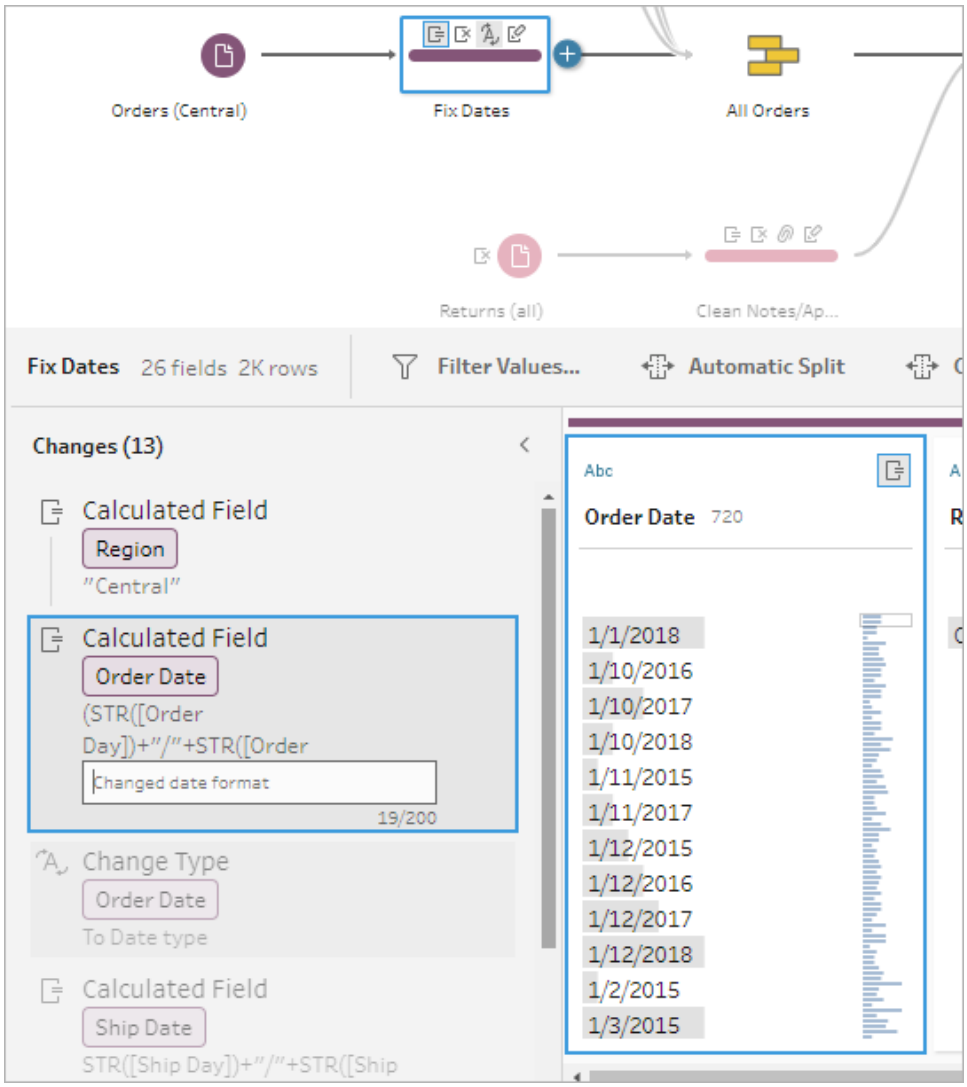
Add a description to a change entry

You can add a description to an entry in the Changes pane starting Tableau Prep Builder version 2019.1.1 and on the web.

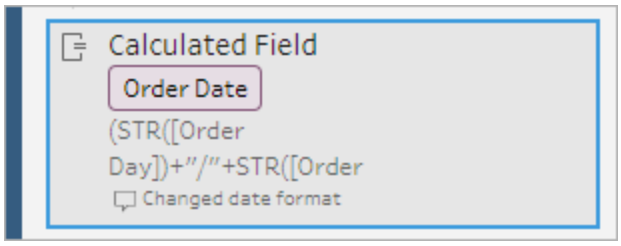
1. Select a step in the flow pane.
2. Open the **Changes** pane or **Changes** tab.
3. Right-click or Ctrl-click (MacOS) on an entry in the **Changes** pane and select **Add Description**.



4. Enter a description for the change action.



The description appears below the generated text for the change with a comment icon.



5. To edit or delete the description, right-click or Ctrl-click (MacOS) on the change item, and select **Edit Description** or **Delete Description**.

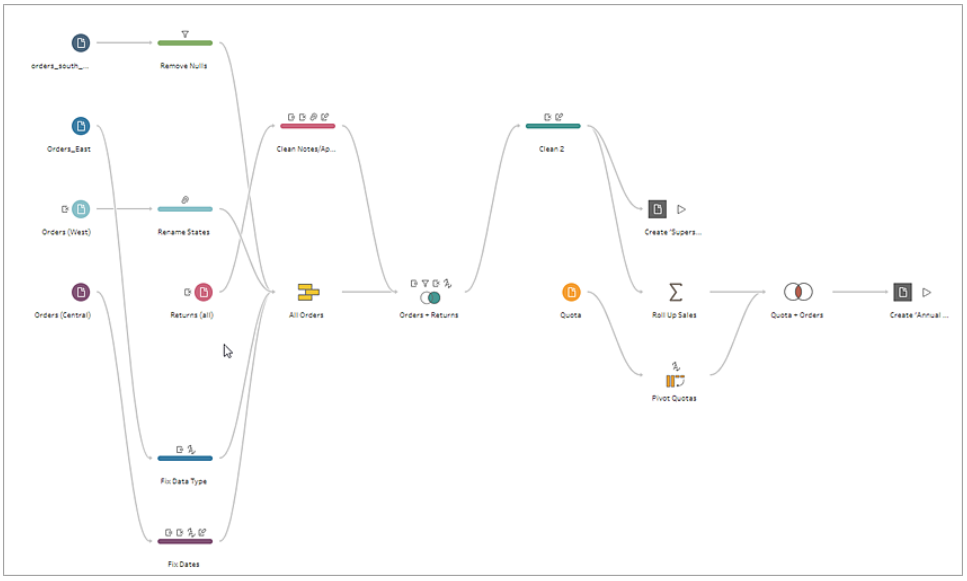
Reorganise the layout of your flow

Supported in Tableau Prep Builder version 2019.2.2 and later and on Tableau Server or Tableau Cloud starting in version 2020.4.

As you build a flow, Tableau Prep Builder uses a default layout. Each flow is laid out and processed from left to right, with Input steps beginning on the far left of the canvas and Output steps ending on the right side of the canvas. However, if you build large, complex flows, they can quickly become hard to follow.

You can clean up the layout of your flow by selecting and moving steps so the flow layout is organised in a way that makes sense to you. For example, you can fix crossed flow lines, move your flow steps to clean up extra white space or rearrange your flow steps to show a clear sequence of events.

For example. the following flow is confusing and hard to follow:



To clean up this flow select and drag steps up, down, left or right and drop them to a new location in the canvas. Flow steps can't be moved to a position that disrupts the left-to-right process flow. For example, you can't drag a union step that is positioned before a join step to a position that is after that join step in the flow.

When dragging flow steps to an allowed location, an orange box displays. If the location isn't allowed, no orange box displays and the steps return to their original location when you try to drop them.

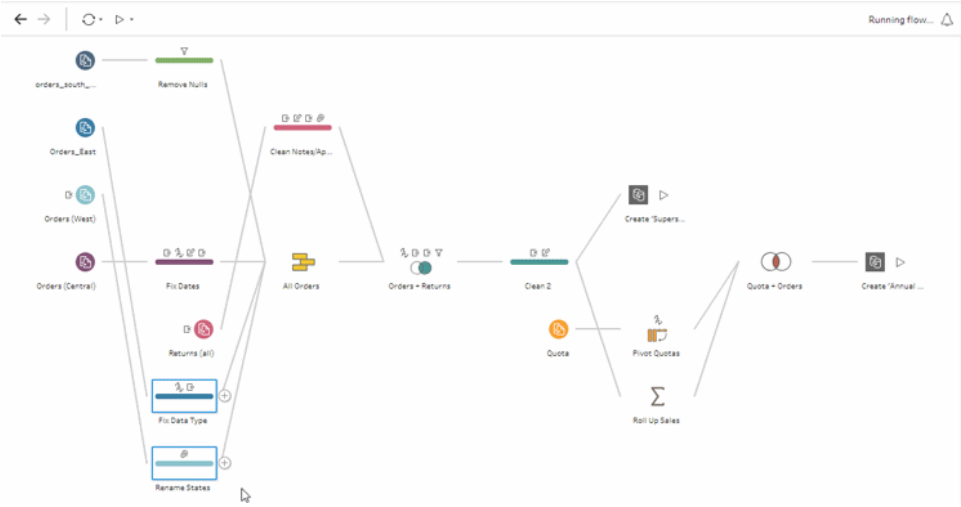
To move steps in your flow, do the following:

1. In the Flow pane, select the steps that you want to move. You can click on a specific step, drag to select multiple steps or Ctrl-click or Cmd-click (MacOS) to select steps that aren't next to each other.
2. Drag and drop the steps to the new location.

> **Note**: If you don't like the reorganisation moves that you make you can click **Undo** in the top menu to reverse them. However, if you perform cleaning actions in between moving steps, you may undo those actions as well. The **Undo** option reverses your actions in the order that you performed them.

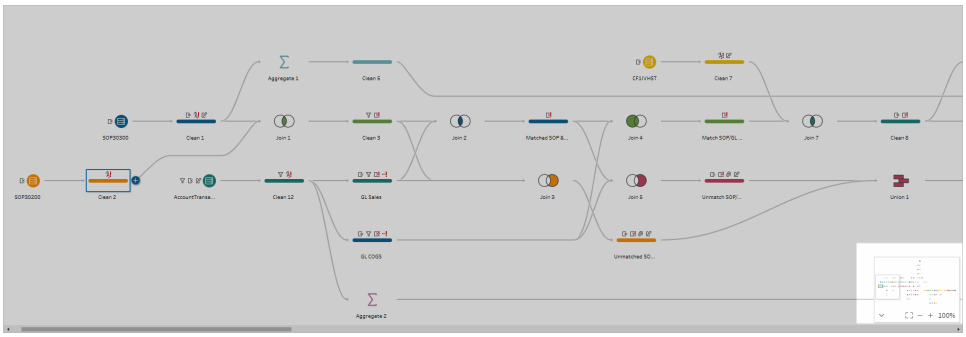
Watch "Reorganise flow steps" in action

The following example shows rearranging a flow using drag and drop.



Use the flow navigator tool

When working with large flows, scrolling back and forth to search for a particular area of the flow that you need to go back to can be difficult. The flow navigator tool makes this easier. The flow navigator is a miniature version of your flow that appears in the lower right corner of the canvas.



Click in any area of the graphic to jump to that area of your flow or use the following toolbar

options to navigate:

Toolbar option	Description
	Collapse the flow navigator graphic. In the collapsed state, you may only see the percentage indicator. Simply hover on this to expand the toolbar and click the up arrow to expand the graphic again.
	Expand the flow navigator graphic.
	Change the size of your flow to fit on your screen.
100%	Zoom in and out of your flow. You can click on the percentage indicator to restore the view to 100 percent.

Examine Your Data

Note: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

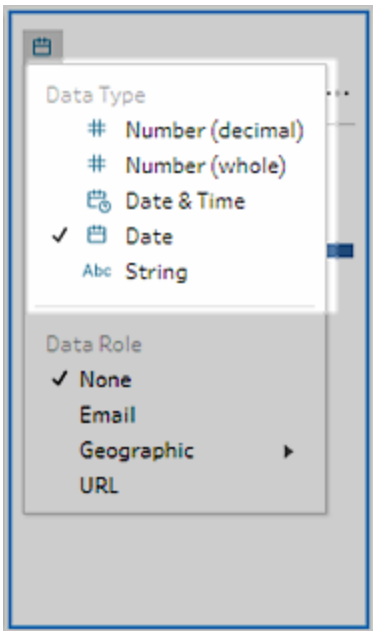
Use the options in this topic to get a good understanding about the composition of your data to better understand changes you need to make and the effect of the operations you include in the flow.

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.



You can change the data type in your Input step after connecting to data from the following data sources:

- Microsoft Excel
- Text files
- PDF files
- Box
- Dropbox
- Google Drive
- OneDrive

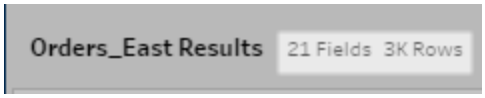
For all other data sources, add a cleaning step or other step type to make this change. To see a list of cleaning options available in the different step types, see **About cleaning operations** on page 227.

See size details about your data

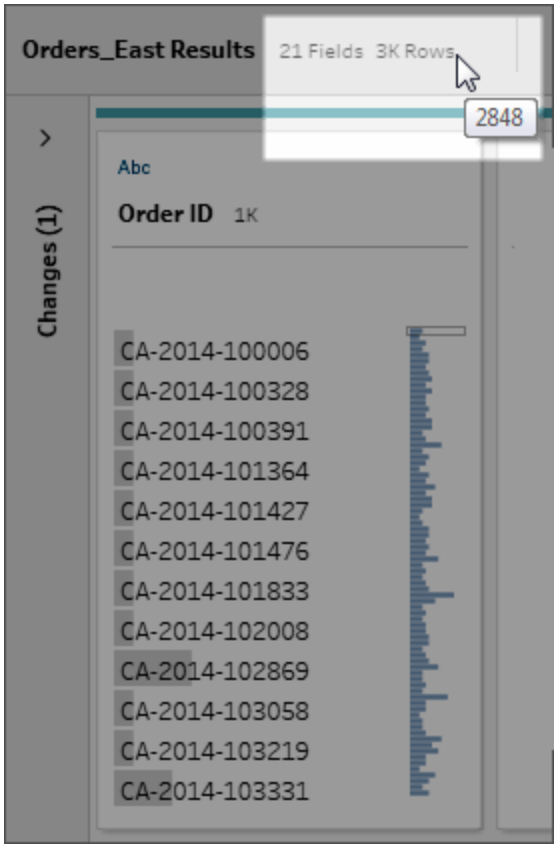
After you connect to your data, add a table to the flow, and then add a step. You can use the **Profile** pane to see the current state and structure of your data and spot nulls and outliers.

- **Number of fields and rows**: In the upper-left corner of the **Profile** pane you can find information that summarises 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 3000 rows in the data set.



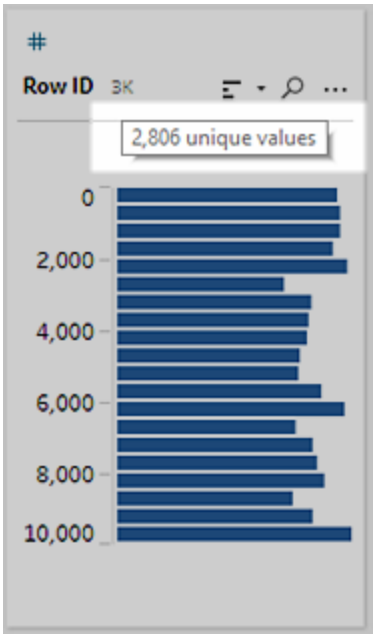
When you hover over the number of fields and rows, you can see the exact number of rows (in this example, 2848).



- **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 badge displays next to the size details in the **Profile** pane to indicate that this is a subset of your data set. You can modify the amount of data that you include in your flow. When creating or editing flows on the web, additional data limits apply. For more information, see **Set your data sample size** on page 121.

- **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 3,000 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.

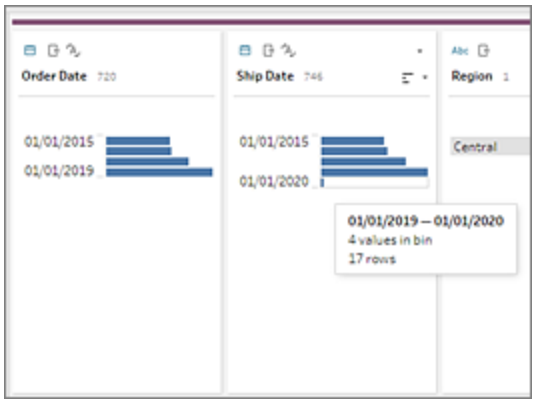


See the distribution of values or unique values

By default, Tableau Prep groups numerical, date and date & time 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 summarised or "binned" by year. Each bin represents a year from January of the beginning year to January of the following year and labelled

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.



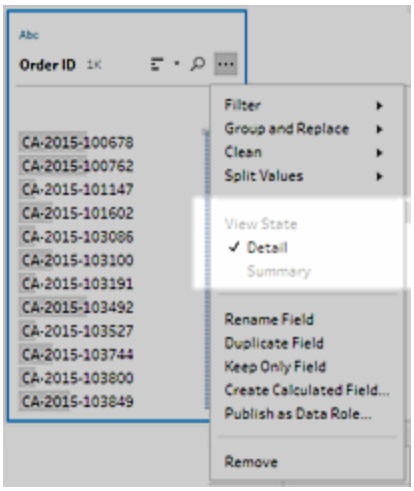
If a discrete (or categorical) data field contains many rows or has a distribution that is large enough that it can't be displayed in the field without scrolling, you can see a summarised distribution to the right of the field. You can click and scroll through the distribution to target specific values.

When your data contains numeric or date fields, you can toggle to display the detailed (discrete) version of the values or a summarised (continuous) version of the values. The summarised 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, Results pane or data grid, click the **More options** ··· menu for a numeric or date field.



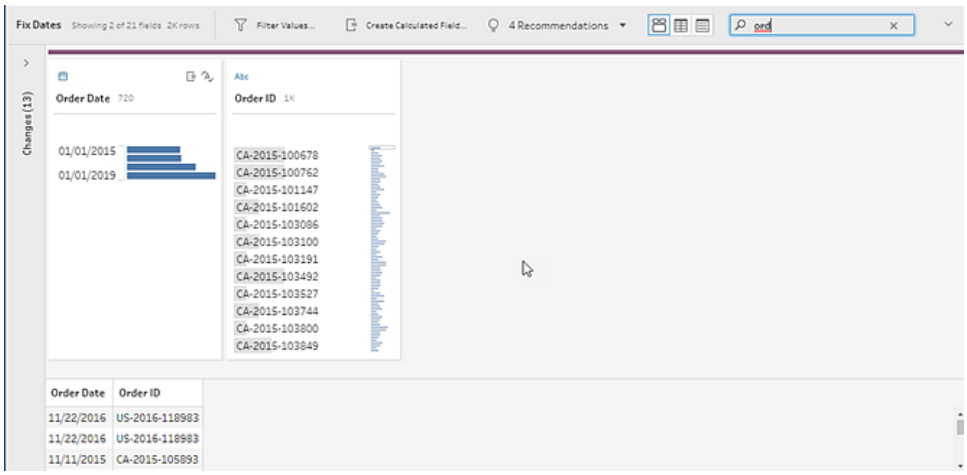
2. In the context menu, select **Detail** to see the detailed version of the values, or **Summary** to see the distributed version of the values.

Search for fields and values

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

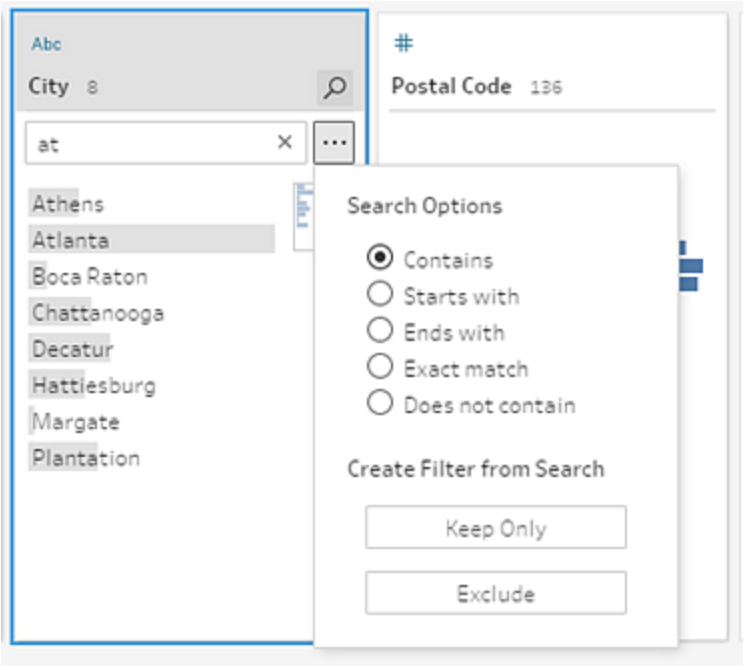
Starting in version 2021.1.1, when you search for fields, a new indicator will show telling you the number of fields found so you can better understand your search results. If no fields are found, additional messaging will show.

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

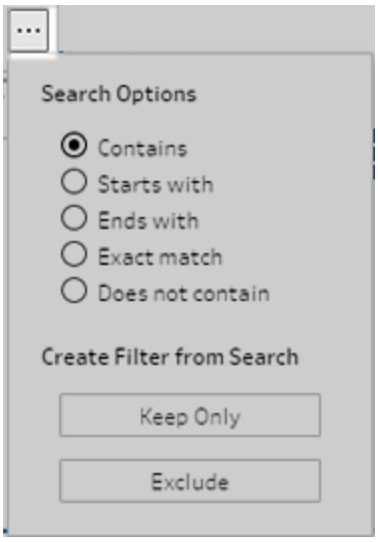


To search for a value in a field:

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**.

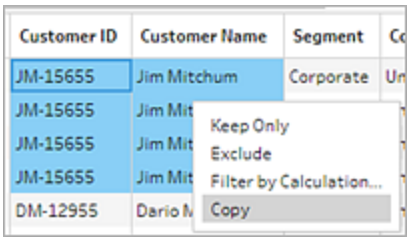
   In the **Flow** pane, a filter icon appears above affected steps.

Copy field values in the data grid

Supported in Tableau Prep Builder and Tableau Server version 2022.3 and later, and in Tableau Cloud version 2022.2 (August) and later.

Easily copy a selected set of values from the data grid and paste them into any document such as Microsoft Excel, Text (.csv) files, email and more. You can even copy and paste them into SQL editor to quickly run an SQL query.

1. In the data grid, select one or more field values to copy.
2. Right-click or cmd-click (MacOS) on the selected field values and select **Copy** from the menu. You can also use keyboard shortcuts Ctrl + C or cmd+C (MacOS) or select **Copy** from the **...** toolbar menu.



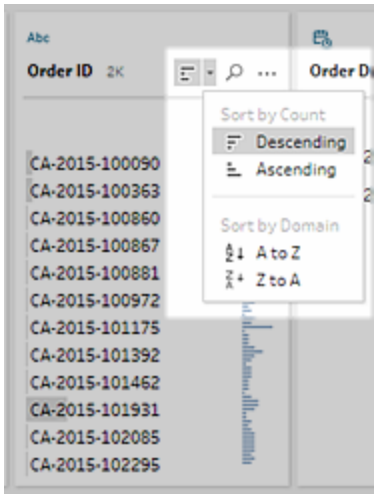
3. Paste the copied fields to your document or location.

> **Note**: **Edit** > **Copy** doesn't currently copy field values from the data grid.

Sort values and fields

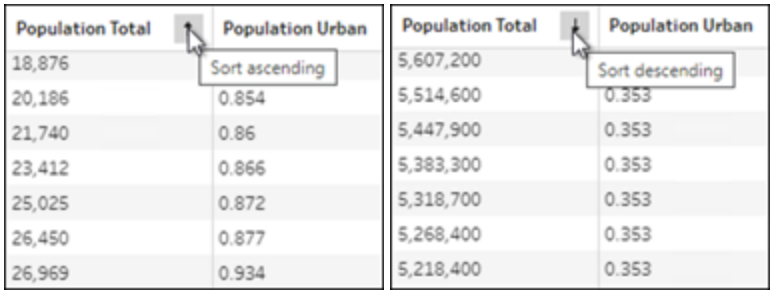
Profile pane

Sort options on a profile card 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.



Data grid

Sort the row-level detail for your fields in ascending or descending order.



Reorder fields

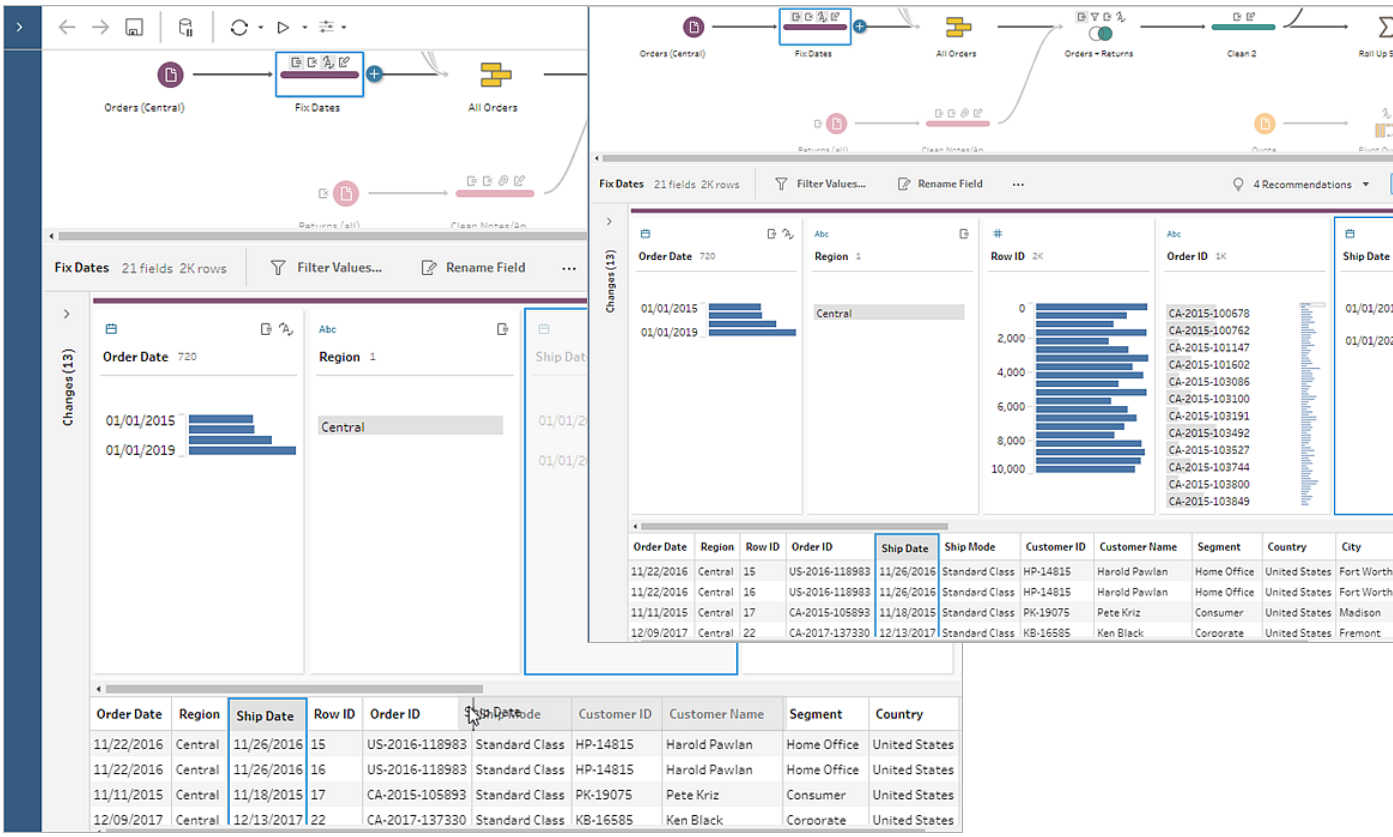
Changing the order of fields using the list view is supported in version 2022.2.1 and later.

You can change the order of fields from the Profile pane, Data grid or List view by dragging and dropping them into a new position.

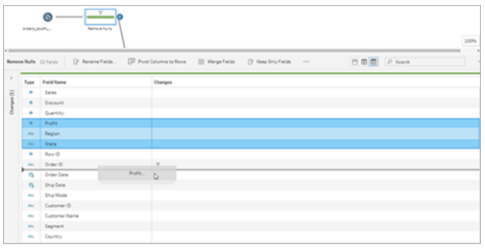
To rearrange the order of your fields:

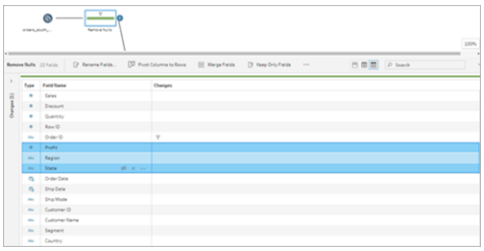
1. From the **Profile** pane, **Results** pane, **Data** grid or **List** view, select one or more profile cards or fields.
2. Drag the profile card or field until you see the black target line appear.
3. Drop the profile card or field into place.
   The Profile pane, data grid and list view are synced so the field will appear in the same order in all places. The new order for the fields is persistent across Tableau products when running and scheduling flows.

Data Grid reorder



List View reorder





Highlight fields and values in a flow

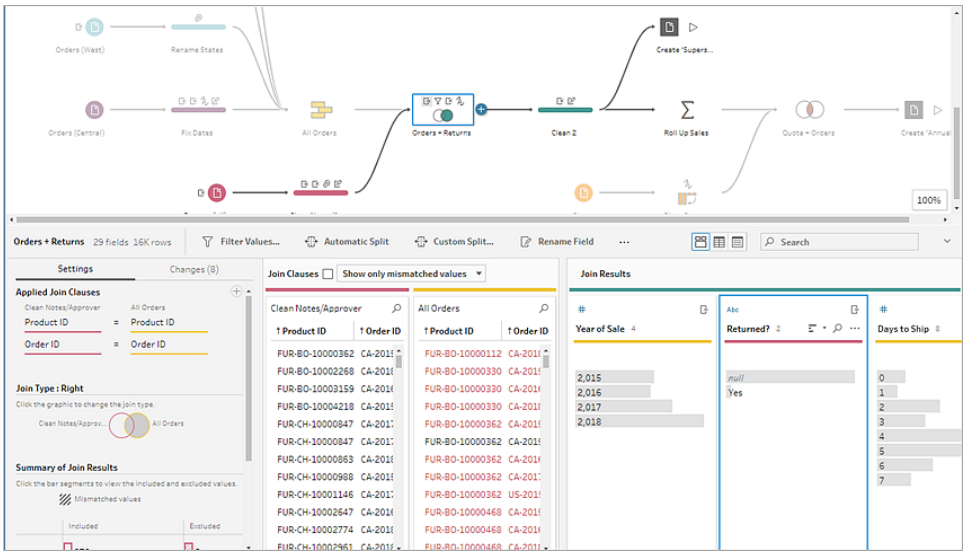
Tableau Prep makes it easy to find fields and values in your flow data. Trace where a field originated and where it is used throughout the flow in the flow pane, or click individual values in a profile card or in the data grid to highlight related or identical values.

Trace fields in a flow

In Tableau Prep, you can highlight everywhere a field is used in a flow, even where it originated to help you track down missing values or troubleshoot a flow when you aren't seeing the results you expect.

Click on a field in the **Profile** pane in a cleaning step or in the **Results** pane in any other step type and the flow pane will highlight the path where that field is used.

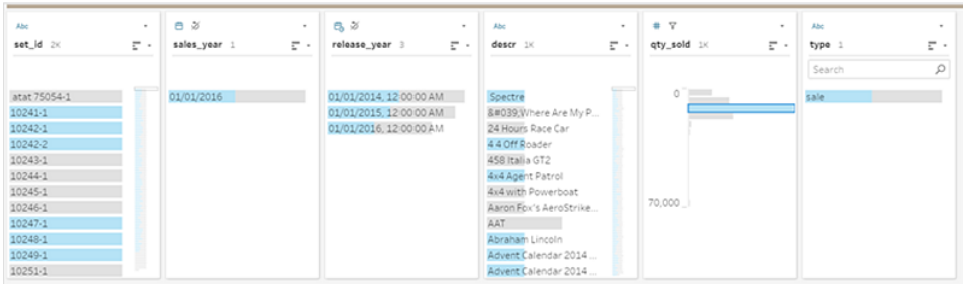
> **Note:** This option is not available for Input or Output step types.



See related values

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

For example, 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.



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.

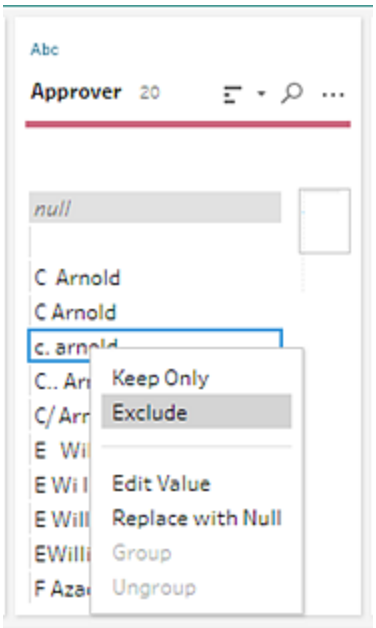
Type	Customer	Purchases	Date
Cash	Wei	5	08/18/2016
Cash	Jim	7	07/15/2016
Credit	Arnold	5	06/29/2016
Credit	Lee	1	08/07/2016
Cash	Maria	2	08/30/2016
Cash	Wendy	1	07/21/2016
Credit	Max	2	07/02/2016
Credit	Juan	1	05/10/2016
Cash	Isaac	4	06/28/2016
Credit	Philip	1	08/09/2016
Credit	Lane	5	05/04/2016

Filter Your Data

Note: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

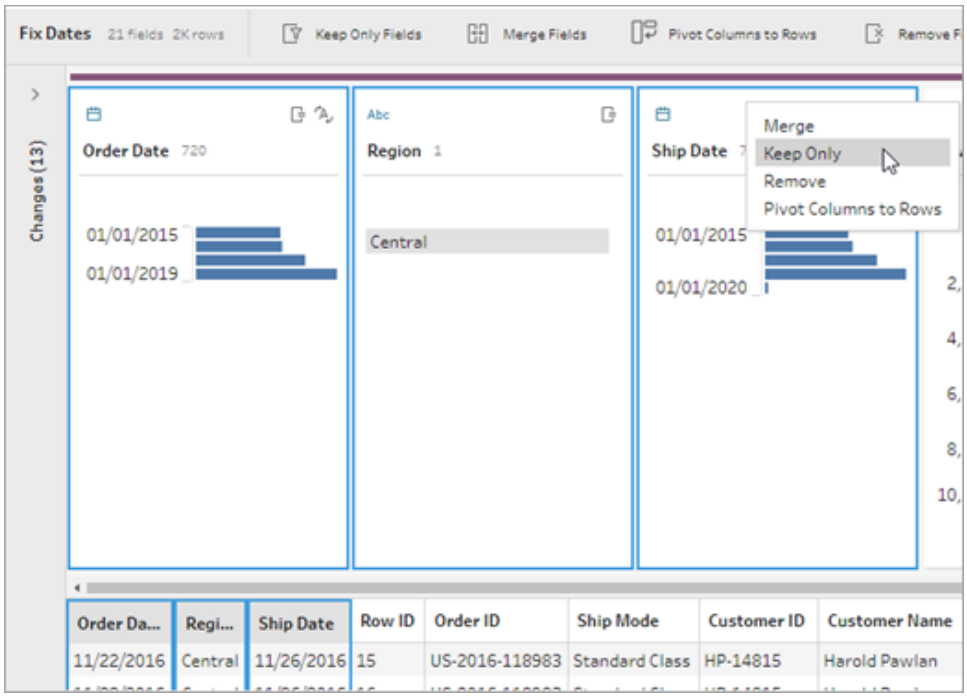
Tableau Prep provides various options that you can use to filter your data. For example, use **Keep Only** or **Exclude** to do one-click filtering on a specific value for a field in a profile card, data grid or results card or select from a variety of filter options for more complex filtering needs. You can also keep or remove entire fields.

Filter data at any step in the flow. If you want to simply change a specific value, you can select **Edit Value** to edit the value in-line or replace the value with Null. For more information about editing field values, see **Edit field values** on page 248.



Keep or remove fields

As you work with your data in your flow you might want to remove unwanted fields. In the Profile pane or the data grid in any cleaning or action step, select one or more fields and right-click or Ctrl-click (MacOS) and select **Remove** to remove the selected fields, or select **Keep Only** (Tableau Prep Builder version 2019.2.2 and later and on the web) to keep only the selected fields and remove all unselected fields.



Hide fields

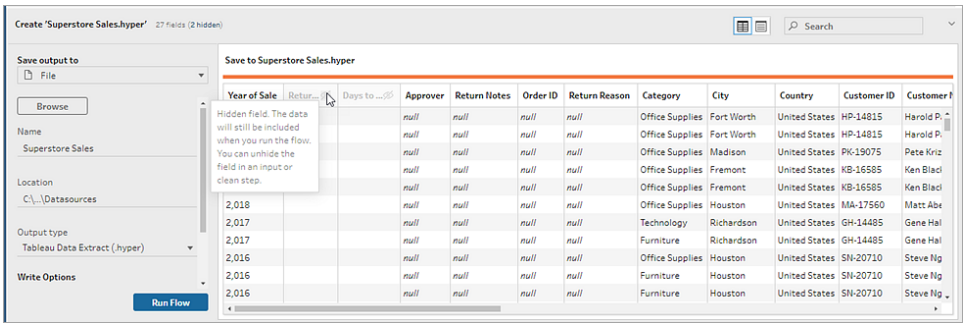
Supported in Tableau Prep Builder version 2021.1.4 and later and Tableau Server or Tableau Cloud starting in version 2021.1.

If you have fields in your flow that don't need cleaning, but you still want to include them in your flow, you can hide the fields instead of removing them. Data for those fields won't be loaded until you either unhide the fields or run your flow to generate your output.

When you hide fields, a new profile card called **Hidden Fields** is automatically added to the Profile pane, letting you easily unhide fields from the list as you need them.

You can include hidden fields in most operations, but joins, aggregations and pivots require the field to be unhidden to use it in one of these step types. If you hide the field after it has been used in one of these operations, the field will show as hidden and the operation won't be affected.

All hidden fields are tagged with an eye icon.



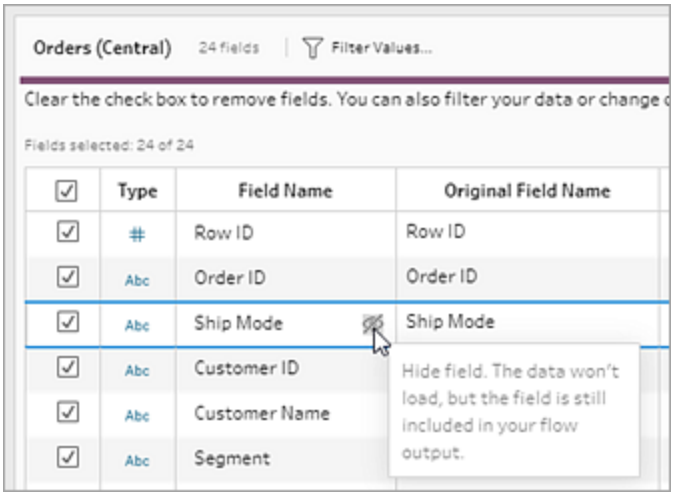
Hide and unhide fields

To hide or unhide fields, you must be in an Input step or in a Clean step. In the Clean step you can hide or unhide fields from the Profile pane, date grid and List view.

From the Input step

1. Connect to your data.
2. In the Input step select the field you want to hide or unhide.
3. Click the eye icon to hide or unhide the field.

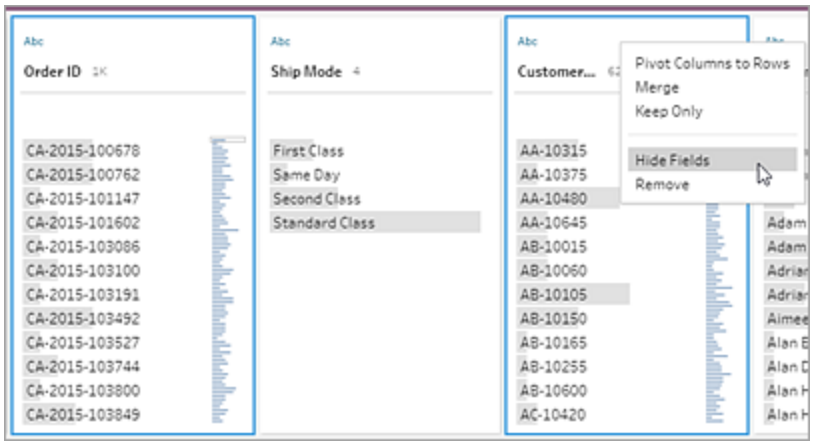
   Multi-selecting fields in the Input step is supported starting in version 2023.1.



From the Profile pane

1. Select the fields you want to hide.

2. Right-click, Ctrl-click (MacOS), from the **More options** ··· menu, or from the toolbar menu, select **Hide Field** or **Hide Fields**.



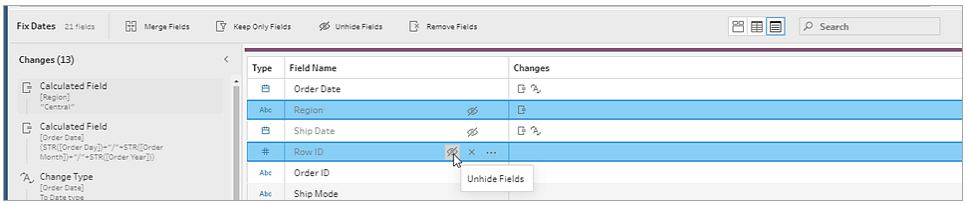
3. A new profile card is generated showing your hidden fields.

4. To unhide fields, in the **Hidden Fields** profile card, select one or more fields, and either click the eye icon, right-click, or Ctrl-click (MacOS) and select **Unhide Fields** from the menu.

From the List view

1. In a Clean step, on the toolbar, click the List view icon to change to the list view.
2. Select one or more fields to hide or unhide.

3. Click the eye icon to hide or unhide the fields.

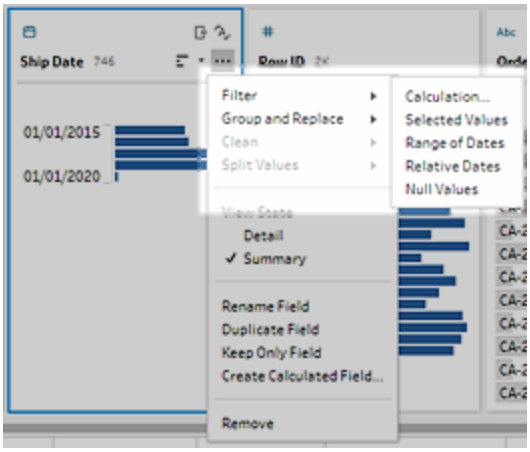


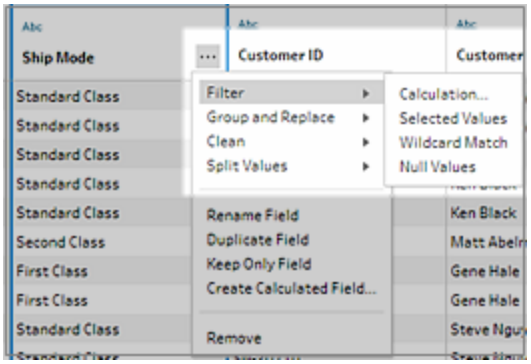
Filters available for each data type

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

Where are my filter options?

To see the different filter options available for your fields, on the profile card, in the data grid or in the results pane, click the **More options** ··· menu. To see the menu on the data grid, you must click the **Hide profile pane** button first, and then click **More options** ··· .

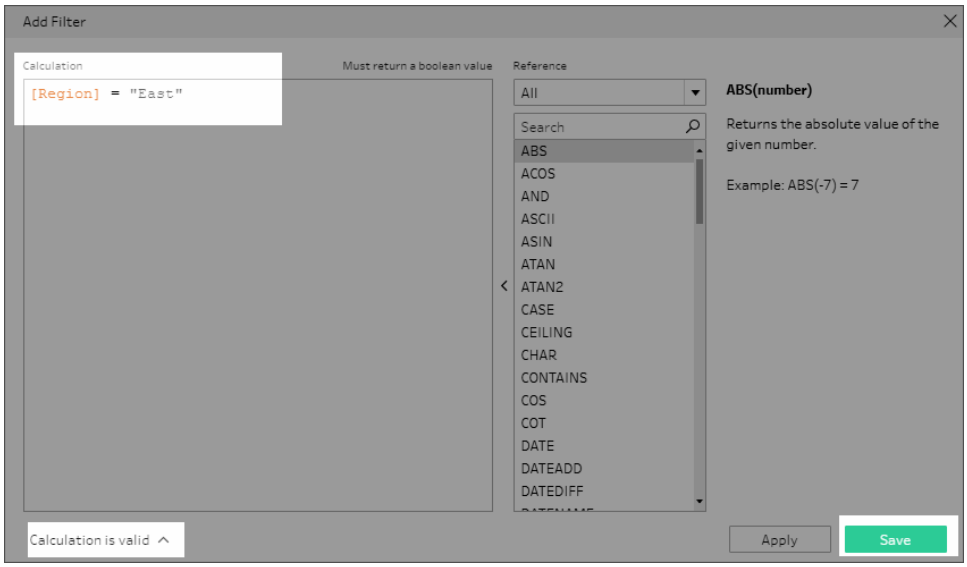




Calculation filter

When you select **Calculation**, the **Add Filter** dialog box opens. Enter the calculation, verify that it's valid, and click **Save**. Starting in version 2021.4.1, you can also include parameters in calculation filters. For more information, see **Apply user parameters to filter calculations** on page 219.

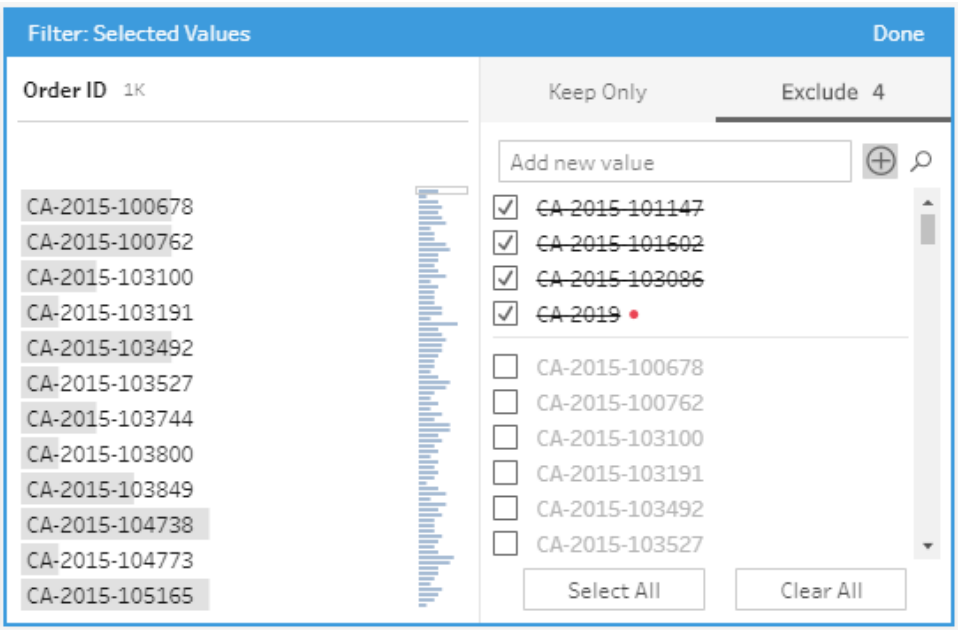
> **Note:** In the Input step, this is the only type of filter that is available. All other filter types are available in the profile cards, data grid or results pane.



Selected Values filter

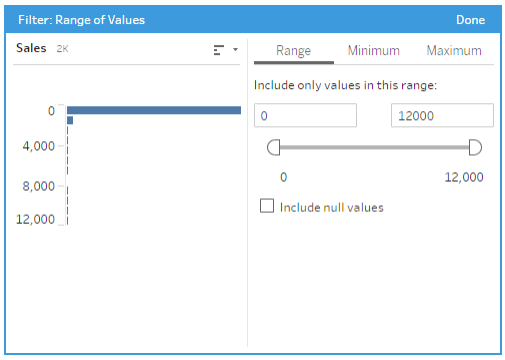
In Tableau Prep Builder version 2019.2.3 and later and on the web, you can use the **Selected Values** filter to pick and choose the values that you want to keep or exclude for a field, even values that aren't in your sample. In the right pane, click the **Keep Only** or **Exclude** tab to select your action, then enter search terms to search for values or click **Add a value** to add values that are in your data set but aren't included in your sample. Click **Done** to apply your filter.

> **Note:** This filter option isn't available for **Aggregation** or **Pivot** step types.



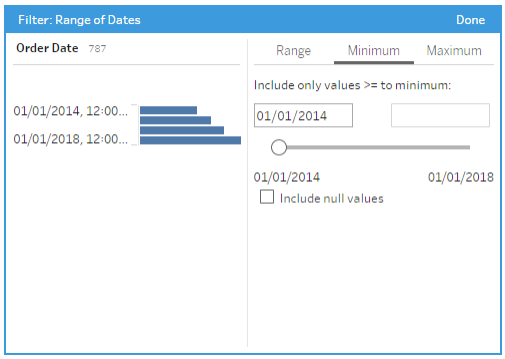
Range of Values filter

Filter out values that fall within a specific range. When you select **Range of Values**, you can specify a range or set minimum or maximum values.



Range of Dates filter

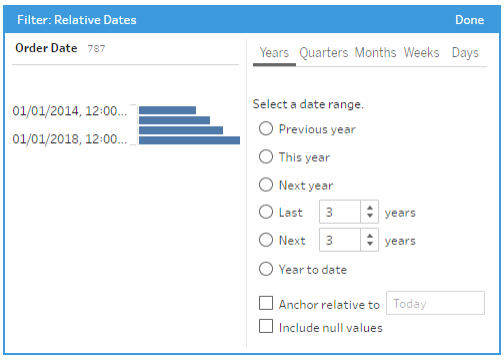
Filter out values that fall within a specific date range. When you select **Range of Dates**, you can specify a range of dates or set a minimum or maximum date.



Relative Date filter

Use the **Relative Dates** filter to specify the exact range of years, quarters, months, weeks or days that you want to see in your data. You can also configure an anchor relative to a specific date, and include null values.

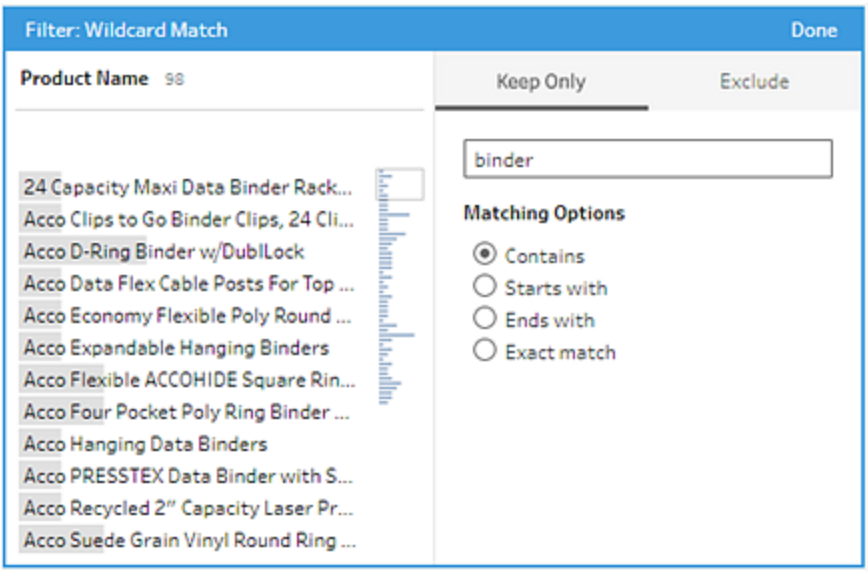
> **Note**: “Last” date periods include the complete current unit of time, even if some dates haven't occurred yet. For example, if you select the last month and the current date is 7th January, Tableau will display dates for 1st January to 31st January.



Wildcard Match filter

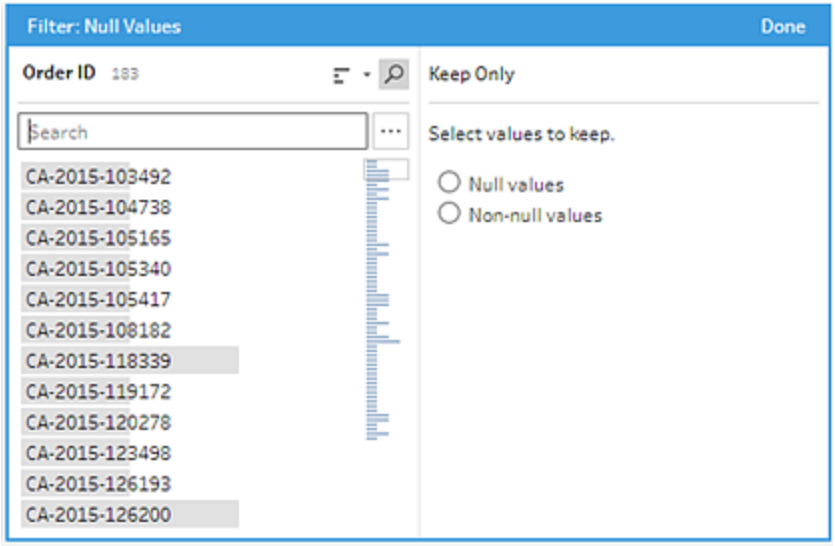
When you select **Wildcard Match**, you can filter the field values to keep or exclude values that match a pattern. In the filter editor, select the **Keep Only** or **Exclude** tab, enter a value to match and then set the **Matching Options** criteria to return the values you are looking for.

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.



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.



Remove Duplicate Rows

Supported in Tableau Prep Builder version 2024.1 and later and on the web in Tableau Cloud.

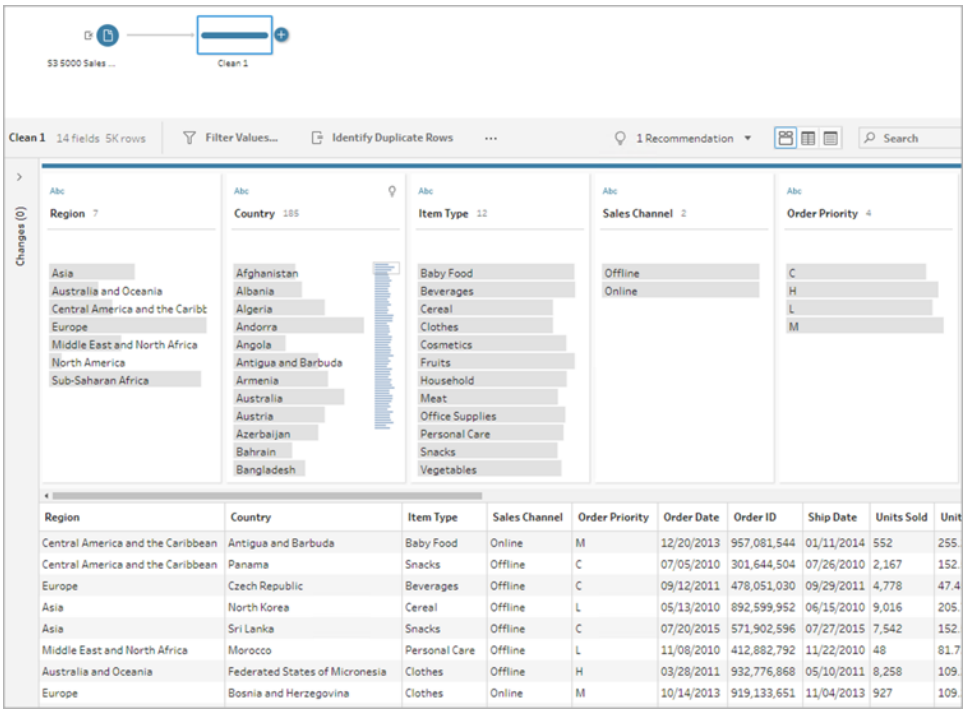
Duplicate data can have an impact on your data quality, bias prediction results and waste storage space. Removing duplicates ensures that each value is unique, providing an accurate representation of your data. The first step to removing duplicates is to identify the duplicate rows in your data. After you've identified duplicates, you can then remove them from the workflow.

Identify and remove duplicates

Tableau Prep identifies a duplicate as any row that has at least one other row with the same values.

To identify and remove duplicate rows.

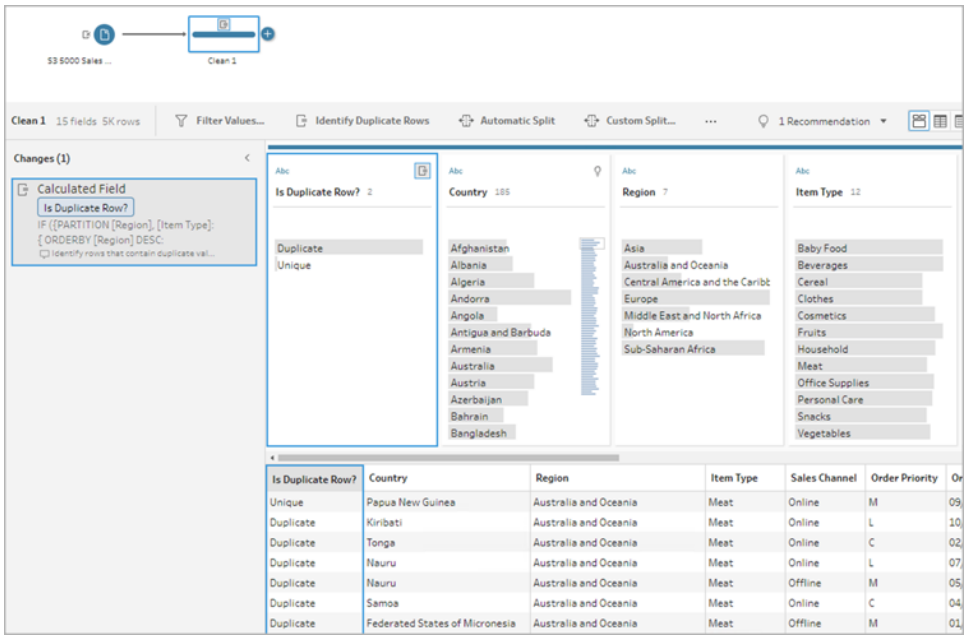
1. (Optional) For Excel and text files, you can click on the Input step and add a Source Row Number field to sort rows by the original data source. For more information, see Add the Source Row Number field to your flow and **Sort order for the calculated field** on page 189.
2. Click on any step other than an Input or Output step.



3. Identify duplicate rows.
   - To identify duplicate rows across all fields, from the toolbar, click **Identify Duplicate Rows**.
   - To identify duplicate rows across specific fields, select one or more fields, then click **Identify Duplicate Rows**. Optionally, in the profile pane, you can click the **More options** ··· menu from the selected field and select **Identify Duplicate Rows**.

A calculated field is created and listed in the Changes pane. The calculated field provides a description and the names of the fields that the duplicate rows span. The Data Grid shows which rows are unique and which rows are duplicates.

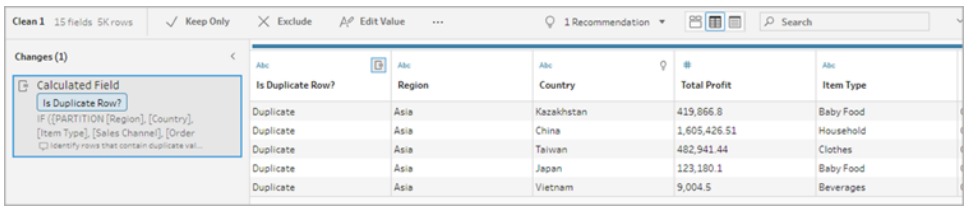
The following image shows the results of selecting the two fields *Region* and *Item Type* to identify duplicate rows across only those selected fields.



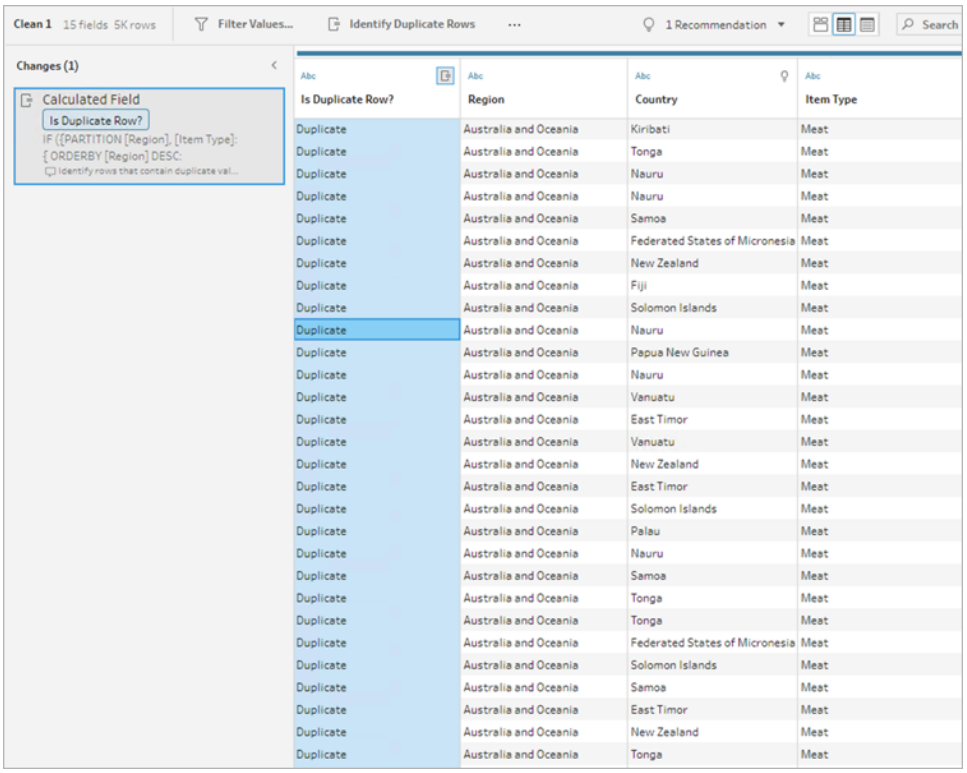
If a Source Row Number is available, it's used to sort the order of the rows. Otherwise, rows are sorted based on the first field in the table schema. To change the values in the calculated field, you can directly edit the calculation or write your own calculation. See **Sort order for the calculated field** on page 189.

4. To evaluate duplicate rows, click on **Duplicate** or **Unique** in the Is Duplicate Row? field or in the Data Grid.

   The following image shows duplicate rows across all fields.



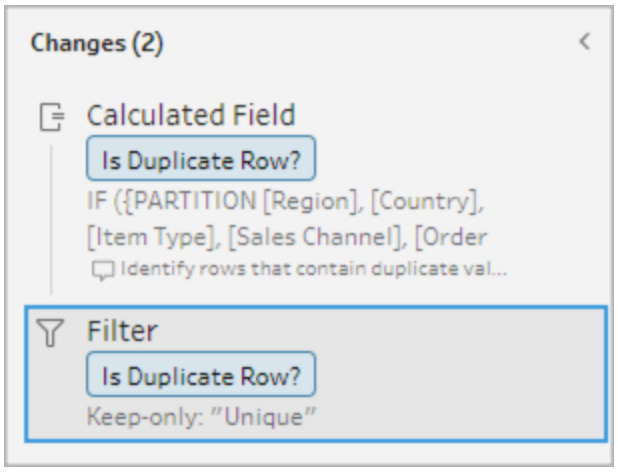
   The following image shows duplicate rows across only the fields *Region* and *Item Type*.



5. To filter and keep only the unique rows, click **Unique** in the Data grid or in the Is Duplicate Row? field, and select **Keep Only** from the toolbar. Optionally, you can click **Unique** in the Data grid or in the Is Duplicate Row? field, then right-click to select **Keep Only**.

   You can also select **Duplicate** and then select **Exclude** to filter data for duplicates. This produces the same results as selecting Unique and Keep Only and doesn't affect which rows are excluded or kept.

   A filter is created in the changes pane showing that only unique rows are kept and duplicate rows are excluded.



6. From the Is Duplicate Row? calculated field, click the **More options** ··· menu and select **Remove**.

   The duplicate rows are removed when you run the flow and create an output.

   > **Note**: Don't remove the Is Duplicate Row? calculated field if you'll be adjusting for sampled data.

Duplicates in sample data

Duplicate rows may not appear in sampled data and could introduce a bias in the model.

To address duplicate rows sampled data:

1. Don't remove the **Is Duplicate Row?** calculated field if you'll be adjusting for sampled data. This gives you the option to change the sample size or adjust the sample data in the input step.
2. Adjust the sample size to load as much data as possible for row selection. See Set your data sample size.
3. Duplicates may not appear in sampled data depending on the size of the sample. Using the **Keep Only** option for unique rows allows you to remove duplicate rows, even if they aren't visible in the sampled data.

Sort order for the calculated field

By default, if multiple rows contain the same value, they're sorted by the first field or, if available, the Source Row Number. The fields are used to identify the first row as *Unique* and the remaining as *Duplicate*. Changing the sort order of fields allows you to specify which duplicate row you want

to identify as Unique. You can make changes to the order by editing the calculation and changing the `ORDERBY` field.

For example:

```
IF ({PARTITION [Field1], [Field2], [Field3]: { ORDERBY [FieldName]
ASC: ROW_NUMBER() } } = 1) THEN 'Unique' ELSE 'Duplicate' END
```

For Excel and text files, you can sort by the original data source by adding a Source Row Number field.

For example:

```
IF ({PARTITION [Source Row Number], [Field1], [Field2], [Field3]: {
ORDERBY [Source Row Number] DESC: ROW_NUMBER() } } = 1) THEN 'Unique'
ELSE 'Duplicate' END
```

For more information, see Add the Source Row Number field to your flow.

Write your own filter calculation to find and remove duplicates

You can write your own filter calculation across multiple fields to find and remove duplicates.

The following filter calculation will only return results that return True. The calculation returns either True or False based on the fields used to PARTITION

```
{PARTITION [Field1], [Field2], [Field3]: { ORDERBY [Field1] DESC:
ROW_NUMBER() } } = 1
```

Use `CASE` or `IF` for identifying duplicate calculations

You can use the `IF` or `CASE` functions in the calculation editor. For example:

```
CASE {PARTITION [Field1], [Field2], [Field3] : { ORDERBY [[Field3]]:
ROW_NUMBER() } } = 1 WHEN TRUE THEN 'UNIQUE' ELSE 'DUPLICATE' END
```

```
IF ({PARTITION [[Field1]], [[Field2]], [[Field3]]: { ORDERBY
[[Field3]] DESC: ROW_NUMBER() } } = 1) THEN 'Unique' ELSE 'Duplicate'
END
```

Use Data Roles to Validate your Data

Note: Data source owners and Tableau administrators can add synonyms for specific data field names and values for Ask Data. For information about using data roles for Ask

Data, see Add Synonyms for Ask Data in the Tableau Desktop help.

Note: Tableau's Ask Data feature was retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2.

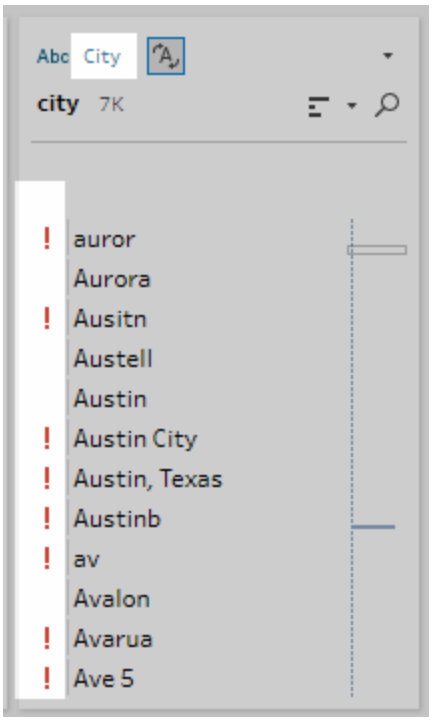
Use data roles to quickly identify whether the values in a field are valid or not. Tableau Prep delivers a standard set of data roles that you can select from or you can create your own using the unique field values in your data set.

When you assign a data role, Tableau Prep compares the standard values defined for the data role with the values in your field. Any values that don't match are marked with a red exclamation mark. You can filter your field to view only the valid or invalid values and take the appropriate actions to fix them. Once you've assigned a data role to your fields, you can use the **Group Values** option to group and match invalid values to valid ones based on spelling and pronunciation.

Note: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see Tableau Prep on the Web in the Tableau Server help.

Assign standard data roles to your data

Assign data roles provided by Tableau Prep to your field the same way you assign a data type. The data role identifies what your data values represent so Tableau Prep can automatically validate values and highlight ones that aren't valid for that role.



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 to identify values that don't match.

> **Note**: Each field is analysed 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 Builder provides 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 (US)
    - CBSA/MSA
    - City
    - Congressional District (US)
    - Country/Region

- County
- NUTS Europe
- State/Province
- Zip code/Postcode

Tip: In Tableau Prep Builder version 2019.1.4 and later and on the web, if you assign a geographic role to a field, you can also use that data role to match and group values with the standard value defined by your data role. For more information about grouping values using data roles, see **Clean and Shape Data** on page 227.

To assign a data role to a field, do the following:

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

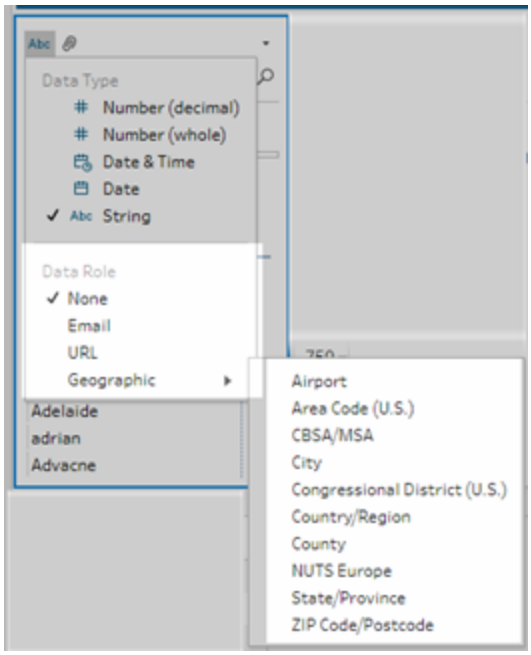
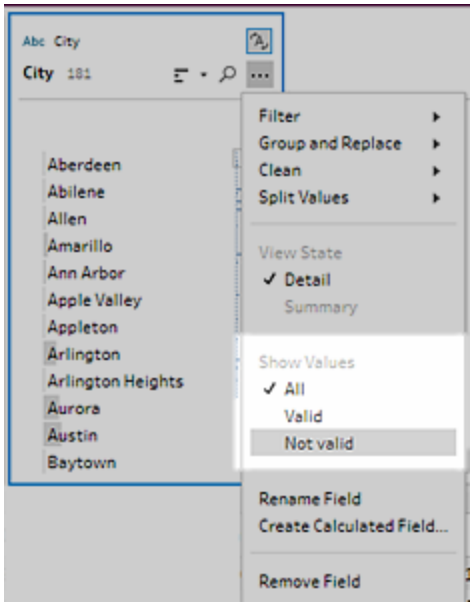
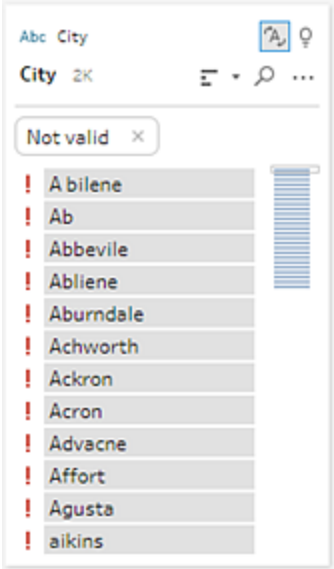


Tableau Prep compares the field's data values to known domain values or patterns (for email or URL) for the data role you select and marks any values that don't match with a red exclamation mark.

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 **More options** ··· menu for the field to correct any values that aren't valid. For more information about how to clean your field values see **About cleaning operations** on page 227.

Create customised data roles

Important: Custom data roles will be retired in Tableau Cloud in February 2024 and in Tableau Server version 2024.2.

Starting in Tableau Prep Builder version 2019.3.1 and on the web, you can create your own customised data roles using the field values in your data sets to create a standard set of values that you or others can then use to validate fields when cleaning data. Select the field that you want to use, apply any cleaning operations to it if needed, then, publish it to Tableau Server or Tableau Cloud to use it in your flow or share your data roles with others.

If creating customised data roles when editing flows on the web, you can publish the customised data role directly to the server you are signed into.

Requirements

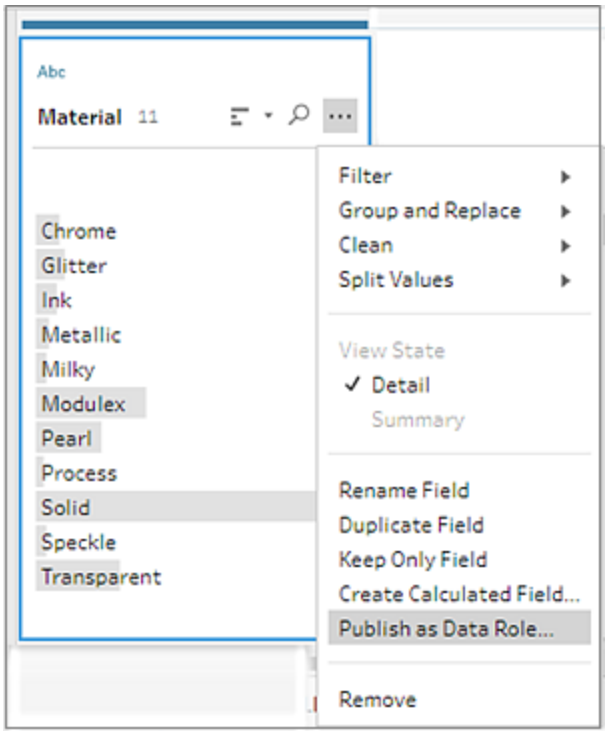
- You can create custom data roles from single fields in your data set. Creating custom data roles from a combination of fields isn't supported.
- Publishing data roles to projects with locked permissions isn't supported.

- You can create custom data roles only for fields assigned to a data type of **String** and **Number (whole)**.
- When you create a customised data role, Tableau Prep creates an output step in your flow that is specific to publishing the data role.
- Publishing custom data roles to multiple sites in the same flow isn't supported. If you publish the flow, you must publish the custom data role to the same site or server where the flow is published.
- Custom data roles are specific to the site, server and project where you publish them. All users with permissions to the location can use the customised data role, but must be signed into the site or server to select it or apply it. Customised data roles are assigned the default permission for the **All Users** group for new projects instead of **None**.
- Custom data roles aren't version specific. When applying a custom data role, the most current version is applied.
- Once published to Tableau Server or Tableau Cloud user with access to the site, server and project can view all data roles in that location.
    - Users with appropriate permissions can move, delete or edit permissions for the data roles.
    - The permissions you can set and actions you can take on a custom data role are similar to what you can do with a flow. For more information, see Manage a Flow and Permission capabilities in the Tableau Server help.
- To edit a data role, you must make your changes in Tableau Prep Builder or in the flow on the web, then republish the data role using the same name to overwrite it. This process is similar to editing a published data source.

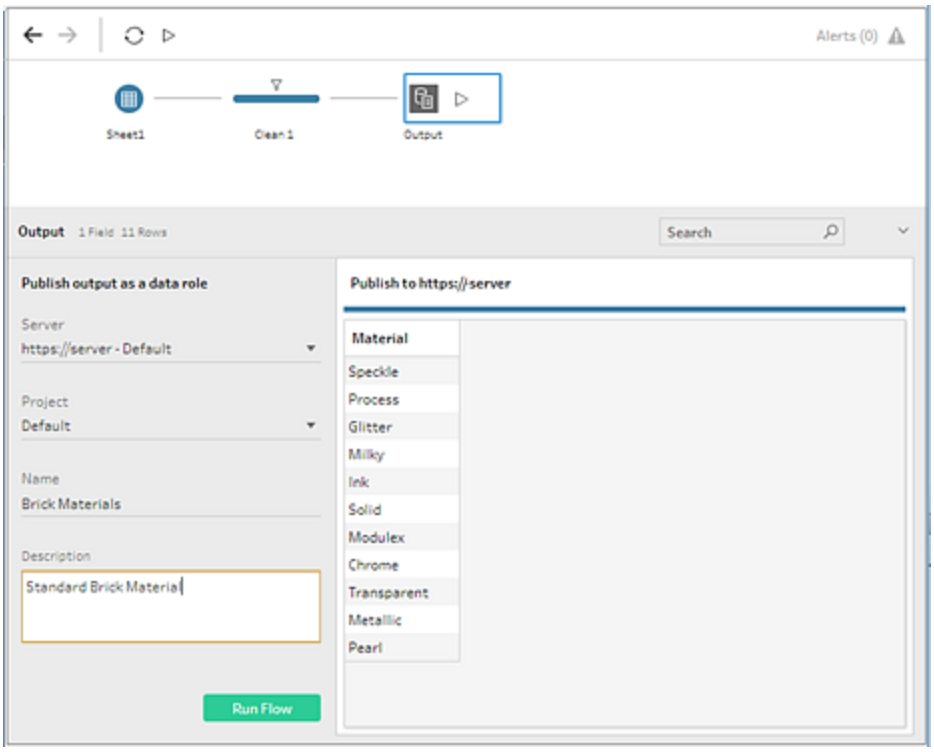
Create a custom data role

1. In the **Profile** pane, data grid or Results pane select the field you want to use to create a custom data role.

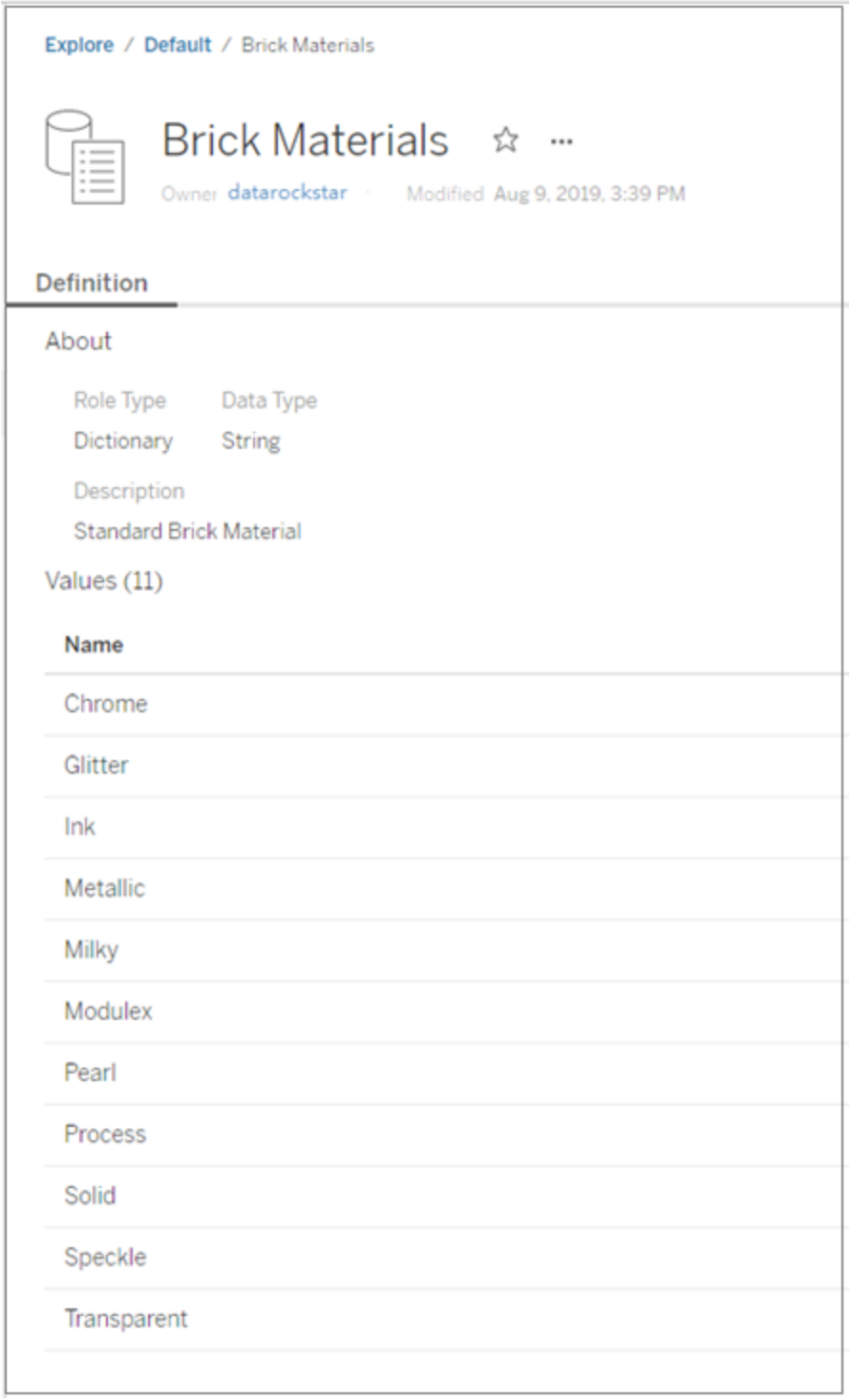
2. Click **More options** ··· for the field, and select **Publish as Data Role**.



3. Select the server and project where you want to publish the data role.



4. Click **Run Flow** to create the data role. After the publishing process completes successfully, you can view your data role in Tableau Server or Tableau Cloud. Processing the data role can take some time based on the load on your Tableau Server or Tableau Cloud site. If your data role isn't available right away, wait a few minutes, then try selecting it again.


Explore / Default / Brick Materials
Brick Materials
Owner datarockstar
Modified Aug 9, 2019, 3:39 PM
Definition
About
Role Type
Data Type
Dictionary
String
Description
Standard Brick Material
Values (11)
Name
Chrome
Glitter
Ink
Metallic
Milky
Modulex
Pearl
Process
Solid
Speckle
Transparent

Apply a custom data role

1. In the Profile pane, Results pane or data grid, click the data type for the field where you want to apply the custom data role.
2. Select **Customise** then select the data role that you want to apply to the field.

   Important: In Tableau Prep Builder, make sure you are signed into the site or server where the data role was published or you won't see this option.

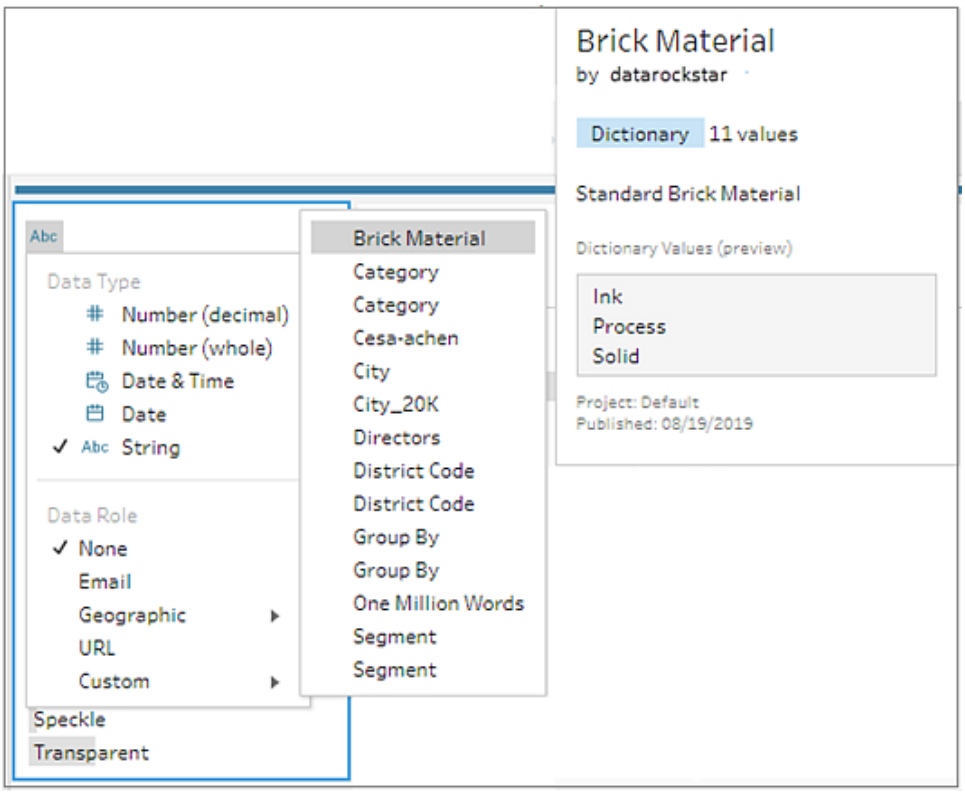
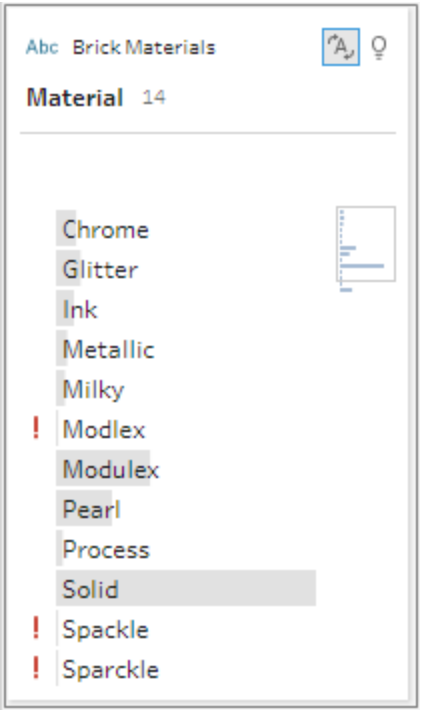
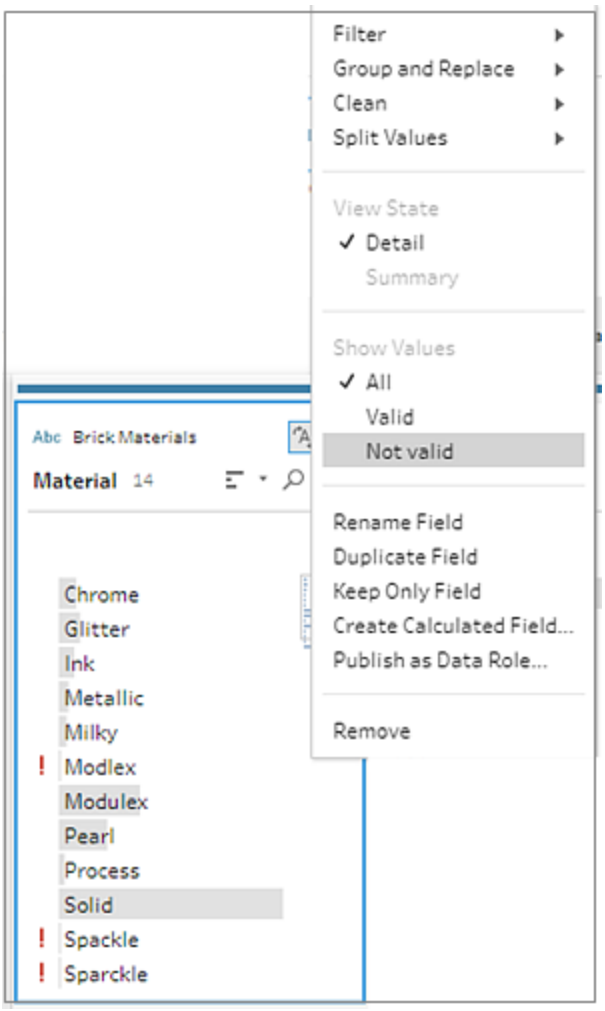


Tableau Prep compares the field's data values to known domain values for the data role you select and marks any values that don't match with a red exclamation mark.



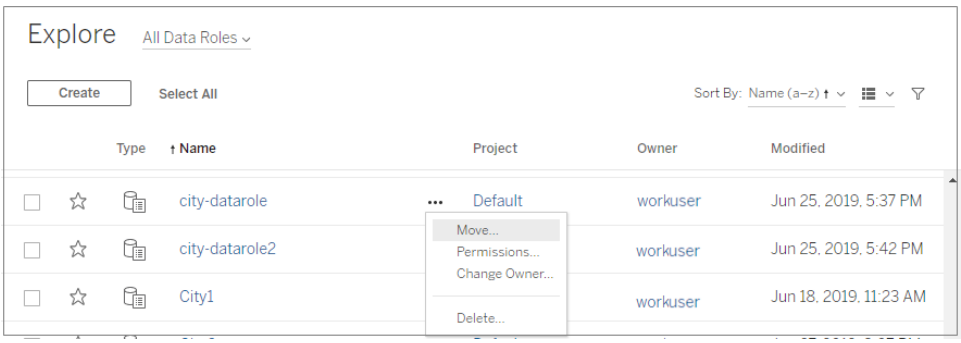
3. Click the drop-down arrow for the field and from the **Show Values** section select an option to show all values or only values that are valid or not valid for the data role.



4. Use the cleaning options on the **More options** ··· menu for the field to correct any values that aren't valid. For more information about how to clean your field values see **About cleaning operations** on page 227.

View and manage custom data roles

You can view and manage your published custom data roles on Tableau Server and Tableau Cloud. You can view all custom data roles published to your site or server. Click **More actions** ··· for a selected data role to move it to a different project, change permissions or delete it.



Group similar values by data role

> **Note**: In Tableau Prep Builder version 2019.1.4 and 2019.2.1 this option was labelled **Data Role Matches**.

If you assign a geographic data role to a field you can use the values in the data role to group and match values in your data field based on spelling and pronunciation to standardise them. You can use either **Spelling** or **Spelling + Pronunciation** to group and match invalid values to valid ones.

These options uses the standard value defined by the data role. If the standard value isn't in your data set sample, Tableau Prep adds it automatically and marks the value as not in the original data set. For more information about assigning data roles to fields, see **Assign standard data roles to your data** on page 191.

To use data roles to group values, complete the following steps.

1. In the Profile pane, Results pane or data grid, click the data type for the field.
2. Select one of the following data roles for the field:
   - Airport
   - City
   - Country/Region
   - County
   - State/Province

   Starting in Tableau Prep Builder version 2019.3.2 and on the web, you can also select from your customised data roles.

Standard data roles (version 2019.1.4 and later)	Custom data roles (version 2019.3.2 and later)

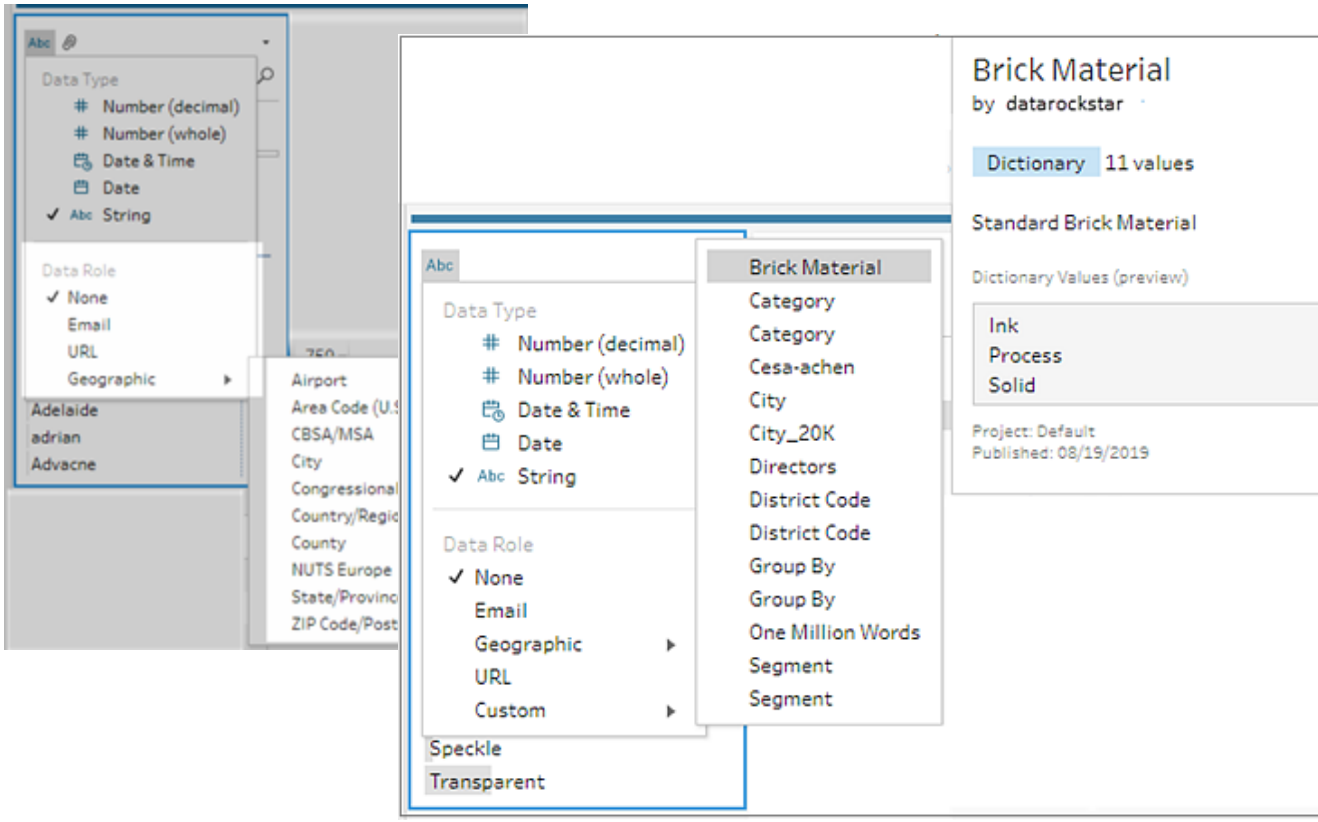
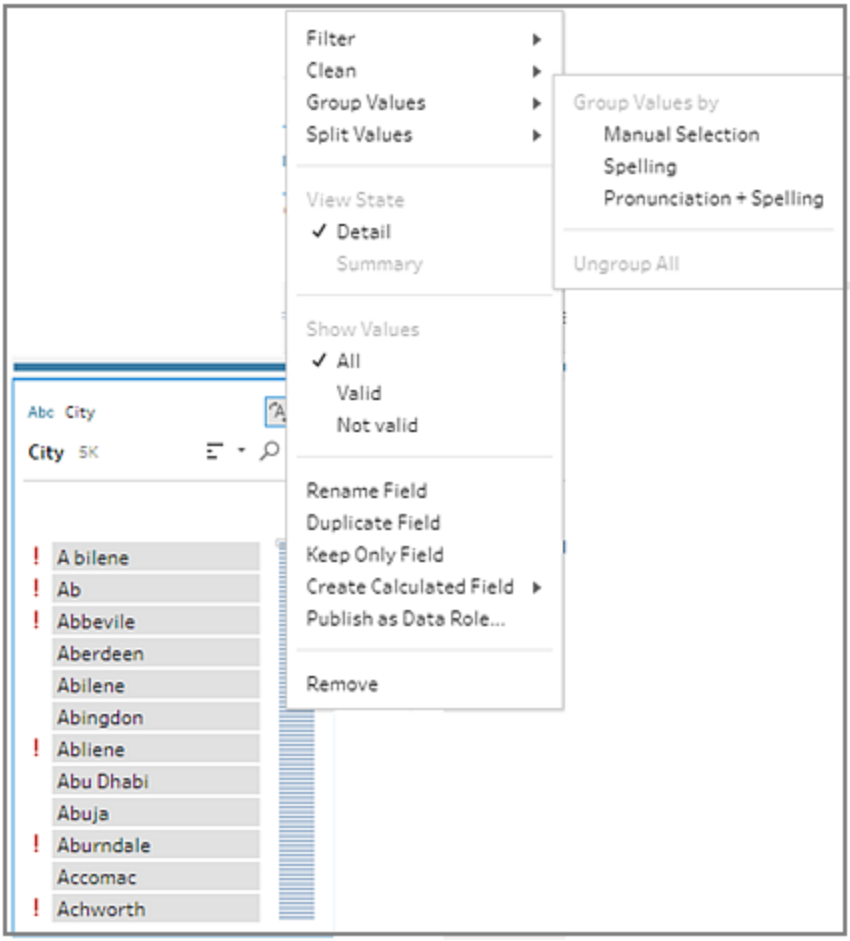


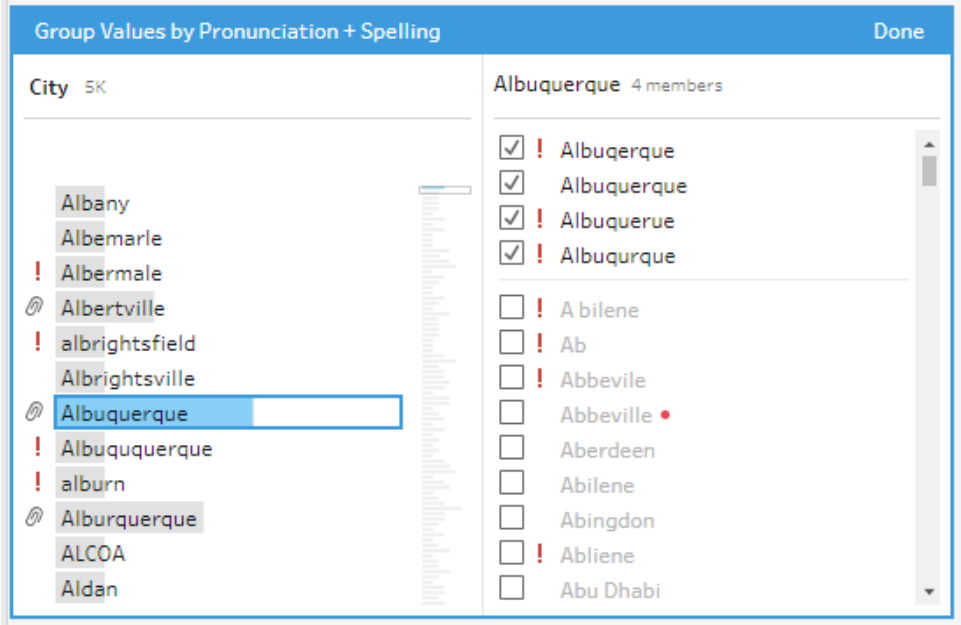
Tableau Prep compares the field's data values to known domain values for the data role you select and marks any values that don't match with a red exclamation mark.

3. Click **More options** ··· , select **Group Values** (**Group and Replace** in previous versions), then select one of the following options:
   - **Spelling**: Matches invalid values to the closest valid values that differ by adding, removing or substituting characters.
   - **Pronunciation + Spelling**: Matches invalid values to the most similar valid value based on spelling and pronunciation.



You can also click on the **Recommendations** icon on the field to apply the recommendation to group and replace the invalid values with valid ones. This option uses the **Pronunciation + Spelling** Group Values option.

Tableau Prep compares the values by spelling or spelling and pronunciation and then groups similar values under the standardised value for the data role. If the standardised value isn't in your data set, the value is added and marked with a red dot.



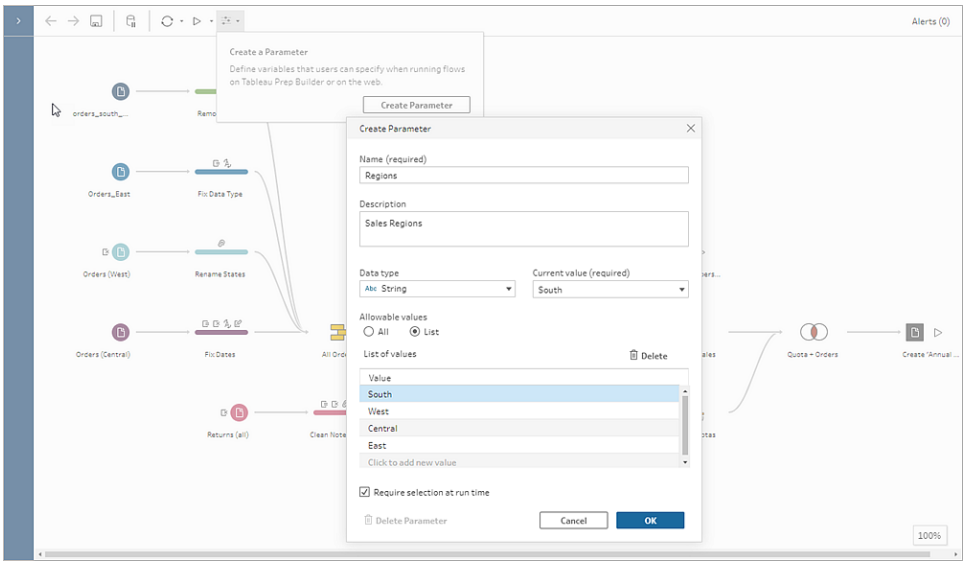
Create and Use Parameters in Flows

Supported in Tableau Prep Builder version 2021.4.1 and later and on the web in Tableau Cloud and Tableau Server version 2021.4.0 and later

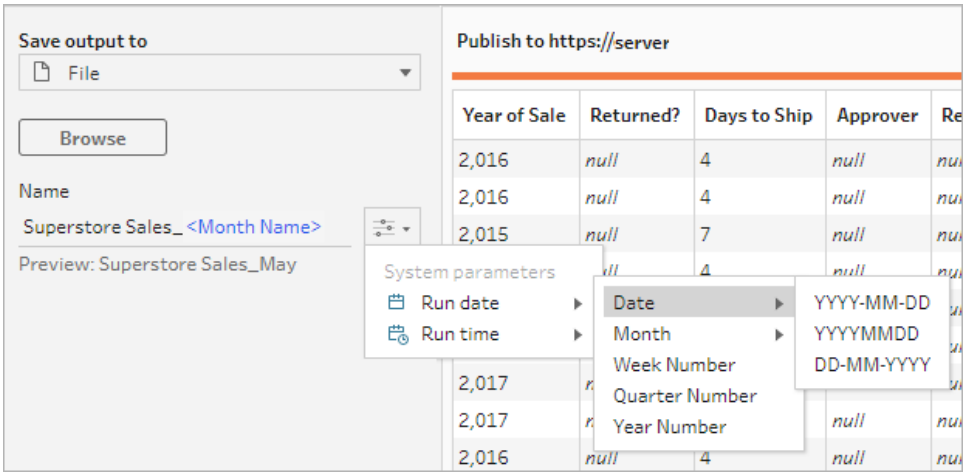
> **Note**: The content in this topic applies to authoring flows in Tableau Prep Builder and on the web, unless specifically noted. For more information about authoring flows on the web, see Tableau Prep on the Web in the Tableau Server and Tableau Cloud help.

If you often reuse flows using different data with the same schema, you can create and apply user parameters to your flows to easily transition between scenarios. A parameter is a global placeholder value such as a number, text value or boolean value that can replace a constant value in a flow.

Instead of building and maintaining multiple flows, you can now build one flow and use parameters to run the flow with your different data sets. For example, you can create a parameter for various sales regions, then apply a parameter value to the input file path to run the flow using just that region's data.



Starting in Tableau Prep Builder and Tableau Cloud version 2023.2, you can also add system parameters to the file or published data source output name to automatically add a time stamp each time you run the flow.



Where can I apply parameters?

You can apply user Parameters to file names, paths, table names, filter expressions and calculated fields, depending on the step type. Starting in version 2022.1.1, you can even include parameter override values when running flows using the REST API. For more information, see Flow Methods in the Tableau Rest API help.

You can apply system parameters (version 2023.2 and later) to output names for file and published data source output types.

The following table lists the locations where you can apply parameters for each step type.

Step type	Parameter location
Input	User parameters: • Connect to file: Use parameters in the file name or file path • Connect to database: Use parameters for the table name and in Custom SQL • Expression editor: Filters
Output	User or system parameters: • Output to file: Apply user parameters to the file name or file path and starting in version 2022.1.1, to the Microsoft Excel worksheet name. Apply system parameters to the file name. • Output to server: Apply user or system parameters to the published data source name • Output to database: Apply user parameters to the table name and starting in version 2022.1.1, to SQL scripts that you run before or after writing the flow output to a database.
Clean, New Rows, Pivot, Join, Union	User parameters: • Expression editor: Filters and calculated field values
Aggregate	User parameters: • Expression editor: Filters
Script	User parameters: • Expression editor: Filters and calculated field values
Prediction	User parameters: • Expression editor: Filters and calculated field values

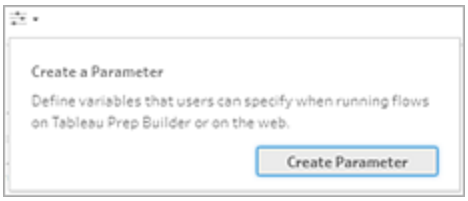
Create user parameters

User parameters are specific to the flow where they are used. Create parameters from the top menu, then define the values that apply to them. You can also define parameters that accept all values, which means any flow user can enter any value when running the flow.

You can make flow parameter values required or optional. When running the flow, users are prompted to enter the parameter values. Required parameter values must be entered before the user can run the flow. Optional parameter values can be entered or you can accept the current (default) value. The parameter values are then applied to the flow run everywhere that parameter is used.

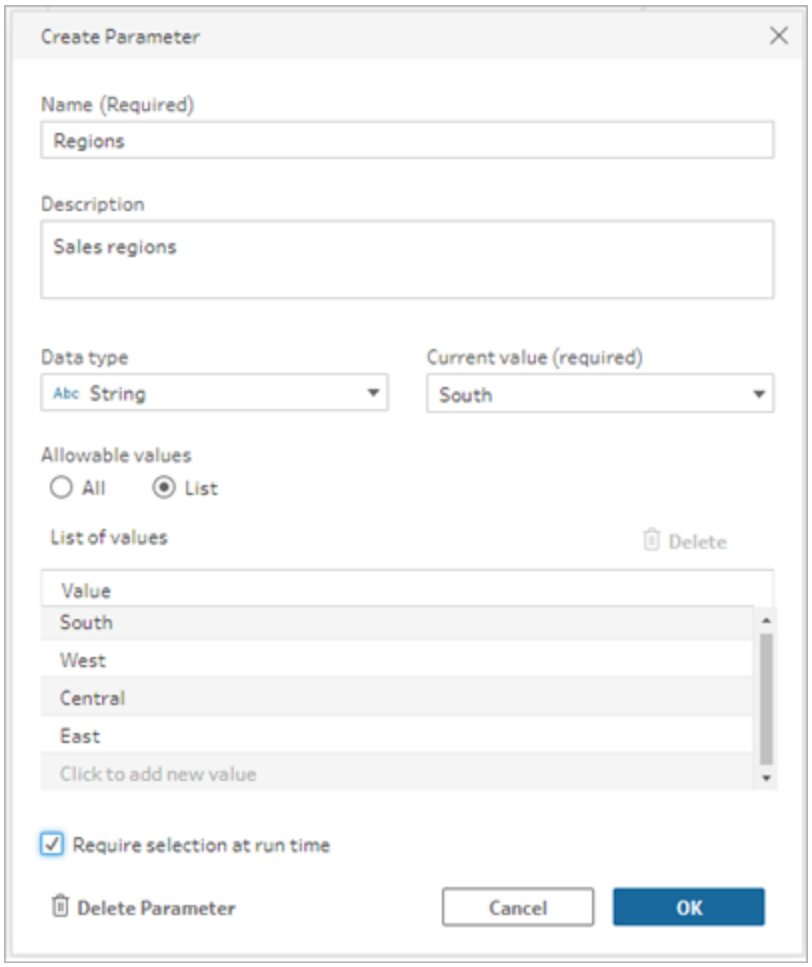
> **Note**: To run or schedule flows that include parameters on Tableau Server or Tableau Cloud, your administrator must enable the **Flow Parameter** settings on your server. For more information, see **Create and Interact with Flows on the Web** in the Tableau Server or Tableau Cloud help.

1. From the top menu, click on the **Parameter** icon, then click **Create Parameter**.



2. In the **Create Parameter** dialog, enter a name and a description (optional). The parameter name must be unique. This is the value that shows in the user interface when you add a parameter.

   If you include a description, users can see this information on hover (starting in version 2022.1.1) in the parameters list and where parameters are used.



3. Select one of the following data types. Parameter values must match the data type that you select.
    - Number (whole or decimal)
    - String
    - Boolean
4. Specify the **Allowable values**. These are the values that users can enter in the parameter.
    - **All**: This option lets users type in any value for the parameter, even when running the flow.

    > **Note**: Using this option for parameters that can be used in input and output steps can be a security risk. For example, Custom SQL queries that allow any value to be entered can expose your data assets to SQL injection attacks.

- **List**: Enter a list of values that users can choose from when applying the parameter. To enter multiple values, press **Enter** after each entry.

5. (optional) Select **Require selection at run time** (**Prompt for value at run time** in prior releases). This makes the parameter required. The user is required to enter a value when running or scheduling the flow.
6. Enter a **Current value**. This is a required value and acts as a default value for the parameter.
   - **All**: Enter a value.
   - **List**: Tableau uses the first value in your list. Use the drop-down option to change it.
   - **Boolean**: Select **True** or **False**.
7. Click **OK** to save the parameter.

Change the user parameter default value

When you create a user parameter, you have to specify a current (default) value. If a parameter is included in a flow, this value is used to:

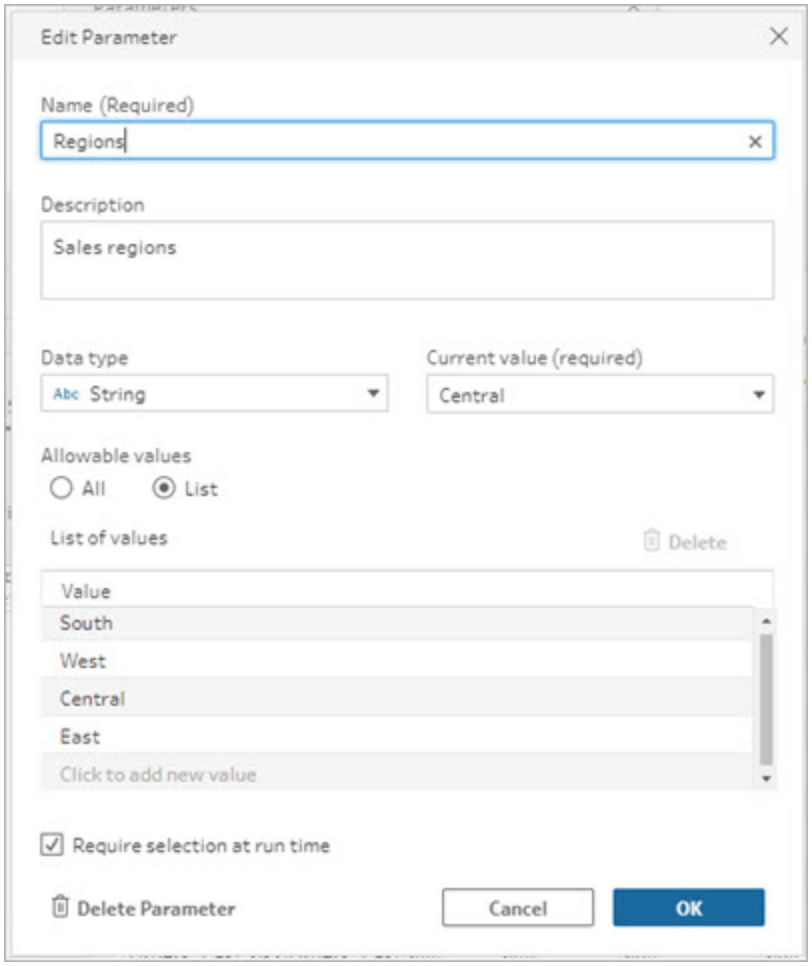
- Run Custom SQL queries defined in an input step.
- Fill in optional parameters that aren't specified at run time.
- Replace the parameter as a static value in saved steps (version 2022.1.1 and later).
- Replace the parameter as a static value in file paths when publishing flows with packaged data sets.

You can change the value at any time. From the top menu, you can edit the parameter or use the **Set** button on the parameter list. From within the flow, you can use the **Set** button anywhere the parameter is applied. When you do this, it resets the parameter's current (default) value everywhere that parameter is used, even in Custom SQL queries.

Edit user parameters

1. From the top menu, click on the **Parameter** icon.
2. Click **Edit parameter**.

3. In the **Edit Parameter** dialog, make any changes, then click **OK**.



Reset user parameter default values

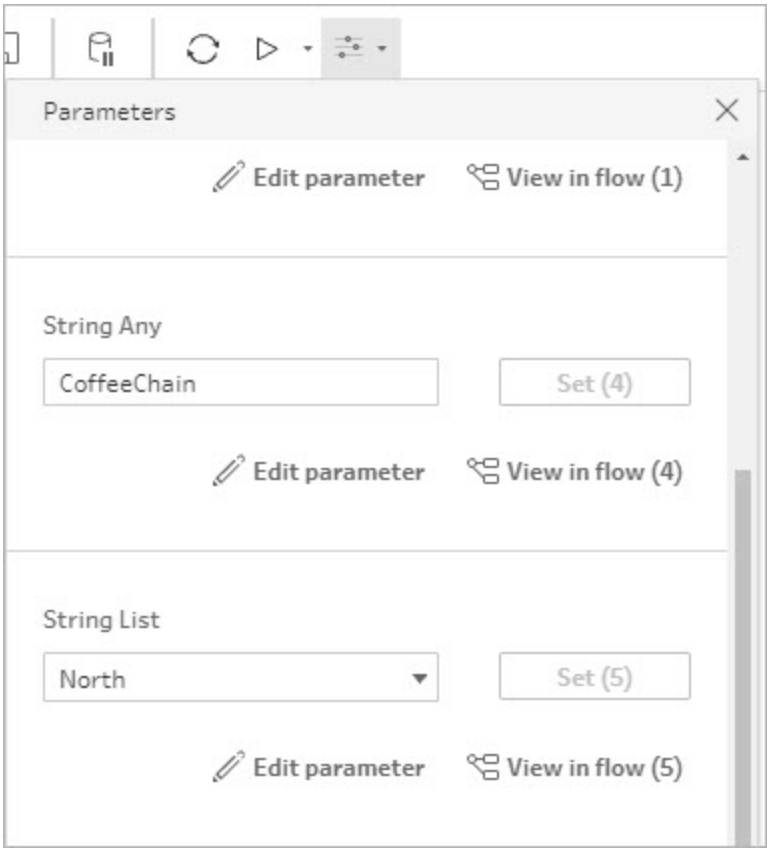
To quickly reset the parameter default value, use the **Set** button. The button shows you a count indicating the number of places in the flow where the parameter is used.

To highlight the steps in the flow that use the parameter, click **View in flow** on the parameter dialog. If there is only one place the parameter is used, you are taken directly to that step with the profile pane opened.

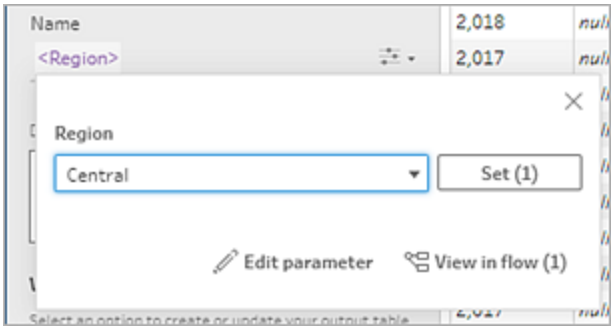
1. Do one of the following:

   - From the top menu, click on the **Parameter** icon. Use this option to reset parameter values used anywhere in the flow, or when used in filters and calculated

fields.



- Click on the parameter where it is applied in the flow. You can use this option for parameters used in filenames, file paths, table names, custom SQL and pre and post SQL scripts.



2. Select or enter the parameter value.
3. Click **Set** to apply the change.

Apply parameters to your flow

After you create user parameters, you can apply them to various places throughout your flow, depending on the step type. When the flow is run, the parameter values are applied to that flow run to produce the output for the specific data scenario.

System parameters (version 2023.2 and later) are automatically generated when you run the flow. Simply apply them to your output step name and every time the flow is run, the parameter is dynamically updated with the flow run start date or time.

Apply parameters to input steps

In an Input step you can use user parameters to replace a file name, sections of your file path, a database table name or when using Custom SQL.

Filename or file path

This option is not available when editing or authoring flows on the web.

You can include user parameters in your file path with some exceptions. Starting in version 2022.1.1, you can also see a preview of the parameter values.

Exceptions

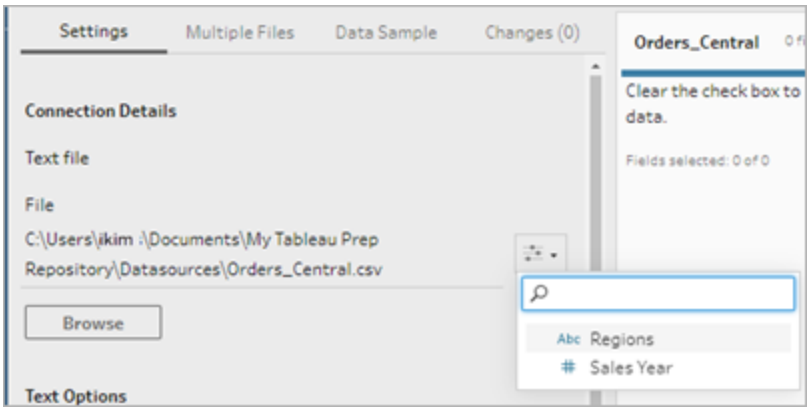
- Starting in version 2022.1.1, you can schedule and run flows on the web that include parameters in the input file path. If using an earlier version, run flows in Tableau Prep Builder or from the command line.
- To include parameters in the file path when publishing flows to the web, a direct file connection is required. Otherwise, the parameter is converted to a static value using the **Current value**.

> **Note**: Direct file connections require that the file locations are included in your organisation's safe list. For more information, see Safe List Input and Output Locations in the Tableau Server help.

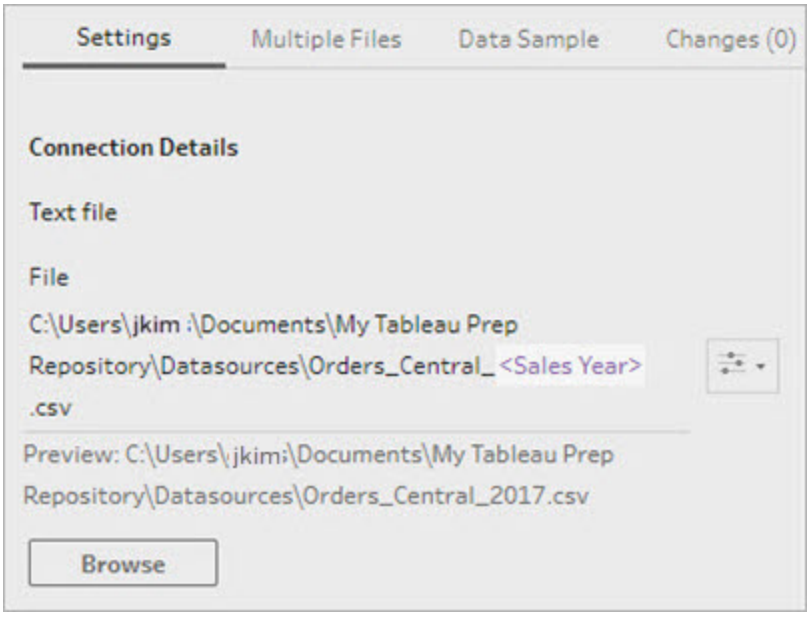
Apply a user parameter to a file name or path

1. In the **Settings** tab, in the file path, place your cursor in the location where you want to add the parameter.

2. Click on the parameter icon and select your parameter.



3. View a preview of the parameter value. The current (default) value is shown in the preview. You'll be prompted to select or enter the parameter value when you run the flow.

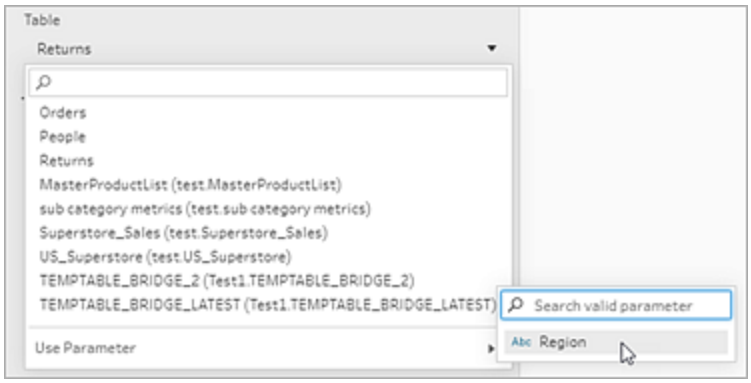


Database table

When using user parameters in table names, the entire table name must be the parameter. Using parameters for parts of a table name is not currently supported.

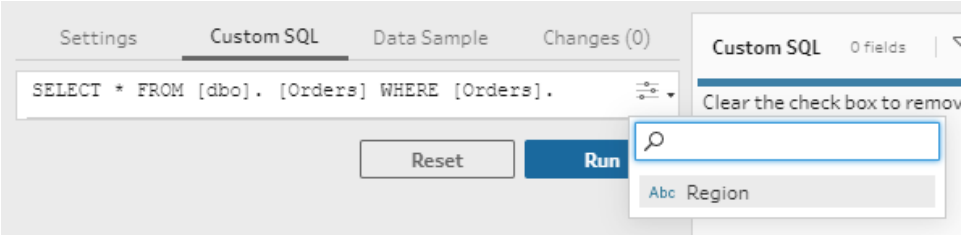
> **Note**: Using a parameter for a table name in a Google BigQuery input connection is not yet supported.

1. In the **Settings** tab, in the **Table** field, click on the drop-down menu.
2. Select **Use Parameter**, then select the parameter from the list.



Custom SQL

1. In the **Connections** pane, click **Custom SQL**.
2. In the **Custom SQL** tab, type or paste the query into the text box.
3. Click on the parameter icon and select your parameter.



4. Click **Run** to run your query. You won't be prompted to enter a parameter value until you run the flow. Instead, the query will run initially using the parameter's **Current value**.

> **Note**: If the parameter is used elsewhere in the flow and the **Current Value** is reset, that change can impact your query.

Apply user parameters to output steps

In an Output step you can apply user parameters to the following places:

- Filename
- Sections of your file path
- Published data source name
- Database table name
- Microsoft Excel worksheet name (version 2022.1.1 and later)
- Custom SQL scripts that run before or after writing flow output data to a database (version 2022.1.1 and later)

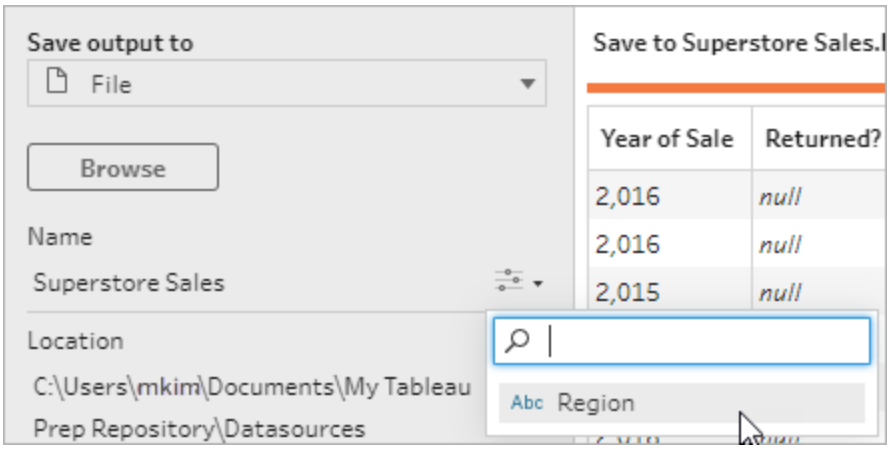
Filename or file path

This output option is not available when creating or editing flows on the web

1. In the **Output** pane, select **File** from the **Save output to** drop-down list.

2. In the **Name** or **Location** field, click on the parameter icon and select your parameter.

   For file path, place your cursor in the location where you want to add the parameter.

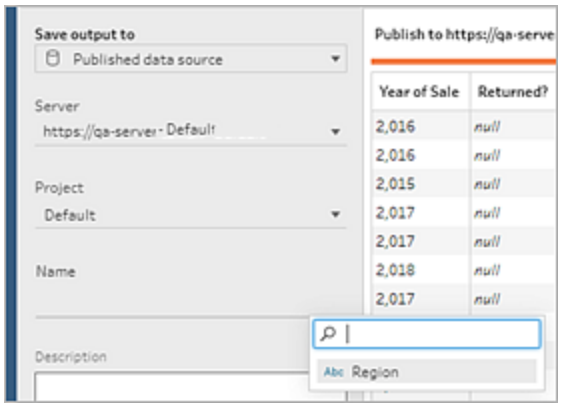


   When you run the flow, you'll be prompted to enter your parameter values.

Published data source name

1. In the **Output** pane, in the **Save output to** drop-down list, select **Published data source**.

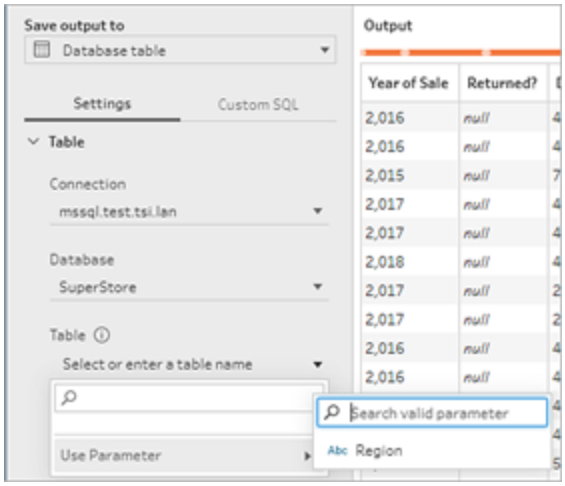
2. In the **Name** field, click on the parameter icon and select your parameter.



When you run the flow, you'll be prompted to enter your parameter values.

Database table and Before and After Custom SQL

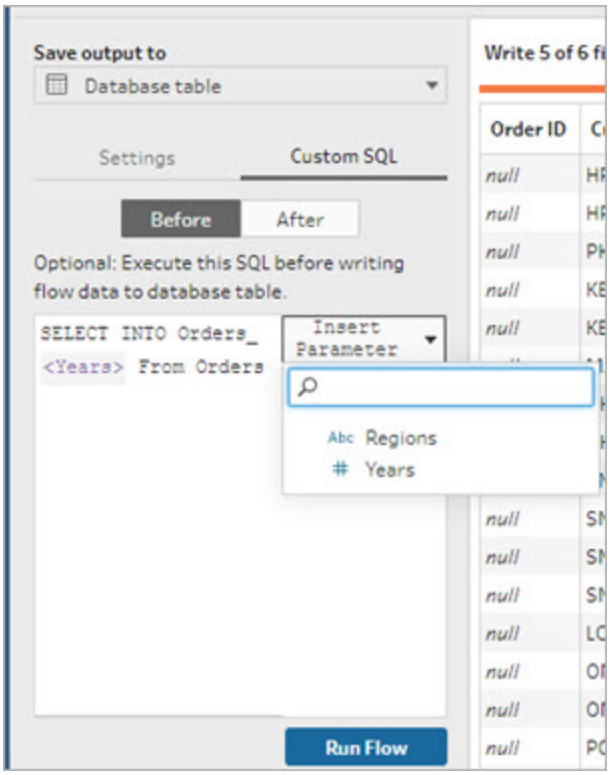
1. In the **Output** tab, in the **Save output to** drop-down list, select **Database table**.
2. In the **Table** field, select **Use Parameter**, then select the parameter from the list.



3. (Optional) Click on the **Custom SQL** tab. Starting in version 2022.1.1, you can enter a SQL script with parameters to run **Before** and **After** the data is written to the table. To include a parameter, click **Insert Parameter** and select your parameter.

   For more information about using SQL scripts when writing output to a database, see **Save flow output data to external databases** on page 394.

   > **Note**: Parameters used in SQL scripts must be manually deleted. See **Manually delete user parameters** on page 222 for more information.



When you run the flow, you'll be prompted to enter your parameter values.

Apply system parameters to output steps

In an Output step you can apply date and time system parameters to the following places:

- Filename
- Published data source name

Filename

This output option is not available when creating or editing flows on the web

1. In the **Output** pane, select **File** from the **Save output to** drop-down list.

2. In the **Name** field, click the parameter icon and select from the following run date or run time parameters. You can combine multiple system parameters to create whatever time stamp you need.

   Run date
   - **Date**: YYYY-MM-DD, YYYMMDD, DD-MM-YYYY
   - **Month**: Month Name, Month Number

- **Week Number**
- **Quarter Number**
- **Year Number**

Run time

- YYYY-MM-DD_HH-MM-SS (24 hour)
- YYYYMMDD_HHMMSS (24 hour)

When you run the flow Tableau Prep applies the flow start run time using your local time zone or the server time zone.

Published data source name

1. In the **Output** pane, in the **Save output to** drop-down list, select **Published data source**.

2. In the **Name** field, click the parameter icon and select from the following run date or run time parameters. You can combine multiple system parameters to create whatever time stamp you need.

   Run date

   - **Date**: YYYY-MM-DD, YYYMMDD, DD-MM-YYYY
   - **Month**: Month Name, Month Number
   - **Week Number**
   - **Quarter Number**
   - **Year Number**

   Run time

   - YYYY-MM-DD_HH-MM-SS (24 hour)
   - YYYYMMDD_HHMMSS (24 hour)

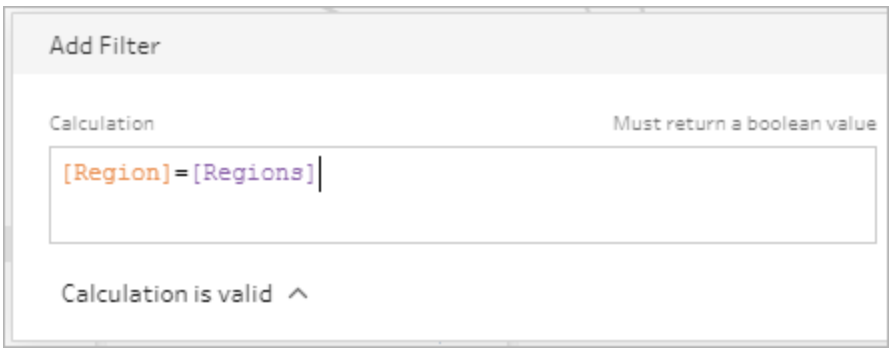
   When you run the flow Tableau Prep applies the flow start run time using your local time zone or the server time zone.

Apply user parameters to filter calculations

Use user parameters to filter data throughout your flow. Filter your data set in the input step or apply filter parameters at the step or field value level. For example, use a filter parameter to only input data for a specific region or filter data in a step to a specific department.

Note: Starting in version 2022.1, you can use copy and paste to reuse filter calculations with parameters in other flows when the same parameter exists with the same name and data type.

1. From the Input step or toolbar on the profile pane, click **Filter Values**. To add a parameter filter to a field, from the **More options** ··· menu select **Filters** > **Calculation**.
2. In the **Add Filter** calculation editor, type the name of the parameter to select it from the list (the parameter shows in purple), then click **Save** to save your filter.



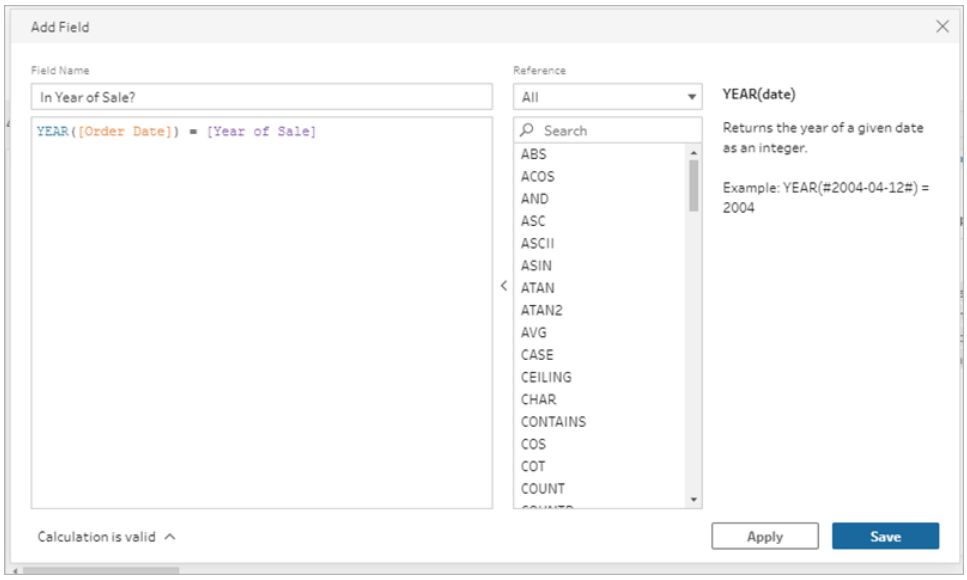
When you run the flow, you'll be prompted to enter your parameter values.

Apply user parameters to calculated fields

Use user parameters to replace constant values in calculations that you use throughout your flow. You can apply calculation parameters at the step or field value level.

Note: Starting in version 2022.1, you can use copy and paste to reuse calculations with parameters in other flows when the same parameter exists with the same name and data type.

1. From the toolbar on the profile pane, click **Create Calculated Field**. To add a parameter to a calculation on a field, from the **More options** ··· menu, select **Create Calculated Field** > **Custom Calculation**.
2. In the **Add Field** calculation editor, enter your calculation, type the name of the parameter to select it from the list and then click **Save** to save your calculation.



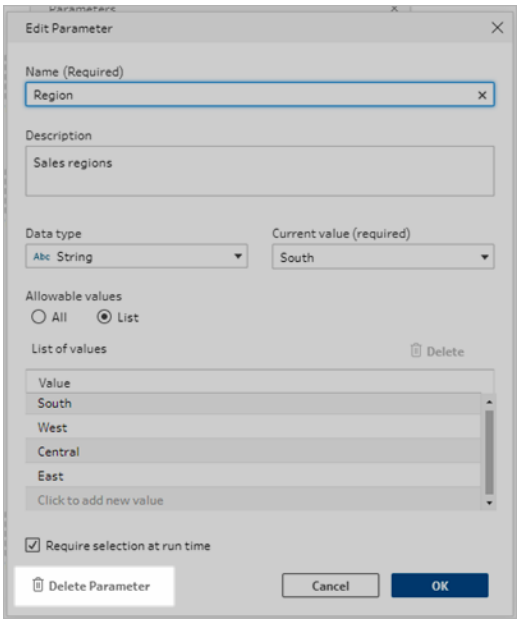
When you run the flow, you'll be prompted to enter your parameter values.

Delete user parameters

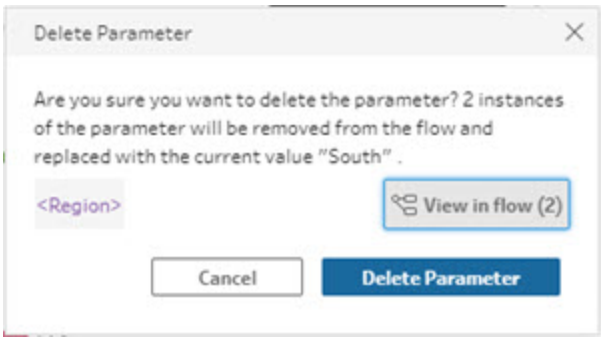
To delete user parameters that you no longer need, click **Delete Parameter** in the **Edit Parameter** dialog. This removes any instance of the parameter used throughout the flow and replaces it with the parameter's **Current value**. This action can't be undone.

> **Note**: The options to delete parameters in a flow vary depending on your version. Use the instructions below for version 2022.1 and later. Use **Manually delete user parameters** on the next page for previous versions and to delete parameters used in Custom SQL scripts that run before or after writing output to a database.

1. From the top menu, click on the parameter icon drop-down menu, then click **Edit parameter** for the parameter you want to delete.
2. In the **Edit Parameter** dialog, click **Delete Parameter**.



3. In the confirmation dialog, click **Delete Parameter** again. You can click **View in flow** to highlight the steps and investigate where the parameter is used before you delete it.



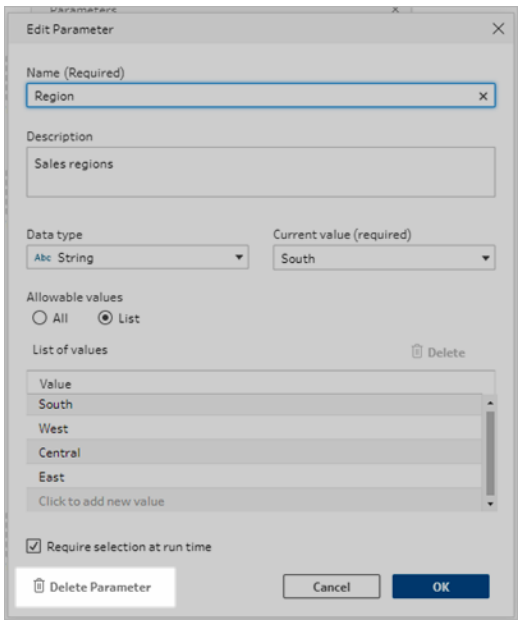
Manually delete user parameters

Applies to version 2021.4.4 and earlier and parameters used in pre and post Custom SQl scripts

Before you can delete a user parameter from your parameters list, you must first find and remove all instances of the parameters from your flow, even from the **Changes** pane.

1. From the top menu, click on the parameter icon drop-down menu.
2. For the parameter that you want to delete, click **View in flow** to find all instances where the parameter is used in the flow.

   If the parameter is not used anywhere in the flow, skip to step 4.
3. For each step where the parameter is used, remove the parameter, including deleting any changes listed in the **Changes** pane.
4. From the top menu, click on the parameter icon drop-down menu and, for the parameter you want to delete, click **Edit parameter**.
5. In the **Edit Parameter** dialog, click **Delete Parameter**.



   The parameter will be replaced with the parameter's **Current value**.

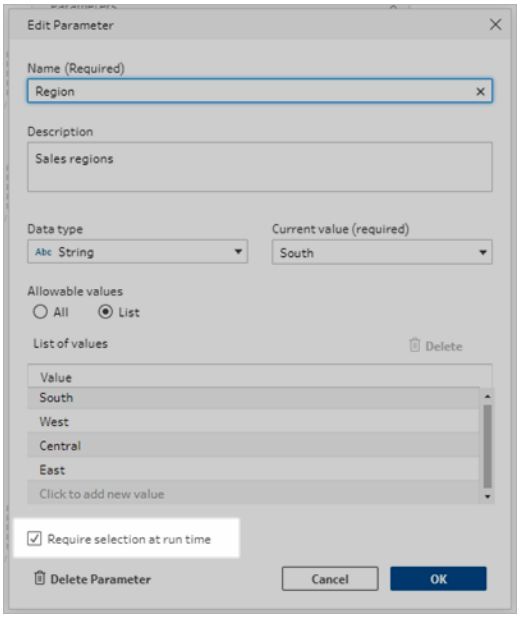
Run flows with parameters

Running flows that include parameters is the same as running flows that don't have them, except that users are prompted to enter user parameter values at run time or when adding the flow to a schedule in Tableau Server or Tableau Cloud.

System parameters are applied automatically when the flow is run.

If a user parameter is marked as a required, users must enter a value before they can run the flow. If a parameter is optional, users can enter a value or accept the parameter's **Current value** by default.

Required parameters are those that have the **Require selection at run time** (**Prompt for value at run time** in prior releases) tick box selected.

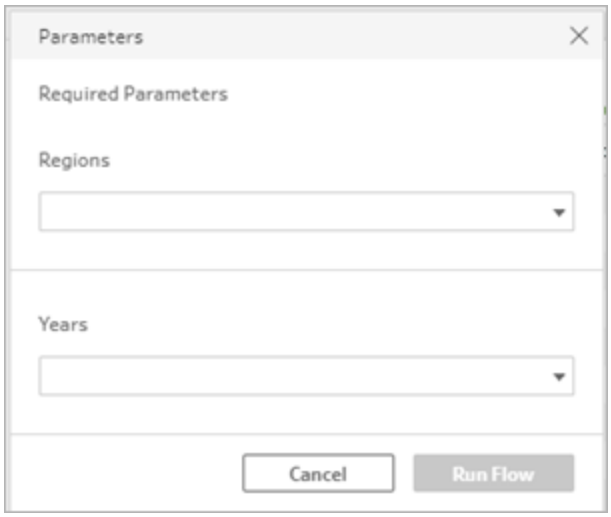


If you run flows using the command line interface and want to override the current (default) parameter values, create a parameters override .json file and include the -p --parameters syntax in your command line. For more information, see **Refresh flow output files from the command line** on page 425.

Run flows manually

When you run a flow from Tableau Prep Builder or manually in Tableau Server or Tableau Cloud, the **Parameters** dialog opens when you click **Run**.

1. Enter or select the user parameter values. If there are optional parameters in the flow, you can enter the values at this time or accept the current (default) parameter value.
2. Click **Run Flow** to run the flow.



For more information about running flows, see **Publish a Flow to Tableau Server or Tableau Cloud** on page 463.

Run flows on a schedule

When you schedule flows to run on Tableau Server or Tableau Cloud, you will need to enter any required user parameter values when scheduling the flows.

1. On the **New Tasks** or **Linked Tasks** tab, in the **Set Parameters** section, enter or select the parameter values. If there are optional parameters in the flow, you can enter the values at this time or leave the field empty to use the current (default) parameter value.
2. Click **Create Tasks** to schedule your flow.

Clean and Shape Data

Note: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

Tableau Prep offers various cleaning operations that you can use to clean and shape your data. Cleaning up dirty data makes it easier to combine and analyse your data or makes it easier for others to understand your data when sharing your data sets.

You can also clean your data using a pivot step or a script step to apply R or Python scripts to your flow. Script steps aren't supported in Tableau Cloud. For more information, see **Pivot Your Data** on page 332 or **Use R and Python scripts in your flow** on page 341.

About cleaning operations

You clean data by applying cleaning operations such as filtering, adding, renaming, splitting, grouping or removing fields. You can perform cleaning operations in most step types in your flow. You can also perform cleaning operations in the data grid in a cleaning step.

You can apply limited cleaning operations in the Input step and can't apply cleaning operations in the output step. For more information about applying cleaning operations in the Input step, see **Apply cleaning operations in an input step** on page 114.

Available cleaning operations

The following table shows which cleaning operations are available in each step type:

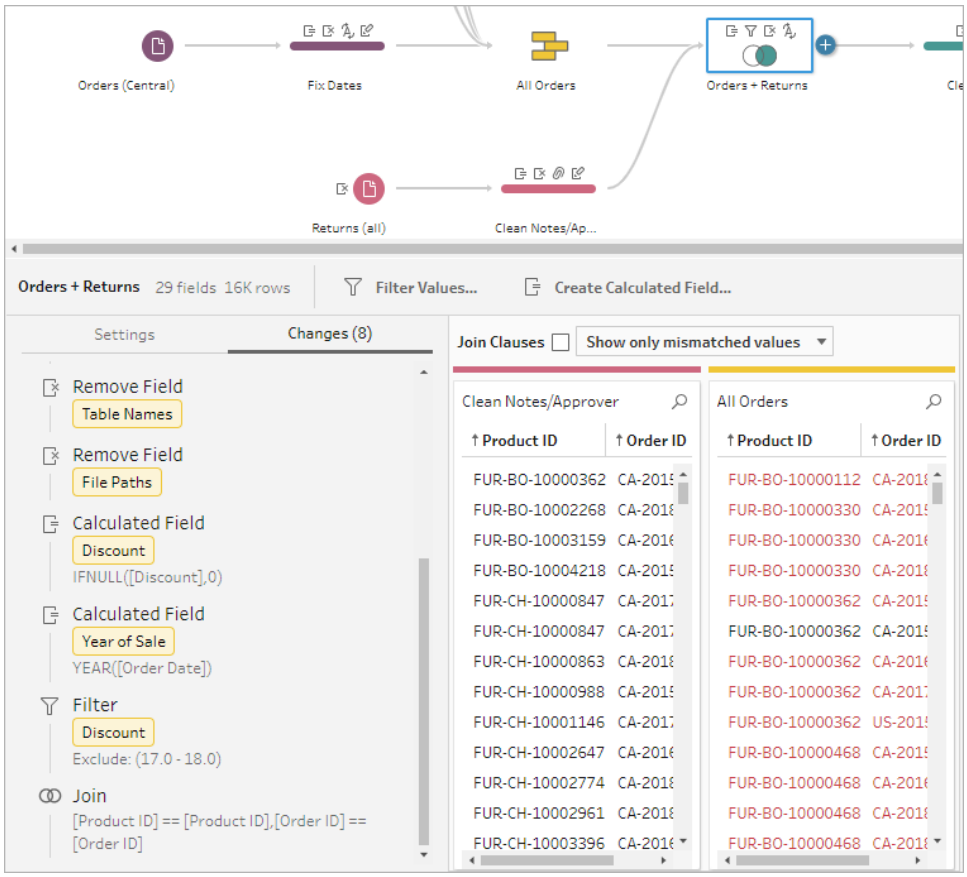
	Input	Clean	Aggregate	Pivot	Join	Union	New Rows	Output
Filter	X	X	X	X	X	X	X	
Group Values		X		X		X	X	

Clean		X		X	X	X	X	
Convert Dates		X	X	X	X	X	X	
Split Values		X		X	X	X	X	
Rename Field	X	X		X	X	X	X	
Rename Fields (in bulk)		X						
Duplicate Field		X		X	X	X	X	
Keep Only Field	X	X	X	X	X	X	X	
Remove Field	X	X	X	X	X	X	X	
Create Calculated Field		X		X	X	X	X	
Edit Value		X		X	X	X	X	
Change Data Type	X	X	X	X	X	X	X	

As you make changes to your data, annotations are added to the corresponding step in the **Flow** pane and an entry is added in the **Changes** pane to track your actions. If you make changes in the Input step, the annotation appears to the left of the step in the **Flow** pane and appears in the **Input profile** in the field list.

The order that you apply your changes matters. Changes made in Aggregate, Pivot, Join and Union step types are performed either before or after those cleaning actions, depending on where the field is when you make the change. Where the change was made is shown in the **Changes** pane for the step.

The following example shows changes made to several fields in a Join step. The change is performed before the join action to give the corrected results.



Order of operations

The following table shows where the cleaning action is performed in Aggregate, Pivot, Join and Union step types depending on where the field is in the step.

Action	Step Type:	Aggregate	Aggregate	Pivot	Pivot	Join	Join	Union	Union	New Rows
	Field Location:	*Grouped fields*	*Aggregated fields*	*Not in pivot*	*Created from pivot*	*Included in one tabl-*	*Included in both table-*	*Mismatched fields*	*Combined fields*	*Field used to generate*

						*e**	*s**			*rows*
Filter		Before Aggregation	After Aggregation	Before Pivot	After Pivot	Before Join	After Join	Before Union	After Union	After New Rows
Group Values		NA	NA	Before Pivot	After Pivot	Before Join	After Join	Before Union	After Union	After New Rows
Clean		NA	NA	Before Pivot	After Pivot	Before Join	After Join	Before Union	After Union	After New Rows
Convert Dates		Before Aggregation	After Aggregation	Before Pivot	After Pivot	Before Join	After Join	Before Union	After Union	After New Rows
Split Values		NA	NA	Before Pivot	After Pivot	Before Join	After Join	Before Union	After Union	After New Rows
Rename Field		NA	NA	Before Pivot	After Pivot	Before Join	After Join	Before Union	After Union	Before New Rows
Duplicate Field		NA	NA	Before Pivot	After Pivot	Before Join	After Join	Before Union	After Union	After New Rows
Keep Only		After Aggreg-	After Aggreg-	Before	After Pivot	Before	After Join	Before Union	After Union	After New

Field		ation	ation	Piv-ot		Join				Rows
Remo-ve Field		Remov-es from Aggreg-ation	Remov-es from Aggreg-ation	Bef-ore Piv-ot	After Pivot	Befo-re Join	After Join	Before Union	After Union	After New Rows
Create Cal-culate-d Field		NA	NA	Bef-ore Piv-ot	After Pivot	After Join	After Join	Before Union	After Union	After New Rows
Edit Value		NA	NA	Bef-ore Piv-ot	After Pivot	Befo-re Join	After Join	Before Union	After Union	After New Rows
Chang-e Data Type		Before Aggreg-ation	After Aggreg-ation	Bef-ore Piv-ot	After Pivot	Befo-re Join	Befo-re Join	Before Union	After Union	Befor-e New Rows

Note: For joins, if the field is a calculated field that was created using a field from one table, the change is applied before the join. If the field is created with fields from both tables, the change is applied after the join.

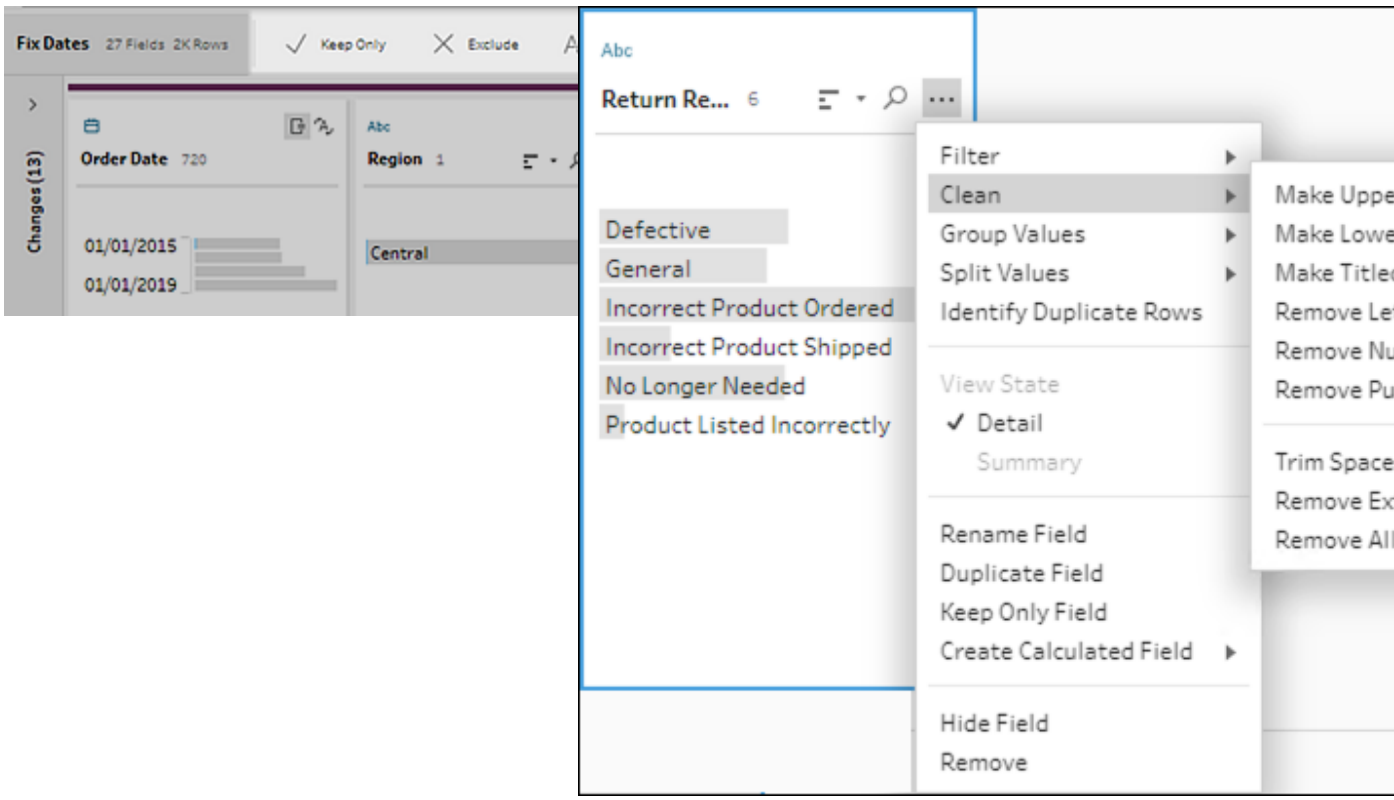
Apply cleaning operations

To apply cleaning operations to fields, use the toolbar options or click **More options** ··· on the field profile card, data grid or Results pane to open the menu.

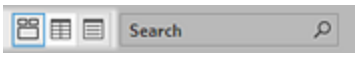
In Aggregate, Pivot, Join and Union step types, the **More options** ··· menu is available on the profile cards in the Results pane and corresponding data grid. If you perform the same cleaning

operations or actions again and again throughout your flow, you can copy and paste your steps, actions or even fields. For more information see **Copy steps, actions and fields** on page 262.

Profile pane toolbar **Drop-down menu**

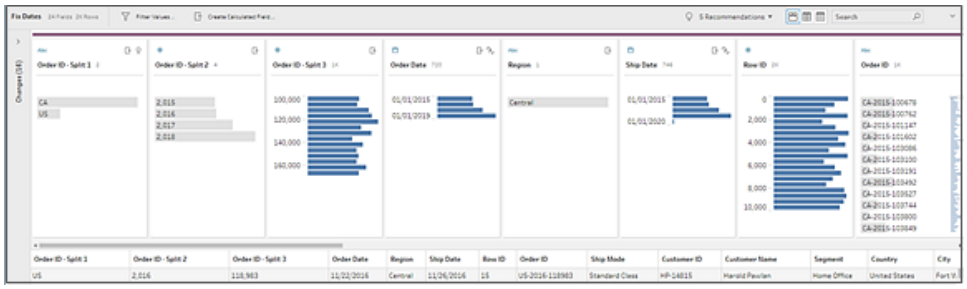


Select your view

You can perform cleaning operations outside of the profile or results pane in the data grid or in the list view. Use the view toolbar 

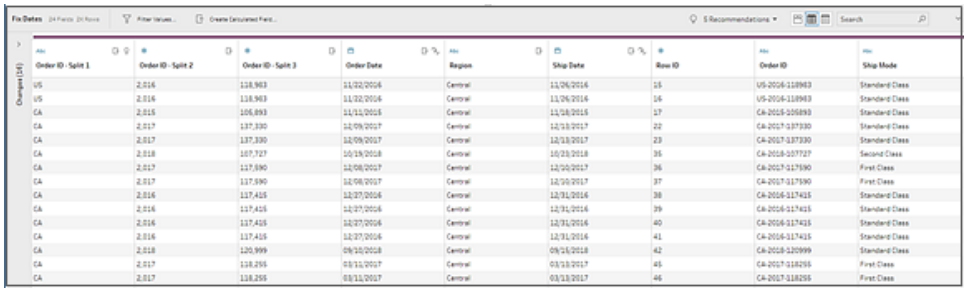
(Tableau Prep Builder version 2019.3.2 and later and on the web) to change your view, then click **More options** ··· on a field to open the cleaning menu.

- **Show profile pane**: This is the default view. Select this button to go back to the Profile pane or Results pane view.



- **Show data grid**: Collapse the profile or results pane to expand and show only the data grid. This view provides a more detailed view of your data and can be useful when you need to work with specific field values. After you select this option, this view state persists across all steps in your flow, but you can change it at any time.

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



- **Show list view** (Tableau Prep Builder version 2019.3.2 and later and on the web): Convert the profile pane or results pane into a list. After you select this option, this view state persists across all steps in your flow, but you can change it at any time.

  In this view you can:

  - Select and remove multiple rows using the **X** option.
  - (version 2021.1.4 and later) Select and hide or unhide multiple rows using the option.
  - (version 2021.2.1 and later) Rename fields in bulk.
  - Use the **More options** ··· menu to apply operations to selected fields.

If you assign a data role to the field, or select **Filter**, **Group Values**, **Clean** or **Split Values**, you're returned to the Profile or Results view to complete those actions. All other options can be performed in the list view.

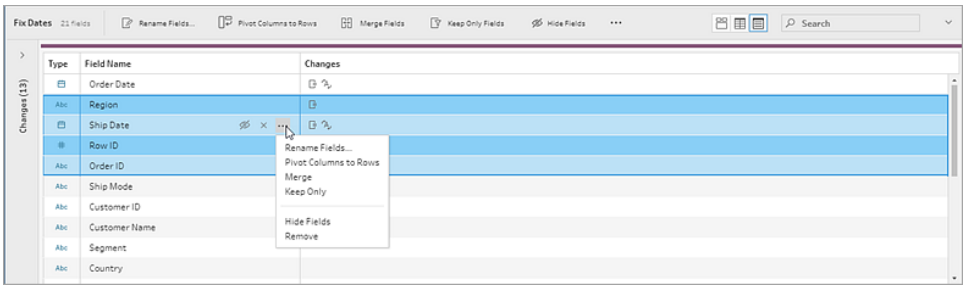
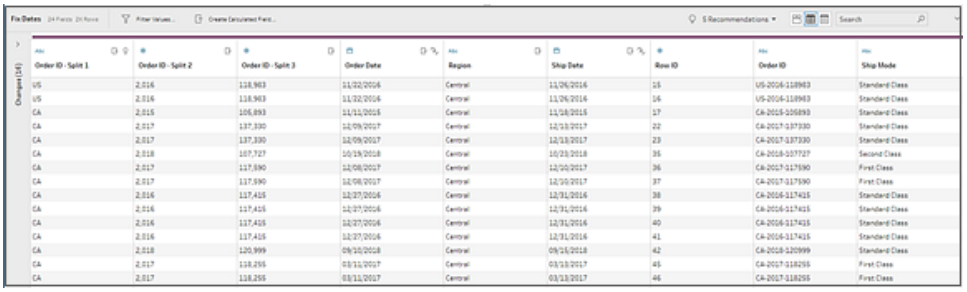


Tableau Prep Builder version 2019.3.1 and earlier

Use the view toolbar

to hide the Profile pane and show only the data grid. Then click **More options** ··· on a field in the data grid to open the cleaning menu. This view shows a more detailed view of your data and can be useful when you need to work with specific field values. After you select this option, this view state persists across all steps in your flow, but you can change it at any time.

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



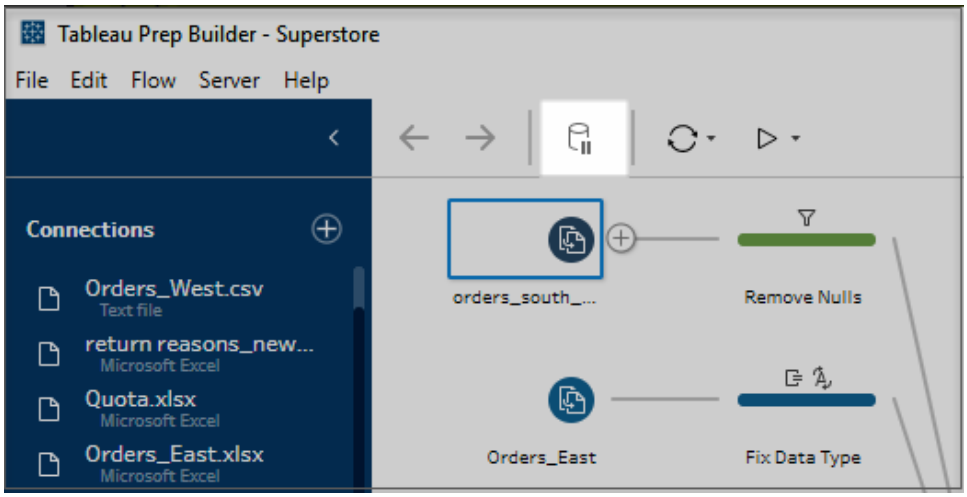
Pause data updates to boost performance

As you perform cleaning operations on your data, Tableau Prep applies your changes as you go to show you the results immediately. To save valuable processing time when you know the changes you need to make and don't need immediate feedback as you make each change, you can boost performance by pausing data updates.

When you pause data updates, you can make all your changes at once, then resume updates to see your results. You can resume data updates and enable all available operations at any time.

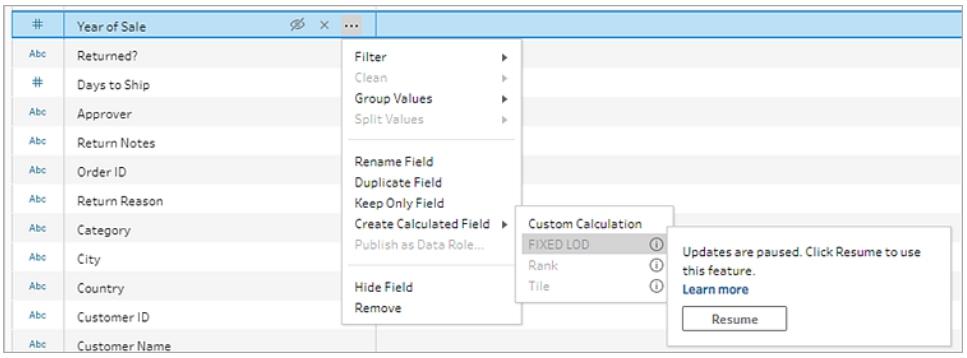
> **Note**: When you pause data updates, any operations that require you to see your values are disabled. For example if you want to apply a filter to selected values, you need to see the values you want to exclude.

1. In the top menu, click **Pause data updates** to pause updates.



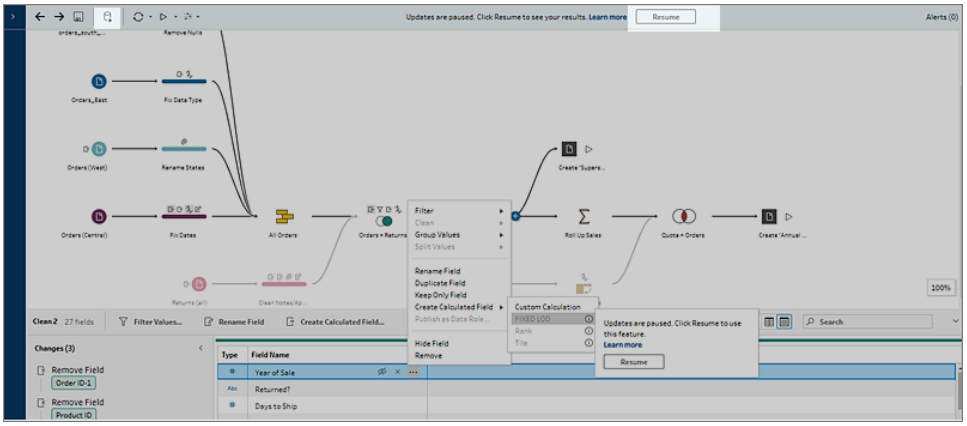
2. Tableau Prep converts the Profile pane into the List view. In List view, use the **More options** ··· menu to apply operations to selected fields. If the operation requires you to see your values, it is disabled. To enable the operation, you must resume data updates.

   For more information about using List view mode, see **Select your view** on page 232.



3. To see the results of your changes or enable a disabled feature, resume data updates. Click the **Resume data updates** button, click the **Resume** button in the menu dialog or in the message banner at the top of the Flow pane.

> **Note**: Tableau Prep Builder gives you an option to resume updates directly from the menu. If editing flows on the web, you'll need to resume updates from the top menu.



Apply cleaning operations

To apply cleaning operations to a field, do the following:

> **Note**: You can perform cleaning operations in a list view beginning in Tableau Prep Builder version 2019.3.2 and on Tableau Server and Tableau Cloud starting in version 2020.4.

1. In the **Profile** pane, data grid, Results pane or list view, select the field you want to make changes to.

2. From either the toolbar or **More options** ··· menu for the field, select from the following options:

   - **Filter** or **Filter Values**: Select from one of the filter options, right-click or Ctrl-click (MacOS) a field value to keep or exclude values. You can also use the **Selected Values** filter to pick and choose the values to filter, included values not in your flow sample. For more information about filter options, see **Filter Your Data** on page 175.

   - **Group Values** (**Group and Replace** in prior versions): Manually select values or use automatic grouping. You can also multi-select values in the Profile card and right-click or Ctrl-click (MacOS) to group or ungroup values or edit the group value. For more information about using **Group Values**, see **Automatically map values to a standard value using fuzzy match** on page 257.

   - **Clean**: Select from a list of quick cleaning operations to apply to all values in the field.

   - **Convert Dates** (Tableau Prep Builder version 2020.1.4 and later and on the web): For fields assigned to a Date or Date & Time data type, select from a list of DATEPART quick cleaning operations to convert your date field values to an integer value representing year, quarter, month, week, day or a date and time value.

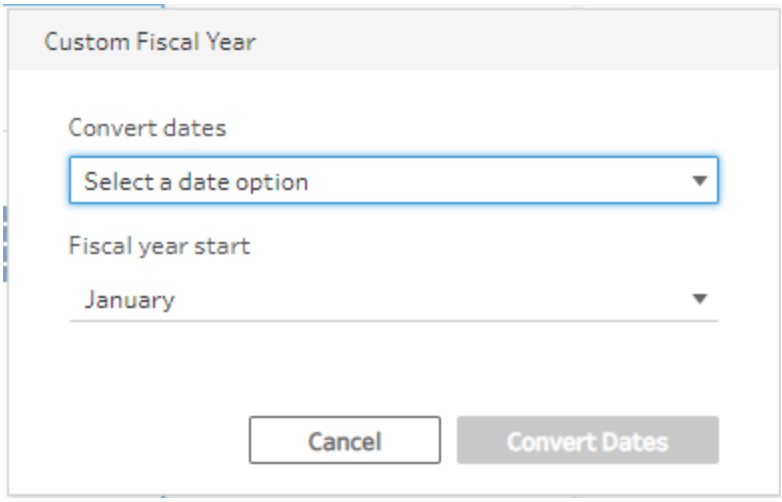
     Starting in version 2021.1.4, you can also select from two DATENAME quick cleaning operations, day of the week or month name, to convert your date field values.

     - **Custom Fiscal Year** (Tableau Prep Builder version 2020.3.3 and later and on the web): If your fiscal year doesn't start in January, you can set a custom fiscal month to convert the date using that month instead of the default month of January.

       This setting is on a per field basis, so if you want to apply a custom fiscal year to other fields, repeat this same step.

       To open the dialog, from the **More options** ··· menu, select **Convert Dates** >

Custom Fiscal Year.



- **Split Values**: Split values automatically based on a common separator or use custom split to specify how you want to split field values.

  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**: Edit the field name.
- **Duplicate Field** (Tableau Prep Builder version 2019.2.3 and later and on the web): Create a copy of your field and values.
- **Keep Only Field**(Tableau Prep Builder version 2019.2.2 and later and on the web): Keep only the selected field and exclude all other fields in the step.
- **Create Calculated Field**: Write a customised calculation in the Calculation editor or use the Visual Calculation editor (Tableau Prep Builder version 2020.1.1 and later and on the web) to create level of detail, rank or row number calculations. For more information, see **Create Level of Detail, Rank and Tile Calculations** on page 275.
- **Publish as Data Role**: Create custom data roles that you can then apply to your fields to validate the field values when cleaning data. For more information about this option, see **Create customised data roles** on page 194.

- **Hide Field**: If you have fields you want to keep in your flow but don't need to clean, you can hide them out of the way instead of removing them. For more information, see **Hide fields** on page 177.
- **Remove** (**Remove Field** in previous versions): Remove the field from the flow.

3. To edit a value, right-click or Ctrl-click (MacOS) one or more values and select **Edit Value** then enter a new value. You can also select **Replace with Null** to replace the values with a Null value or double-click in a single field to edit it directly. For more information about editing field values see **Edit field values** on page 248.
4. Review the results of these operations in the **Profile** pane, Summary panes or data grid.

Rename fields in bulk

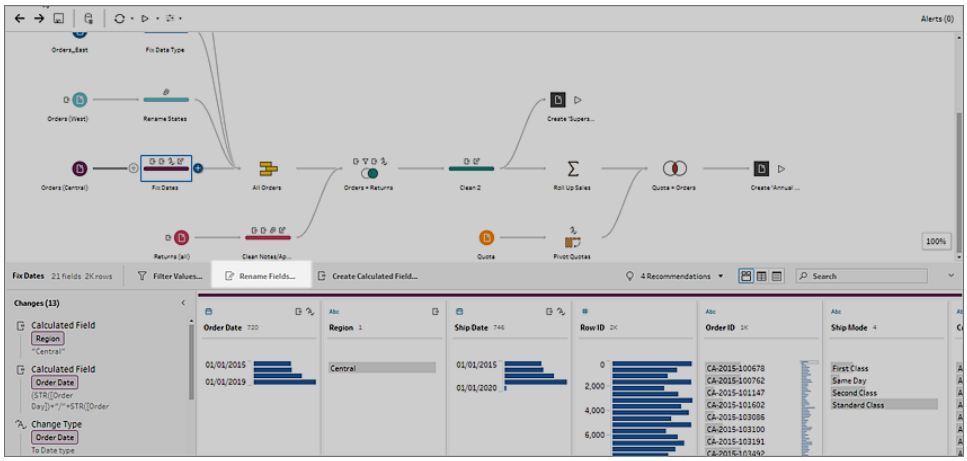
Supported in Tableau Prep Builder version 2021.2.1 and later. Supported in Tableau Prep on the web in Tableau Server and Tableau Cloud version 2021.2 and later.

Use the **Rename Fields** option to rename multiple fields in bulk. Search for parts of a field name to replace or remove it, or add prefixes or suffixes to all or selected fields in your data set.

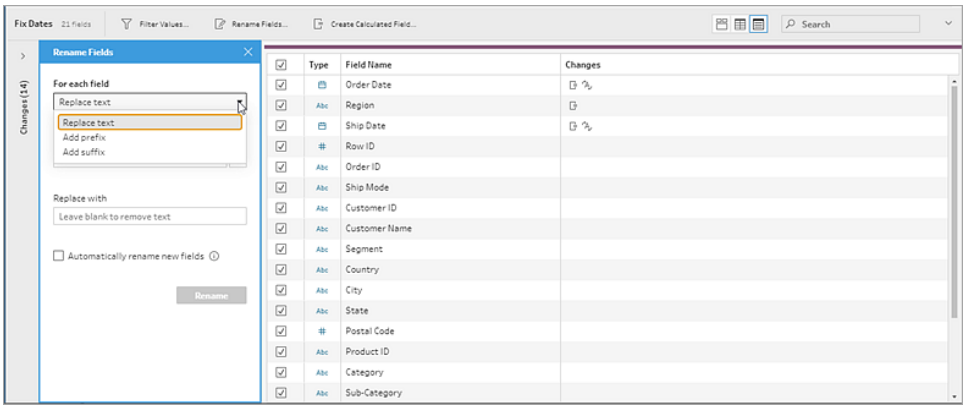
You can also automatically apply the same change to any fields added in the future that match your criteria by selecting the **Automatically rename new fields** check box when making your changes.

> **Note**: This option is only available in a **Clean** step type.

1. In a **Clean** step, from the toolbar, select **Rename Fields**.



Your view is automatically converted to the **List view** showing all the fields in your flow. You can use the **Search** option in the toolbar to narrow your results.

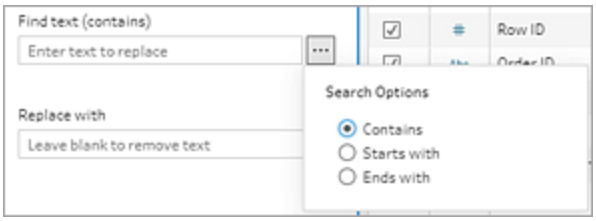


All fields are selected by default. Clear the top check box to clear the selection for all fields to manually select only the fields you want to change.

2. In the **Rename Fields** pane, select from the following options:

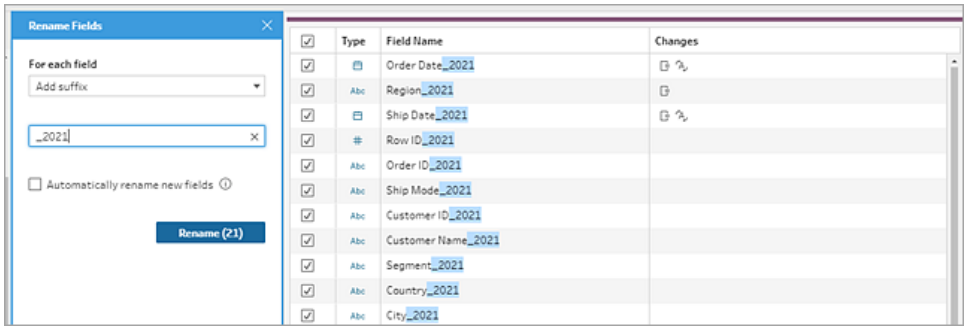
   - **Replace text**: In the **Find text** field, find matching text using the **Search options**, then enter the replacement text in the **Replace with** field. To find blank spaces, press the space bar in the **Find text** field.

     > **Note**: Renaming fields can't result in blank or duplicate field names.

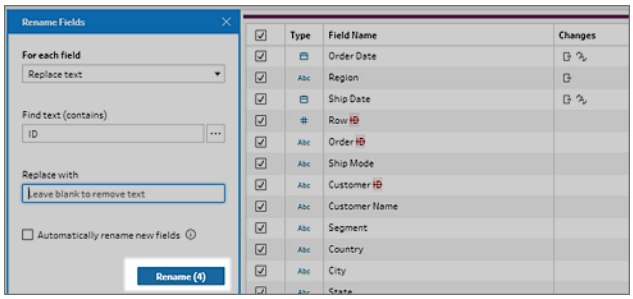
     

   - **Add prefix**: Add text to the beginning of all selected field names.
   - **Add suffix**: Add text to the end of all selected field names.

As you make your entries, your results display in the **List view** pane.



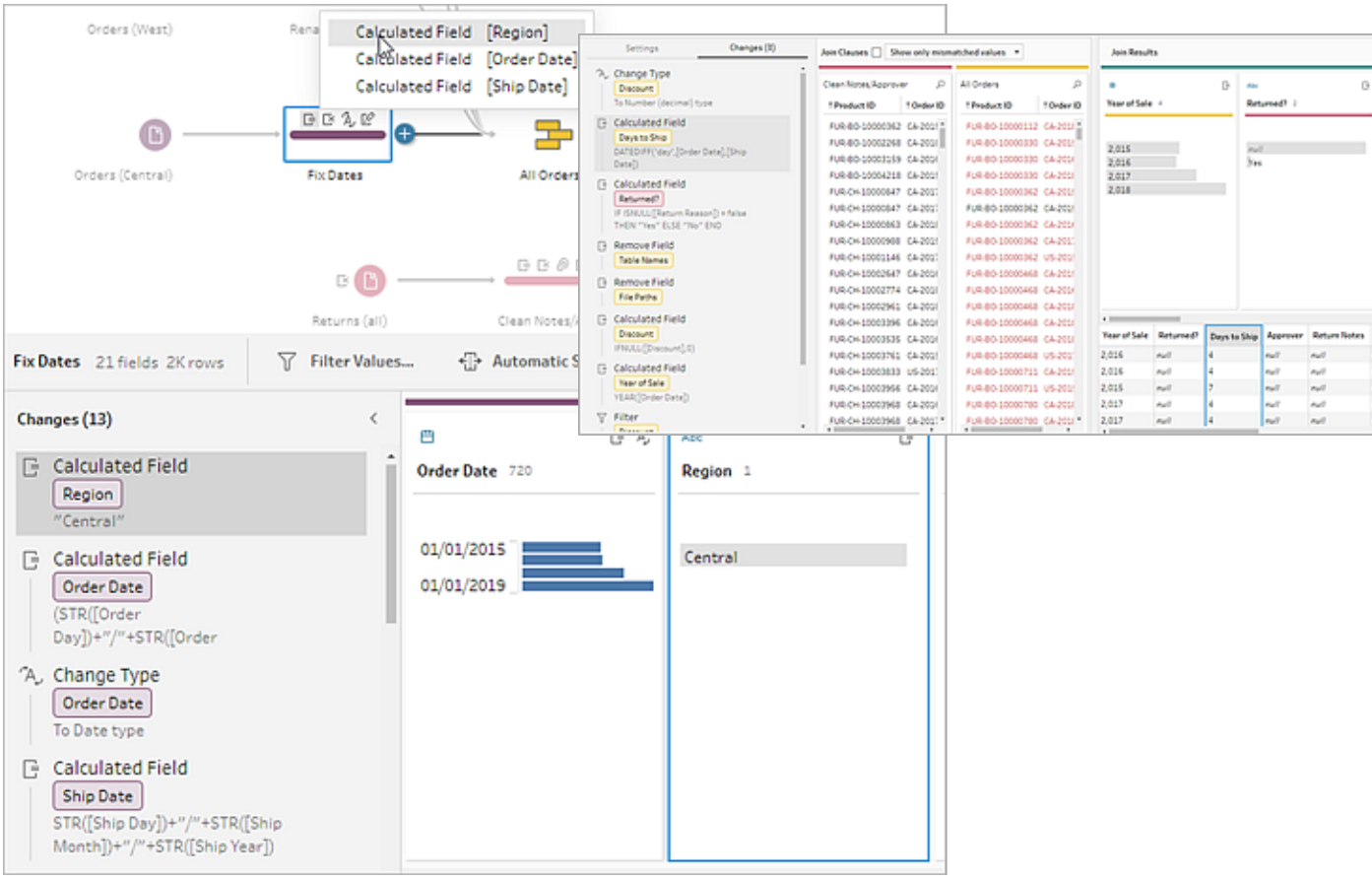
3. (optional) Select **Automatically rename new fields** to automatically apply these same changes to new fields that match your replacement criteria when your data is refreshed.
4. Click **Rename** to apply your changes and close the pane. The **Rename** button shows the number of fields that are impacted by your changes.



View your changes

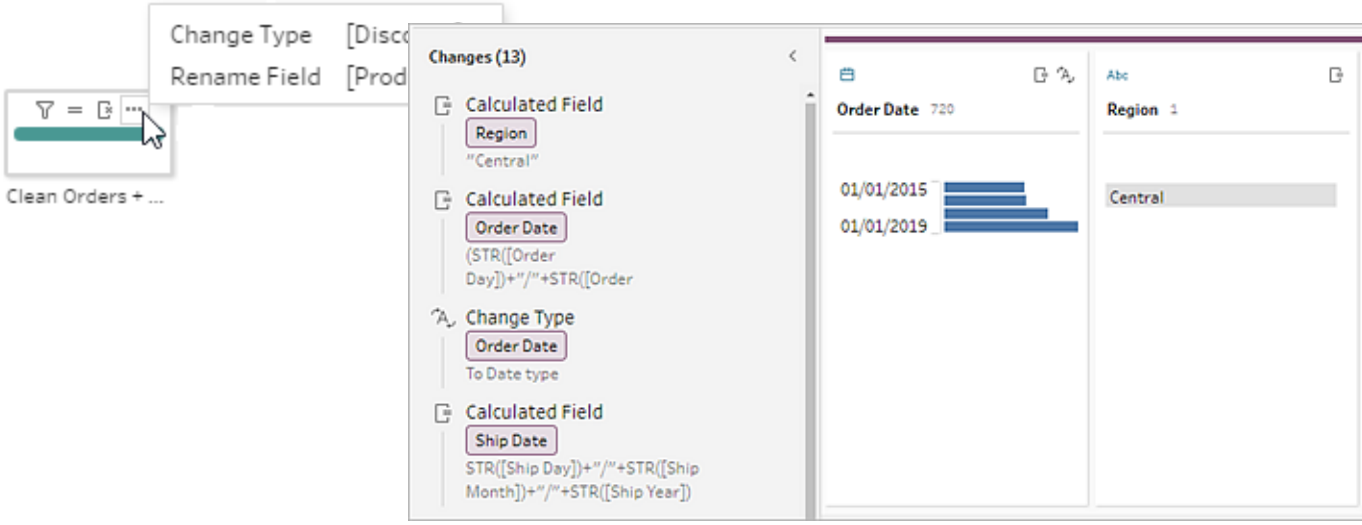
The different types of cleaning operations are represented by icons over the steps in your flow. If more than four types of operations are applied to a step, an ellipsis displays over the step. Hover over these icons to view annotations showing applied operations and the order in which they are performed.

Starting in Tableau Prep Builder version 2019.1.3 and later and on the web, you can click on an annotation on the change icon on a step in the Flow pane or on a profile card in the Profile or Results pane and the change and field it impacts will be highlighted in the **Changes** pane and the **Profile** or **Results** pane.

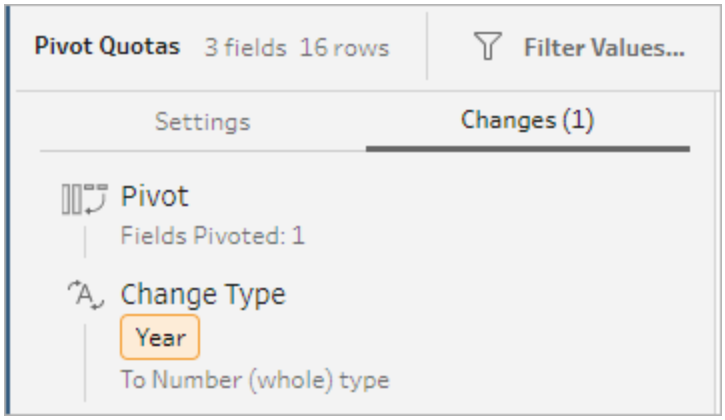


You can also select a step and then expand the **Changes** pane to view the details for each change, edit or remove your changes, drag changes up or down to change the order in which they're applied and add a description to provide context to other users. For more information about adding descriptions to your changes, see **Add descriptions to flow steps and cleaning actions** on page 155.

Cleaning annotation **Changes pane**



When viewing changes in an Aggregation, Pivot, Join, or Union step, the order that the change is applied shows either before or after the reshaping action. The order of these changes is applied by the system and cannot be changed. You can edit and remove the change.



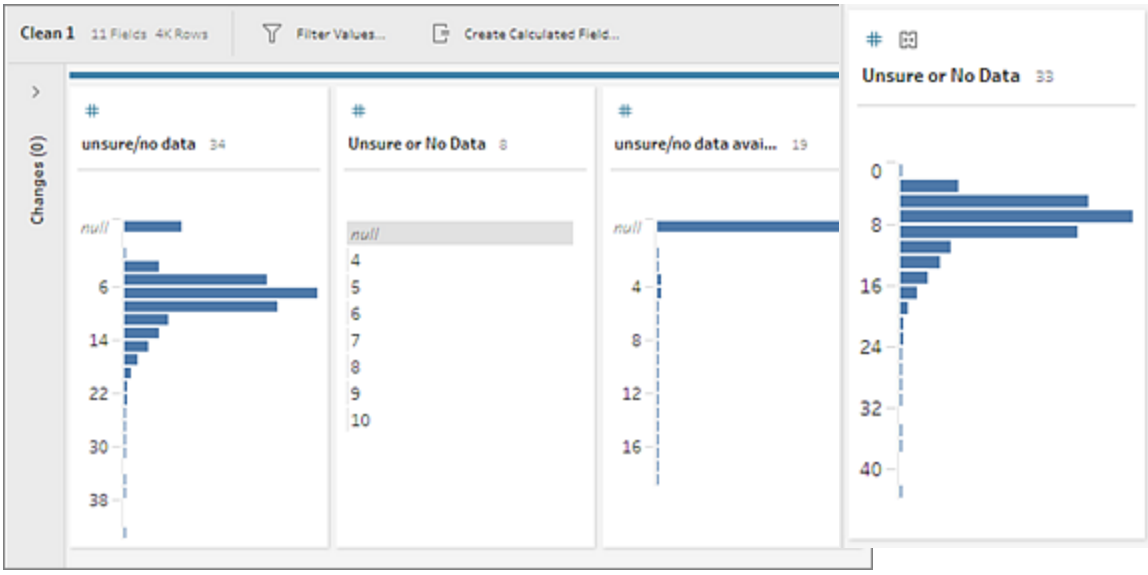
Merge fields

If you have fields that contain the same values but are named differently, you can easily merge them into a single field to combine them by dragging one field on top of the other. When you merge the fields, the target field becomes the primary field and the field name of the target field persists. The field that you merge to the target field is removed.

Example:

Input union results in 3 fields with the same values

Merge 3 fields into 1



When you merge fields, Tableau Prep keeps all of the fields from the target field and replaces any nulls in that field with values from the source fields that you merge with the target field. The source fields are removed.

Example

Name	Contact_Phone	Business_Phone	Cell_Phone	Home_Phone
Bob	123-4567	123-4567	null	null
Sally	null	null	456-7890	789-0123
Fred	null	null	null	567-8901
Emma	null	234-5678	345-6789	null

If you merge the **Business _Phone**, **Cell_Phone** and **Home_Phone** fields with the **Contact_ phone** field, the other fields are removed and results in the following:

Name	Contact_Phone
Bob	123-4567

Sally	456-7890
Fred	567-8901
Emma	234-5678

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 toolbar.

For information about how to fix mismatched fields as a result of a union, see **Fix fields that don't match** on page 370.

Apply cleaning operations using recommendations

Sometimes it can be hard to identify which cleaning operation you need to use to fix problems in your data. Tableau Prep can analyse your data and recommend cleaning operations that you can apply automatically to quickly fix problems in your data fields or help to identify problems so you can fix them. This feature is available in all step types except Input, Output and Join step types.

> **Note**: In Tableau Prep Builder, if you don't want to use this feature, you can turn it off. From the top menu, go to **Help** > **Settings and Performance**. Then click on **Enable Recommendations** to untick the box next to the setting.

Recommendation types include:

- Data roles
- Filter
- Group values (also applies to fields with data roles starting in Tableau Prep Builder version 2019.2.3 and on the web)
- Pivot columns to rows (Tableau Prep Builder version 2019.4.2 and later and on the web)
- Replace values with Null values

- Remove fields
- Split (Tableau Prep Builder version 2019.1.1 and later and on the web)

> **Note**: This option works specifically with data in fixed-width type text files. To use the split recommendation with this file type, after you connect to the data source, in the Input step, in the **Text Settings** tab, select a **Field Separator** character that is not used in the data so the data loads as a single field.

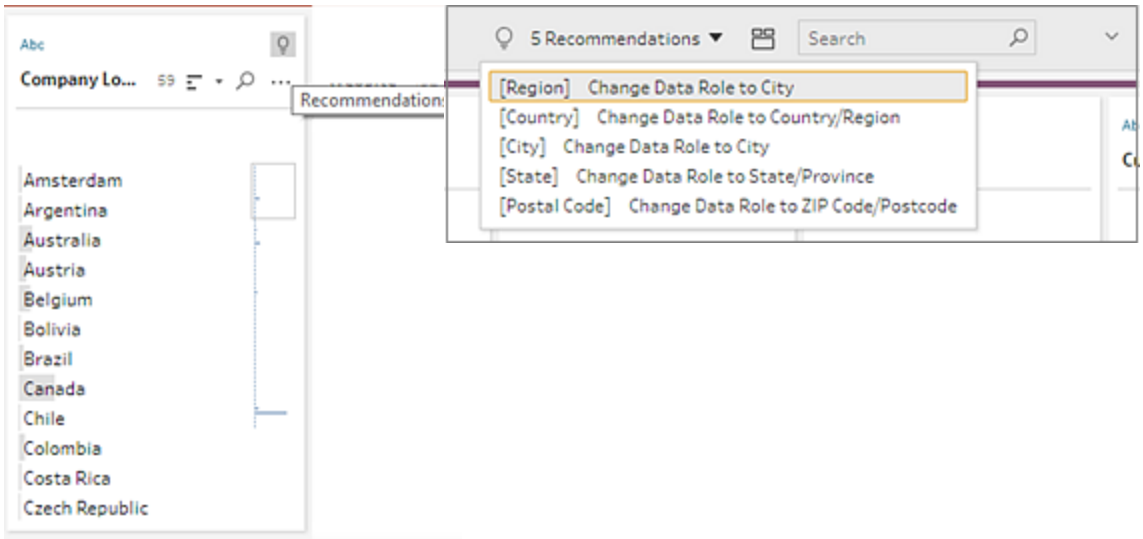
- Trim spaces

Apply recommendations

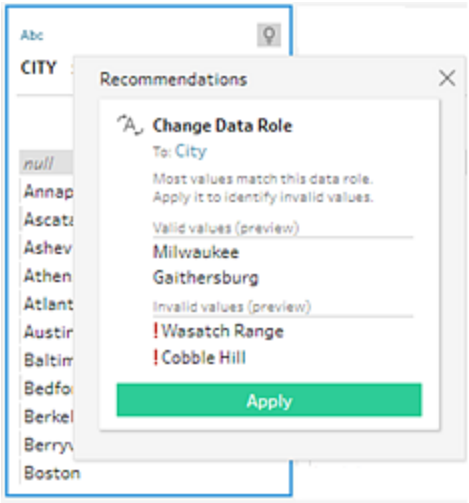
1. Do one of the following:

   - Click the light bulb icon in the top right corner of the profile card.
   - From the toolbar, click the **Recommendations** drop-down arrow to view all recommendations for your data set and select a recommendation from the list.

   This option only appears when recommended changes are identified by Tableau Prep.



2. To apply the recommendation, hover on the Recommendations card and then click **Apply**.

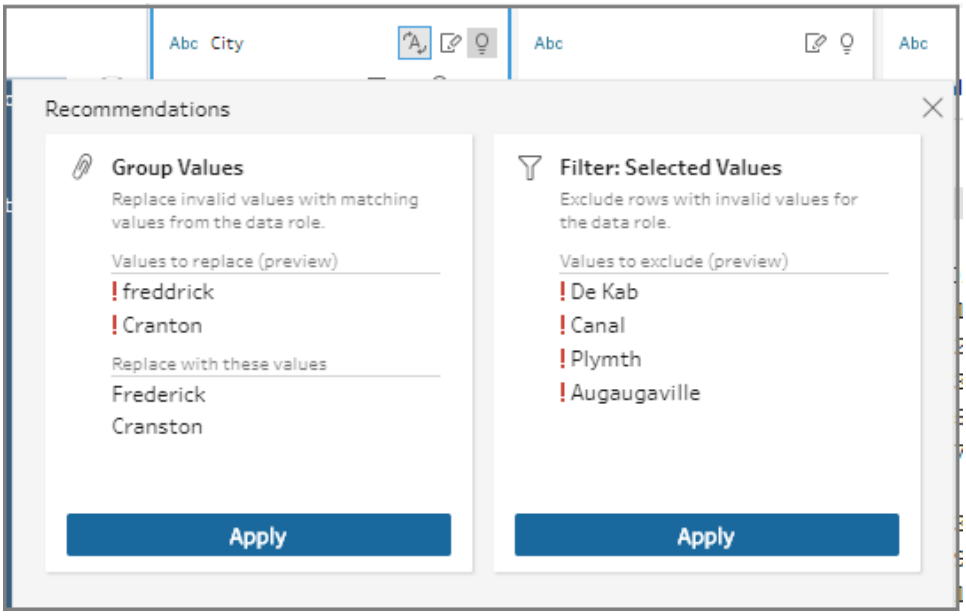


The change is automatically applied and an entry is added to the **Changes** pane. To remove the change, click **Undo** in the top menu or hover over the change in the **Changes** pane and click the **X** to remove it.

If you apply a recommendation to pivot fields, a Pivot step is automatically created where you can then perform any additional pivot actions like renaming the pivoted fields or pivoting on additional fields.

3. If Tableau Prep identifies further recommendations as a result of the change, the light bulb icon remains on the Profile card until no further recommendations are found.

   Repeat the steps above to apply any additional changes or ignore the suggested change and use the other cleaning tools to address the data problems.



Edit field values

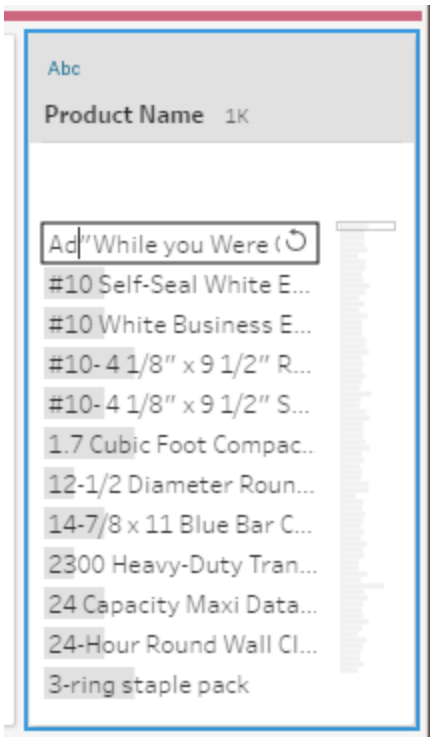
Multiple variations of the same value can prevent you from accurately summarising your data. You can quickly and easily correct these variations using the following options.

> **Note**: Any edits that you make to the values must be compatible with the field data type.

Edit a single value

1. In the **Profile** card, 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, automatically group values together using fuzzy-match algorithms that find similar values or select multiple values and replace them with Null.

> **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.

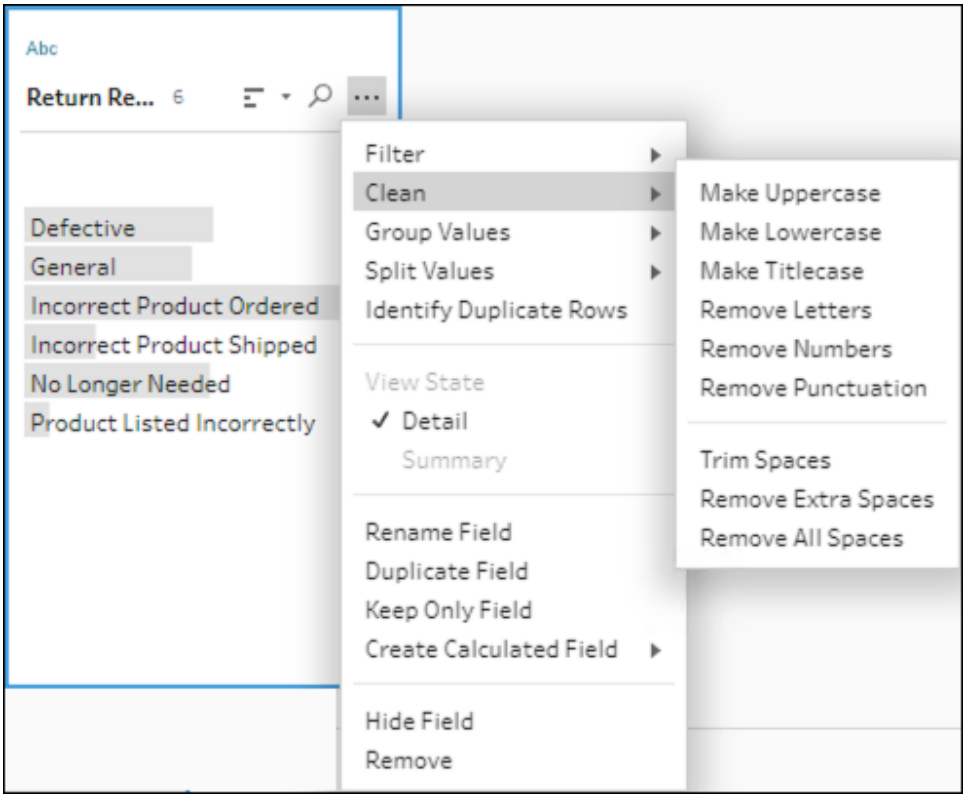
Edit multiple values using quick cleaning operations

This option applies only to text fields.

1. In the **Profile** pane, Results pane or data grid, select the field you want to edit.

2. Click **More options** ··· , 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.
- **Make Titlecase**: Change all values to title case 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.
- **Remove Extra Spaces**: Remove leading and trailing whitespace and replace extra whitespace in-between characters with a single space.
- **Remove All Spaces**: Remove all whitespace, including leading and trailing whitespace and any whitespace in between characters.



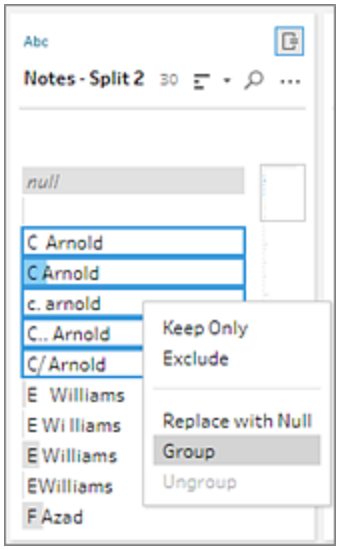
You can stack operations to apply multiple cleaning operations to the fields. For example first select **Clean** > **Remove Numbers** and then select **Clean** > **Remove Punctuation** to remove all numbers and punctuation from the field values.

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

Use this option to manually select multiple values and group them under a standard value in the profile card. To use other methods to group values, see **Manually map multiple values to a standard value** on the next page and **Automatically map values to a standard value using fuzzy match** on page 257.

1. In the **Profile** card, select the field you want to edit.
2. Press Ctrl or Shift+click or Command or Shift+click (MacOS), 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.

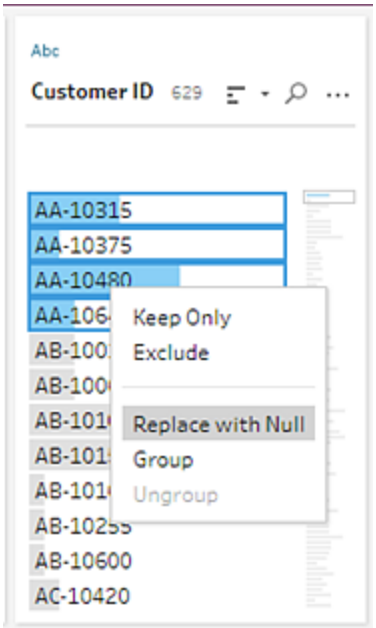


4. To edit the group name, select the grouped field and edit the value or right-click or Ctrl+click (Mac) on the grouped field and select **Edit Value** from the context menu.
5. To ungroup the grouped field values, right-click on the grouped field and select **Ungroup** from the context menu.

Replace one or more values with Null

If you have data rows that you want to include in your analysis but you want to exclude certain field values you can change them to a Null value.

1. In the **Profile** card, press Ctrl or Shift+click or Command or Shift+click (on Mac), and select the values that you want to change
2. Right-click or Ctrl+click (Mac), and select **Repace with Null** from the menu. The values are changed to Null and the group icon shows next to the value.



Manually map multiple values to a standard value

Use **Group Values** (**Group and Replace** in previous versions) 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 organise 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 Values** 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.

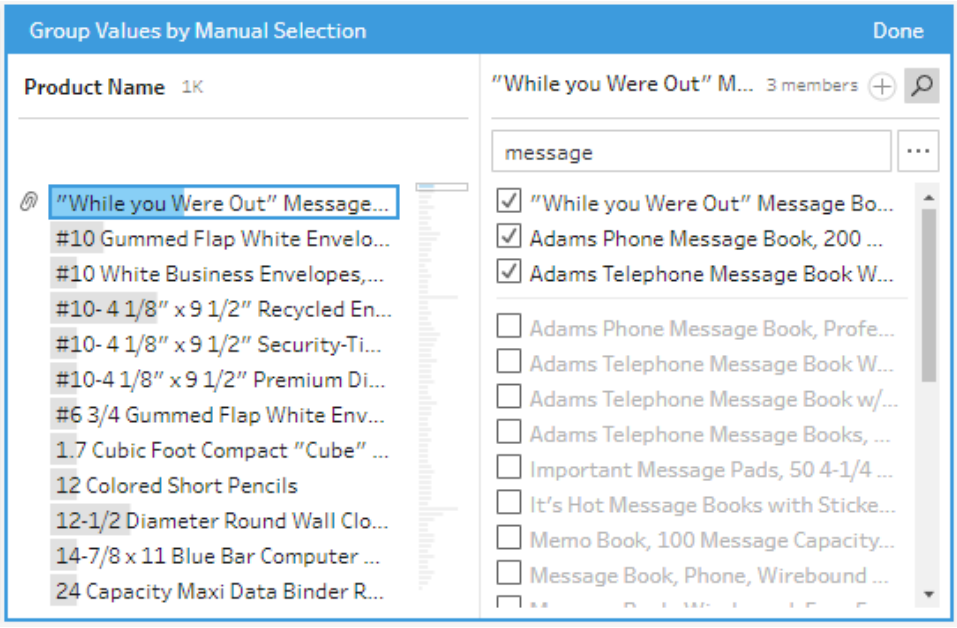
Map multiple values to a single selected field

1. In the **Profile** pane or Results pane, select the field you want to edit.

2. Click **More options** ··· and select **Group Values** (**Group and Replace** in previous versions) > **Manual Selection** from the menu.

3. In the left pane of the **Group Values** 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 Values** 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 Values** editor, untick the box next to the values.

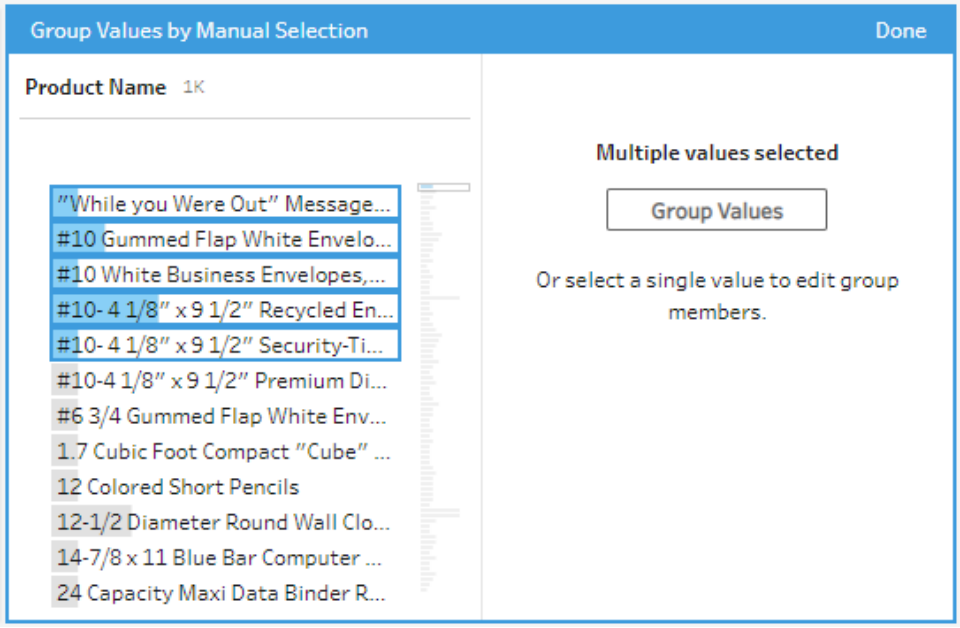
Create a group by selecting multiple values

1. In the **Profile** pane or Results pane, select the field you want to edit.

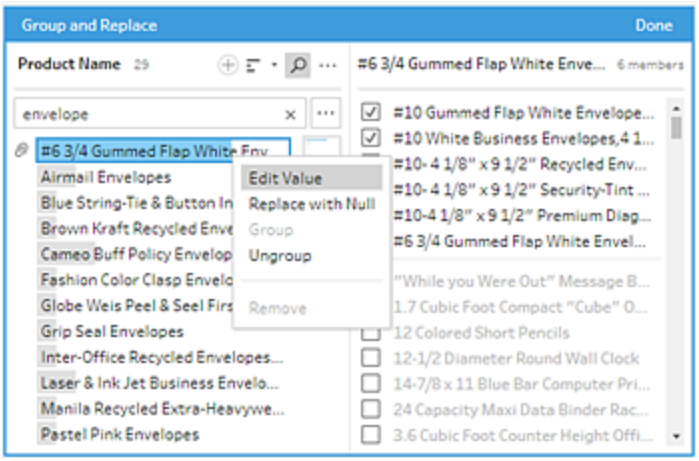
2. Click **More options** ··· and select **Group Values** (**Group and Replace** in previous versions) > **Manual Selection** from the menu.

3. In the left pane of the **Group Values** editor, select multiple values that you want to group.

4. In the right pane of the **Group Values** editor, click **Group Values**.



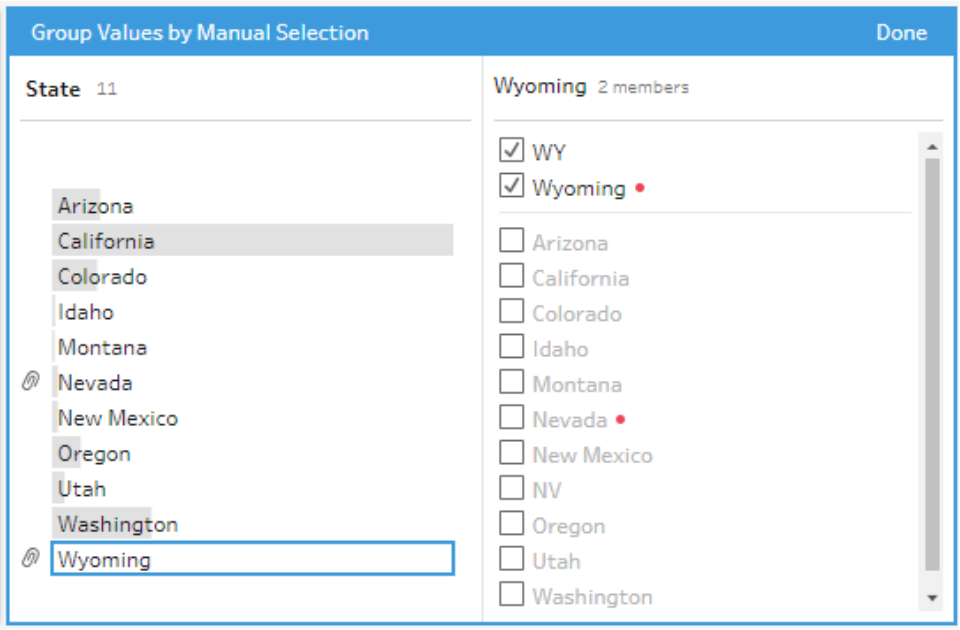
A new group is created using the last selected value as the group name. To edit the group name, select the grouped field and edit the value or right-click or Ctrl+click (MacOS) on the grouped field and select **Edit Value** from the menu.



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 Values** (**Group and Replace** in previous versions). 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 Values** editor.

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



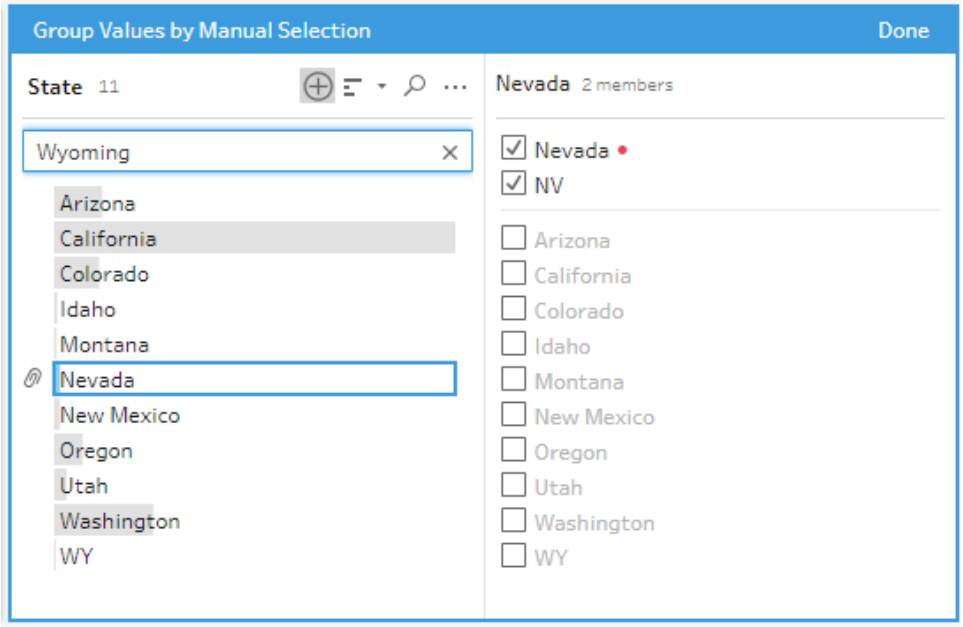
Some reasons why a value might not be in the data set include the following:

- 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.

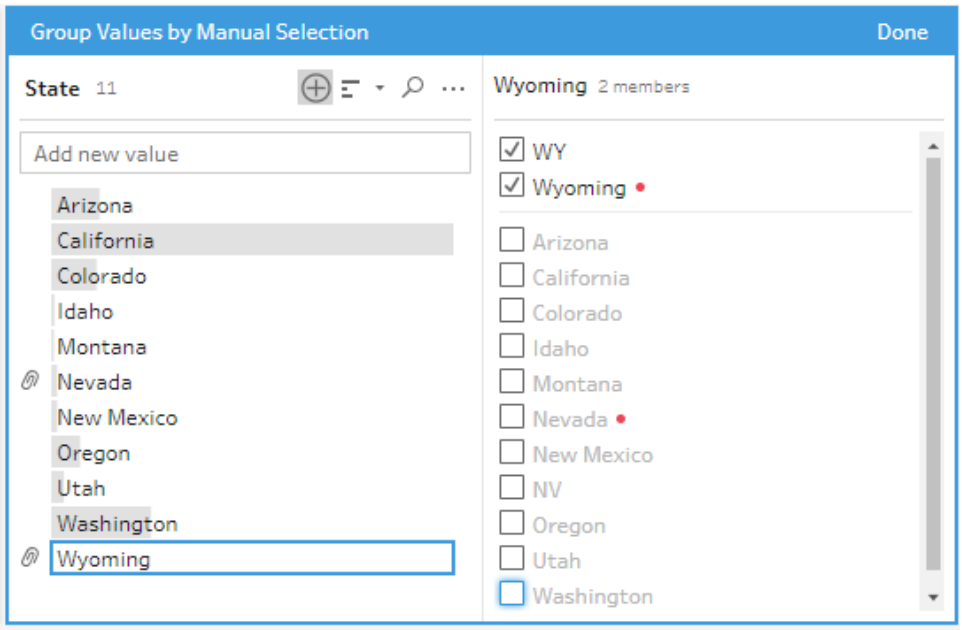
To add a new value:

1. In the **Profile** pane or Results pane, select the field you want to edit.
2. Click **More options** ··· and select **Group Values** (**Group and Replace** in previous versions) > **Manual Selection** from the context menu.
3. In the left pane of the **Group Values** 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 Values** editor.

Automatically map values to a standard value using 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.

If you use data roles to validate your field values, you can use the **Group Values** (**Group and Replace** in previous versions) option to match invalid values with valid ones. For more information, see **Group similar values by data role** on page 202

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. This option isn't available for data roles.
- **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. This option isn't available for data roles.

  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.

  Starting in Tableau Prep Builder version 2019.2.3 and on the web, this option is available to use after a data role is applied. In that case, it matches the invalid values to the closest valid value using the edit distance. If the standard value isn't in your data set sample, Tableau Prep adds it automatically and marks the value as not in the original data set.

- **Pronunciation +Spelling**: (Tableau Prep Builder version 2019.1.4 and later and on the web) If you assign a data role to your fields, you can use that data role to match and group values with the standard value defined by your data role. This option then matches invalid values to the most similar valid value based on spelling and pronunciation. If the standard value isn't in your data set sample, Tableau Prep adds it automatically and marks the value as not in the original data set. This option is most suitable for English words.

  For more information see **Clean and Shape Data** on page 227. Want to read more about these fuzzy match algorithms? See Automated Grouping in Tableau Prep Builder on Tableau.com

  > **Note**: In Tableau Prep Builder version 2019.1.4 and 2019.2.1 this option was labelled **Data Role Matches**.

Group similar values using fuzzy match

1. In the **Profile** pane or Results pane, select the field you want to edit.
2. Click **More options** ··· and select **Group Values** then select one of these options:
   - **Pronunciation**
   - **Common Characters**
   - **Spelling**

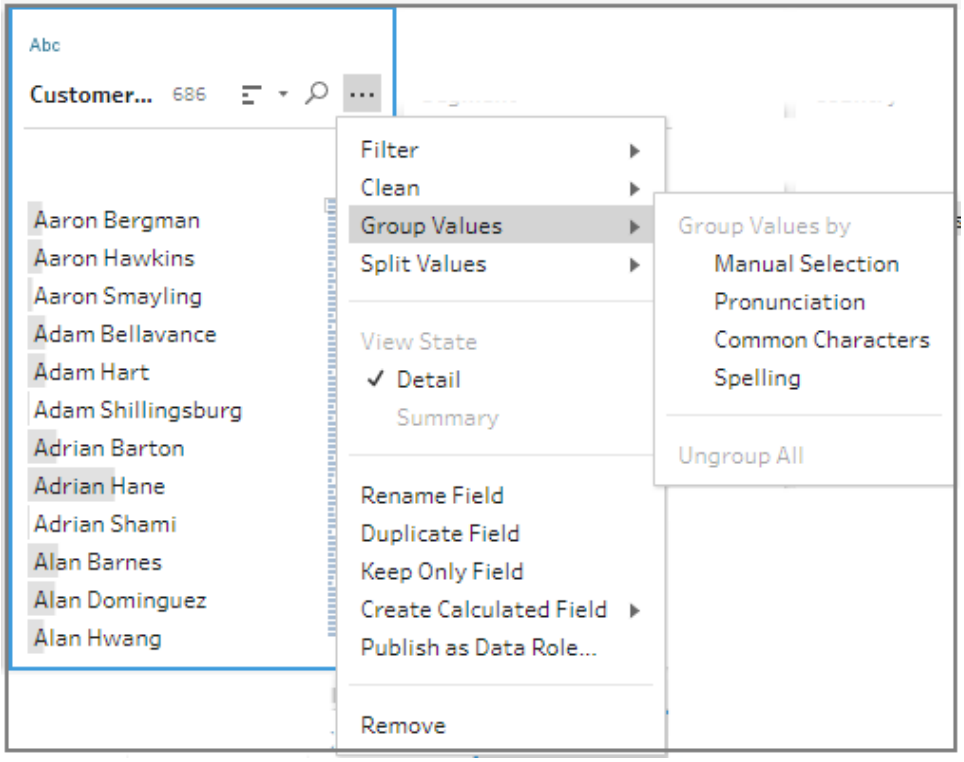
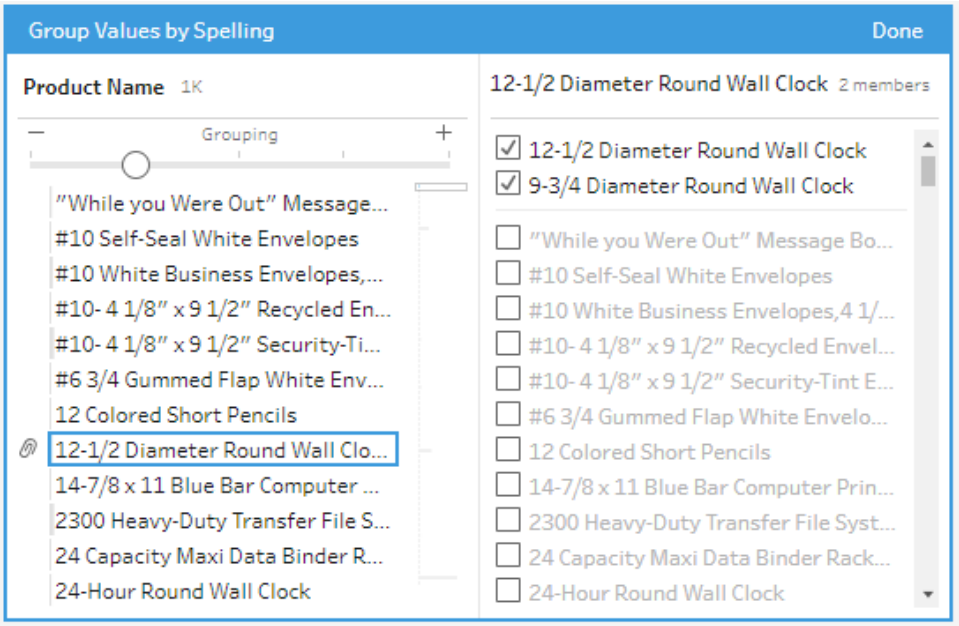


Tableau Prep Builder finds and groups values that match and replaces them with the value that occurs most frequently in the group.

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



Adjust your results when grouping field values

If you group similar values by **Spelling** or **Pronunciation**, you can change your results by using the slider on the field to adjust how strict the grouping parameters are.

Depending on how you set the slider, you can have more control over the number of values included in a group and the number of groups that get created. By default, Tableau Prep detects the optimal grouping setting and shows the slider in that position.

When you change the threshold, Tableau Prep analyses a sample of the values to determine the new grouping. The groups generated from the setting are saved and recorded in the **Changes** pane, but the threshold setting isn't saved. The next time the **Group Values** editor is opened, either from editing your existing change or making a new change, the threshold slider is shown in the default position, enabling you to make any adjustments based on your current data set.

1. In the **Profile** pane or **Results** pane, select the field you want to edit.

2. Click **More options** ··· and select **Group Values** (**Group and Replace** in previous versions) then select one of these options:

   - **Pronunciation**
   - **Spelling**

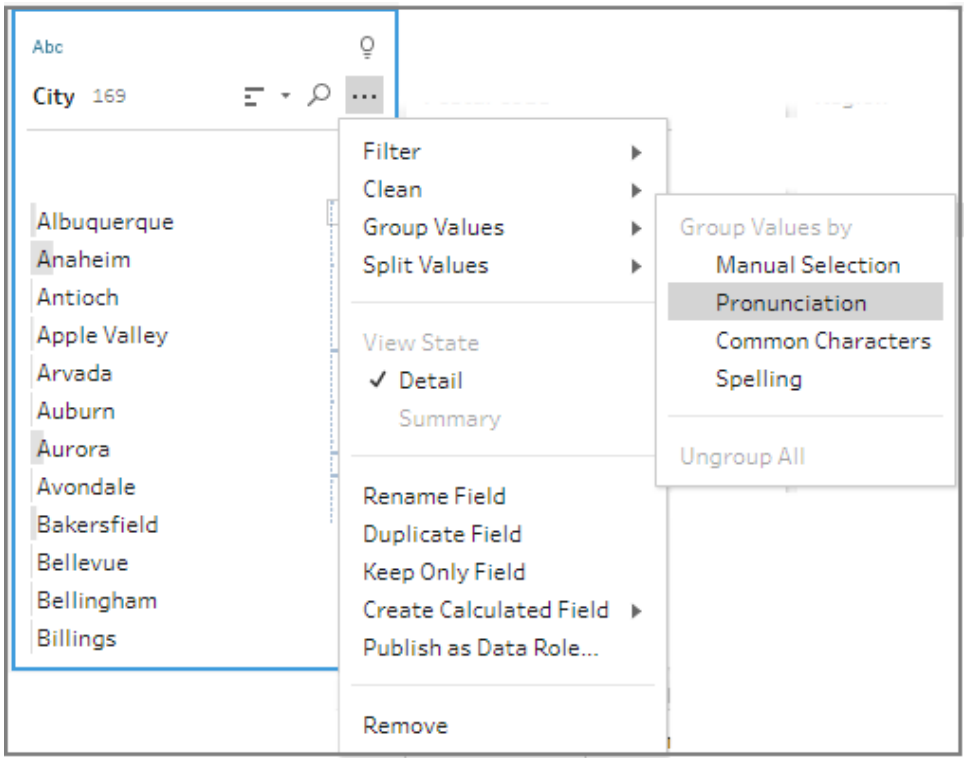
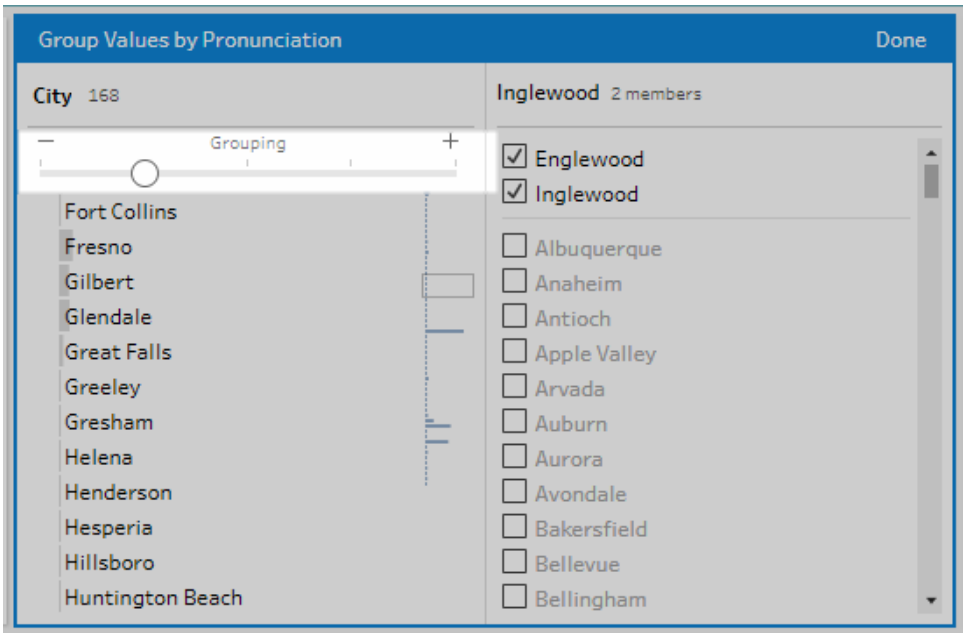


Tableau Prep finds and groups values that match and replaces them with the value that occurs most frequently in the group.

3. In the left pane of the **Group Values** editor, drag the slider to one of the 5 threshold levels to change your results.



To set a stricter threshold, move the slider to the left. This results in fewer matches and creates less groups. To set a looser threshold, move the slider to the right. This results in more matches and creates more groups.

4. Click **Done** to save your changes.

Copy steps, actions and fields

> **Note**: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

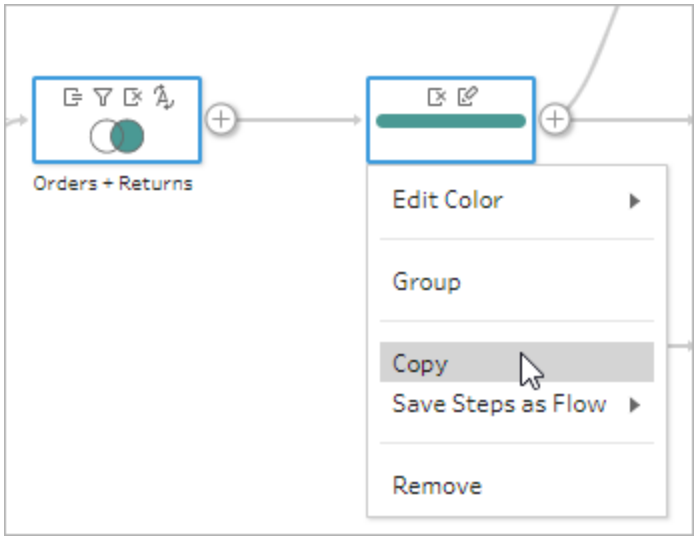
When cleaning your data you often perform the same cleaning operations or actions over and over throughout your flow. To help make cleaning and shaping your data more efficient, you can copy and paste operations or actions throughout your flow, or even copy selected steps or groups and save them so you can perform cleaning operations or actions once, then reuse it where you need it. You can even duplicate fields to experiment with different cleaning operations.

For more information about creating groups in your flow, see **Group steps** on page 151.

Copy and paste steps

Copy one or more steps to use them in another area of the same flow. This option is not available for Input steps that include a union in the input step.

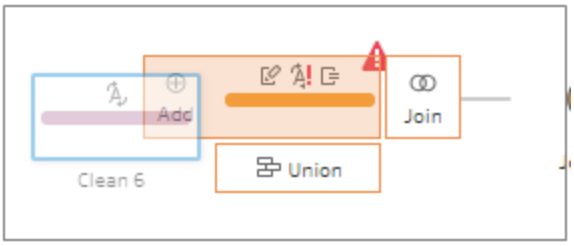
1. In the Flow pane, select one or more steps or group in the flow.
2. Right-click or Ctrl-click (MacOS) on the selected step, then select **Copy**.



3. To paste the copied steps, do one of the following:
   - Hover over a step or flow line until the plus icon appears, then click the icon and select **Paste** from the menu.
   - Right-click or Ctrl-click (MacOS) in any whitespace in the canvas and click **Paste**.

4. If you pasted the steps in the flow whitespace, drag and drop the steps to where you want to place them in the flow. If adding steps to the end of a flow step, the steps are automatically added to the end of the step. If inserting steps between existing flow steps, move the steps to where you want them in the flow and fix any errors.

   You can remove flow lines or move steps around if needed. For example, to connect a step to the copied steps, remove the existing flow line, if there is one, then drag the existing step to the new step and drop on **Add**.



   For more information about organising your flow, see **Reorganise the layout of your flow** on page 160.

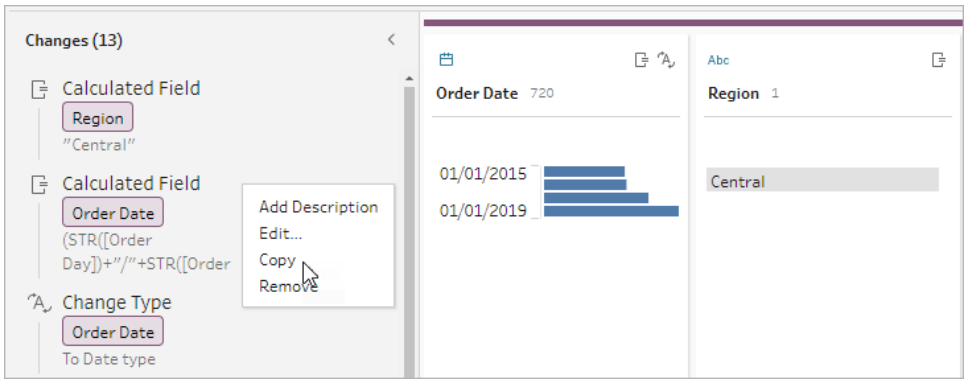
Copy and paste cleaning operations

You can copy and paste cleaning operations in the same flow to reuse your actions using one of the following options:

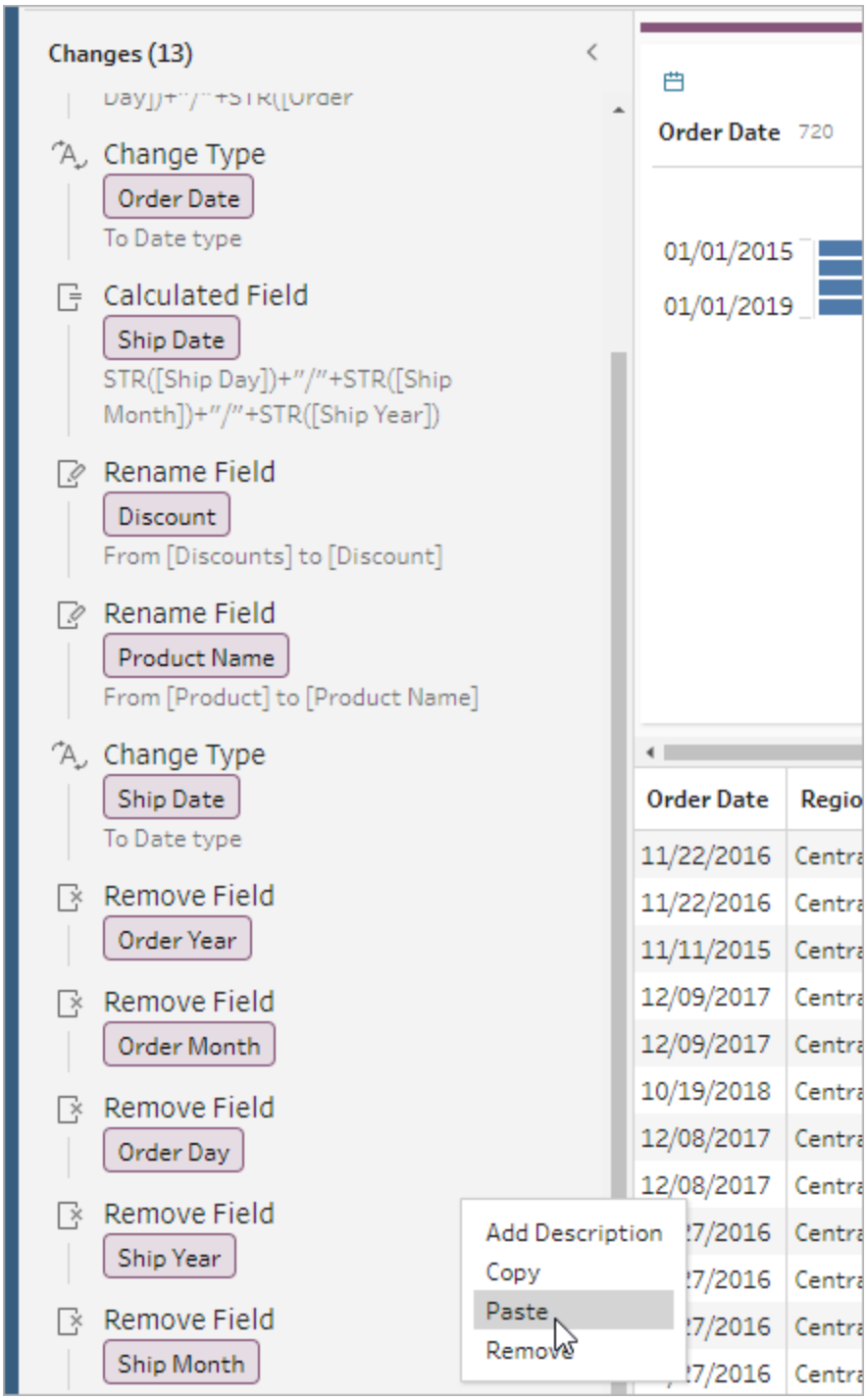
- Copy an operation from the **Changes** pane in one step and paste it in the **Changes** pane for the same step or another step to apply that same operation in that step.
- Drag and drop an operation from the **Changes** pane and drop it to other fields in the Profile pane for that step to apply that operation to multiple fields. This option is not available for operations that impact multiple fields, such as calculated fields.

To copy and paste a change in a step to the same step or another step, do the following:

1. In the **Changes** pane select the change you want to copy.
2. Right-click or Ctrl-click (MacOS) on the change item, then select **Copy** from the menu.

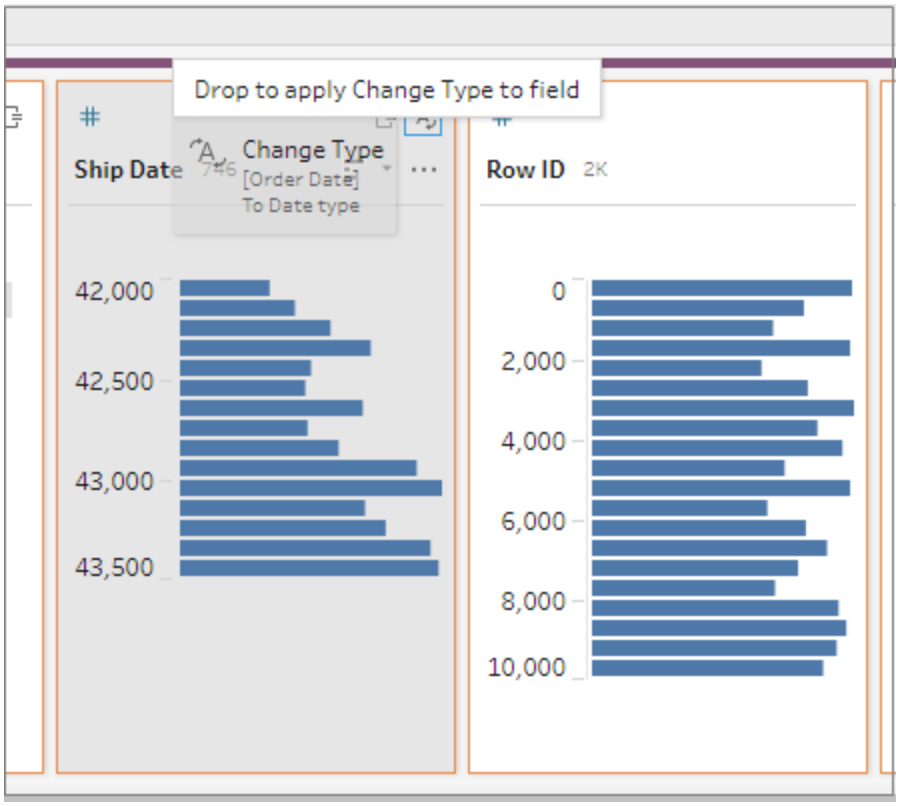


3. In the **Changes** pane where you want to past the change right-click or Ctrl-click (MacOS) and select **Paste**. Select the change and click on **Edit** to make any adjustments as needed.



To drag and drop a change to other fields in the step do the following:

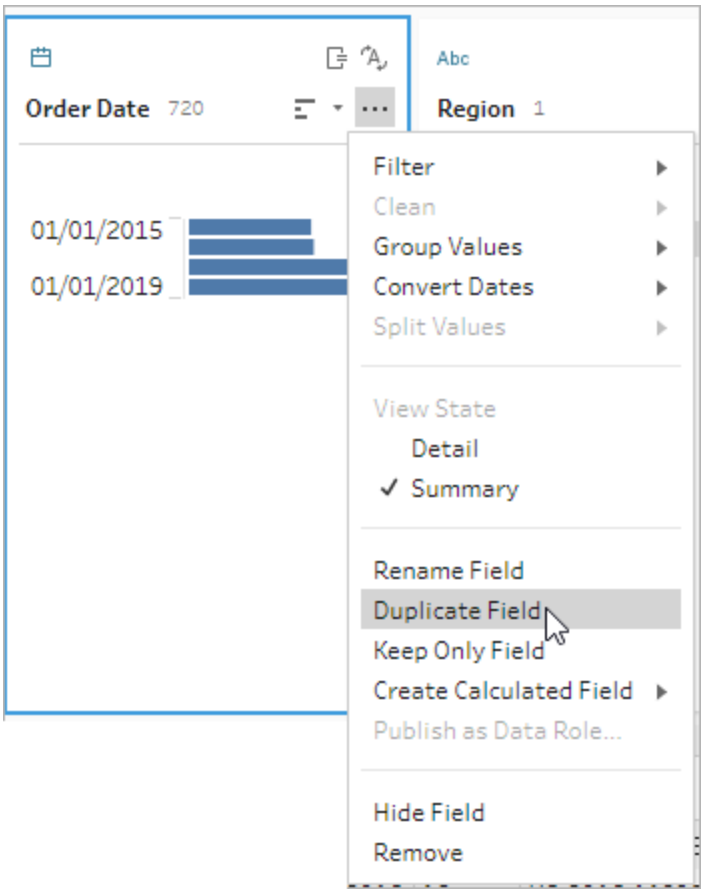
1. In the **Changes** pane select the change you want to copy.
2. Drag the change over the field where you want to apply it and drop it. Repeat this action as needed.



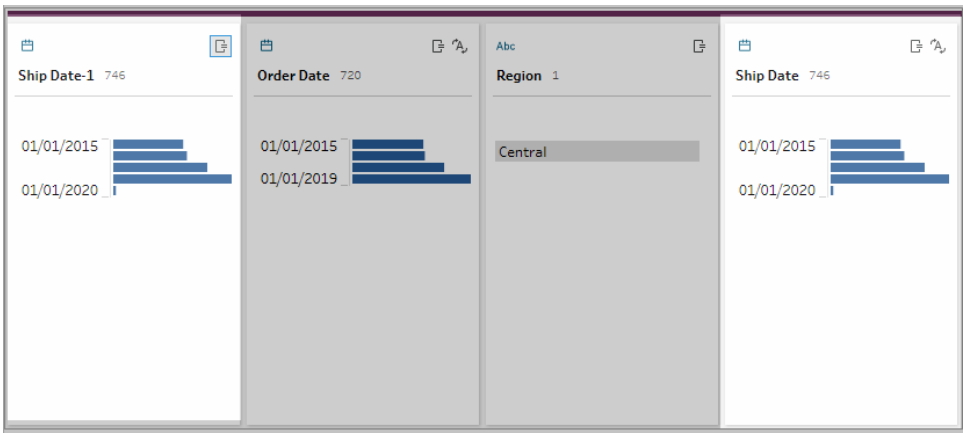
Copy fields

Starting in Tableau Prep Builder version 2019.2.3 and later and on the web, if you wanted to experiment with your cleaning operations on a field but don't want to change the original data, you can copy your fields .

1. In the profile pane, data grid, results pane or list view, select the field you want to copy.
2. From the **More options** menu, select **Duplicate Field**.



A new field is created with the same name and a modifier. For example, "Ship Date -1".



Create reusable flow steps

Supported in Tableau Prep Builder version 2019.3.2 and later.

> **Note**: Reusable flow steps can't be created on the web, but you can use them in your web flows. Reusable steps that include file-based input steps are not yet supported on the web.

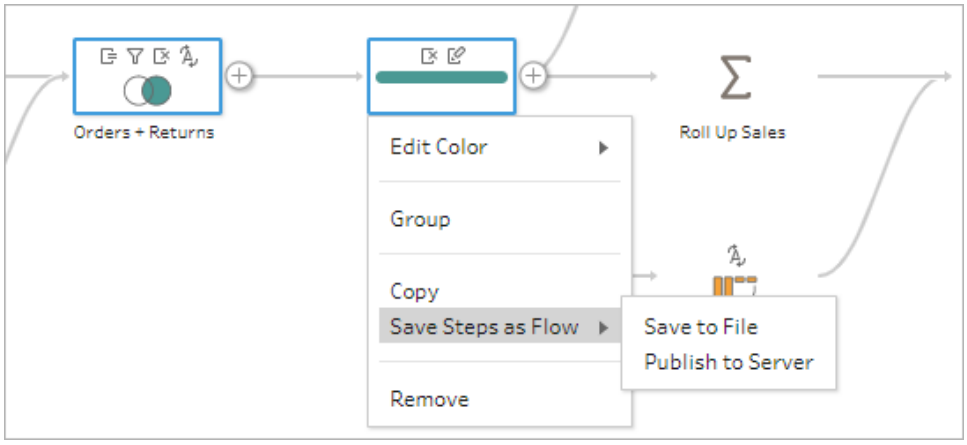
If you commonly perform the same actions over and over again with your data and you want to apply these same steps in other flows, in Tableau Prep Builder version 2019.3.2 and later, you can select one or more flow steps or groups and their associated actions or the entire flow and save it locally to a file on your computer. You can also or publish it to Tableau Server or Tableau Cloud to share with others.

When the flow steps are published to your server, a **Saved Steps** tag is automatically added so you can easily search and find them when adding them to your flows.

Starting in version 2022.1.1, you can create reusable steps that include parameters. When the steps are saved, the parameter is converted to a static value using the parameter's **Current value**. For more information about using parameters in flows, see **Create and Use Parameters in Flows** on page 205.

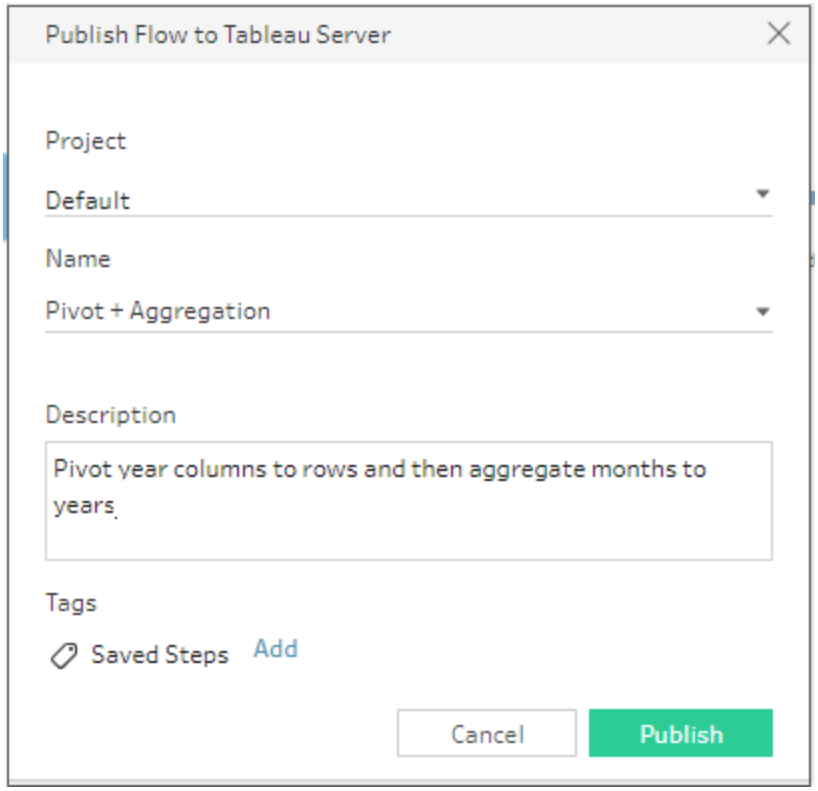
Create reusable steps

1. Select one or more steps.
2. Right-click or Ctrl-click (MacOS) on a selected step and select **Save Steps as Flow**.



3. Select **Save to File** to save the flow locally or **Publish to Server** to publish the flow to Tableau Server or Tableau Cloud.

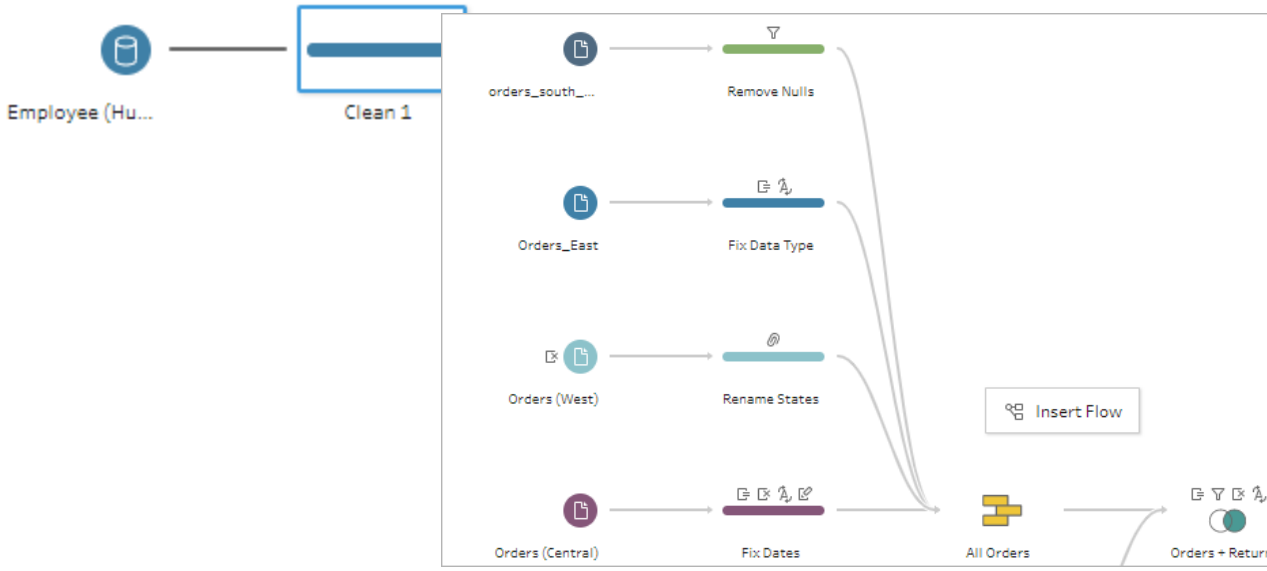
4. If you publish the flow to Tableau Server or Tableau Cloud, sign in to your server if needed, then complete the fields in the **Publish Flow** dialog then click **Publish**.



Insert reusable steps in a flow

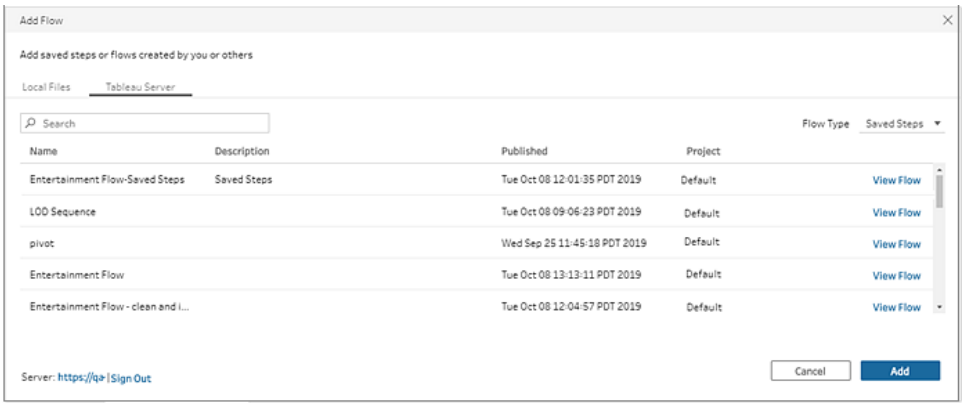
1. Open a flow.
2. In the flow pane, do one of the following:
   - Hover over a step or flow line until the plus icon appears, then click the icon and select **Insert Flow**.
   - In the white area of the canvas, right-click or Ctrl-click (MacOS) and click **Insert Flow** or click **Edit** > **Insert Flow** from the top menu.

Flow Step menu **Canvas menu**

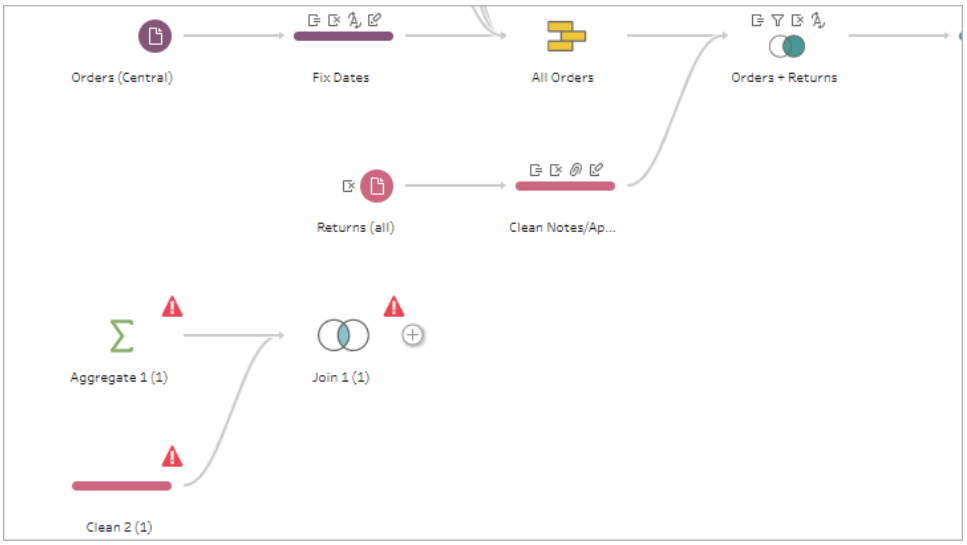


3. In the **Add Flow** dialog, select from flows saved to either your local file or your server then click **Add**. The list of flows is automatically filtered to show flows tagged with **Saved Steps**. To insert other flows, change the **Flow Type** to **All Flows**.

   In Tableau Prep Builder version 2019.4.2 and later and on the web, you can click **View Flow** to open and view the published flow in the server you are signed into.



4. The flow is added to the flow pane. If adding a flow to the end of a flow step, the flow steps are automatically added to the end of the step. If inserting flow steps between existing flow steps, move the steps where you want them in the flow and fix any errors.



Fill Gaps in Sequential Data

Supported in Tableau Prep Builder version 2021.3.1 and later and on the web in Tableau Server and Tableau Cloud version 2021.3.0 and later.

When you have gaps in your sequential data set, you may need to fill those gaps with new rows to effectively analyse your data or perform trend analysis. You can use the **New Rows** step type to generate the missing rows and set configuration options to get the results you need.

New rows can be generated for fields with numeric (whole numbers) or date values. Configuration options include:

- Generate rows using values from a single field or two fields
- Use all data in the field or select a range of values
- Create a new field with the results or add the new rows to your existing fields
- Set the increment (up to 10,000) to use when generating the new rows
- Set the values for the new rows to zero or Null, or copy the value from the previous row.

Examples

- **Example 1**: You have a table of sales data, but there are some days where no sales are recorded. You need a row for every day, not just the days where you had sales. With **New Rows**, you can generate rows for your missing days and add them to you existing field "Days of the week". Since no sales are recorded for those days, you want the quantity sold value to be zero.

Original Table

Day of the week	Quantity Sold
1	100
3	23
5	75
7	14

Table with missing rows added

Day of the week	Quantity Sold
1	100
2	0
3	23
4	0
5	75
6	0
7	14

- **Example 2**: You have a table of sales data where orders filled is recorded using a range of dates. You need a row for each day. Since you don't know how many orders were filled each day, you want the values for the new rows to be Null. With **New Rows**, you can generate the missing rows between the two dates and create a new field called "All Days" to preserve your original data.

Original Table

Start Date	End Date	Orders Filled
1/1/2020	1/3/2020	100
1/4/2020	1/7/2020	75

Table with missing rows added

Start Date	End Date	All Dates	Orders Filled
1/1/2020	1/3/2020	1/1/2020	100
1/1/2020	1/3/2020	1/2/2020	Null
1/1/2020	1/3/2020	1/3/2020	Null
1/4/2020	1/7/2020	1/4/2020	75
1/4/2020	1/7/2020	1/5/2020	Null
1/4/2020	1/7/2020	1/6/2020	Null
1/4/2020	1/7/2020	1/7/2020	Null

Generate new rows

1. In the **Flow** pane, click the plus icon, and select **New Rows**. A **New Rows** step displays in the **Flow** pane.

   Complete the following steps to configure your options to generate the new rows.
2. **How do you want to add new rows?** Use one of the following options to select the field or fields where rows are missing.

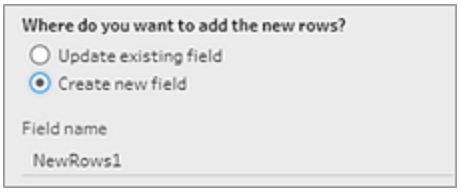
   a. **Values from one field**: Generate missing rows from values in a single field. Use this option for **Number (whole)** or **Date** data types.

By default, use the minimum and maximum value to generate missing rows. This option uses all values in the field. If you only want to use a range of values to generate the missing rows, set a **Start value** and **End value**.

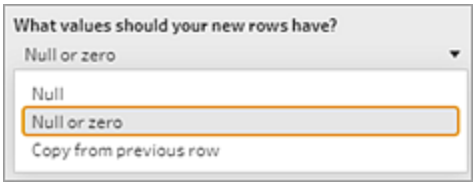
> **Note**: The **Start value** and **End value** fields can't be used to generate rows outside of your current data set.

b. **Value ranges from two fields**: Generate new rows using a value range between two date fields. This option is only available for **Date** data types, uses all values in the field and requires that both fields have the same data type.

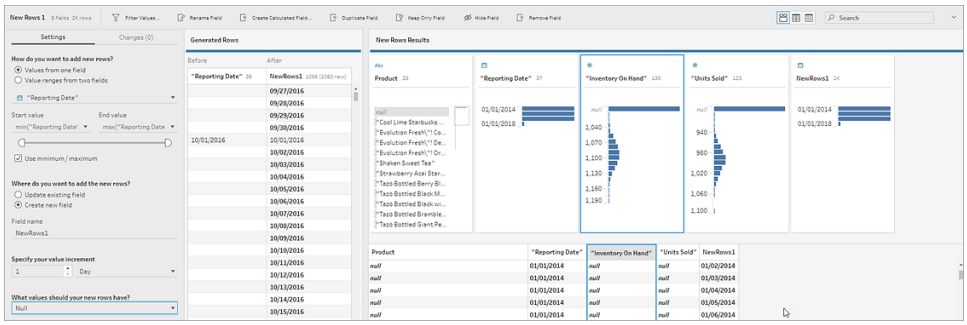
3. **Where do you want to add the new rows?** When using a single field, you can add the new rows to your existing field or create a new field to preserve your original data. When using value ranges from two fields, you must create a new field.
   - **Field Name**: Enter a name for the new field.



4. **Specify your increment value**: Enter a value from 1-10,000. Each new row is incremented by the value you select. If you select a value that is greater than the gap between values, no new rows are generated.
   - **Number fields**: Select a numeric value.
   - **Date fields**: Select a numeric value and select **Day**, **Week** or **Month**.
5. **What values should your new rows have?**: Select an option to fill in the other field values for the new rows.
   - **Null**: Populate all field values with Null.
   - **Null or zero**: Populate all text values with Null and all numeric values with zero.
   - **Copy from previous row**: Populate all field values with the value from the previous row.



New rows are shown in the **Generated Rows** pane in **bold** as you enter your configuration settings. The row details are shown in the **New Rows Results** pane.



Create Level of Detail, Rank and Tile Calculations

> **Note**: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

You can use calculated fields to create new data using data that already exists in your data source. Tableau Prep supports many of the same calculation types as Tableau Desktop. For general information about creating calculations, see Get Started with Calculations in Tableau.

> **Note**: When creating a calculated field, don't use the reserved name ‘Number of Records’. Using this reserved name will result in a permission-related error.

Starting in version 2020.1.3 Tableau Prep Builder and on the web, you can use FIXED Level of Detail (LOD) and RANK and ROW_NUMBER analytic functions to perform more complex calculations.

For example, add a FIXED LOD calculation to change the granularity of fields in your table, use the ROW_NUMBER () analytic function to quickly find duplicate rows or use one of the RANK () functions to find the top N or bottom N values for a selection of rows with similar data. If you want a more guided experience when building these types of expressions, you can use the visual calculation editor.

Starting in version 2021.4.1 Tableau Prep Builder and on the web, you can use the tile feature to distribute rows into a specified number of buckets.

> **Note**: Some functions supported in Tableau Desktop may not yet be supported in Tableau Prep. To view the available functions for Tableau Prep, review the function list in the Calculation editor.

Calculate level of detail

When you need to calculate data at multiple levels of granularity in the same table, you can write a level of Detail (LOD) expression to do this. For example, if you wanted to find the total sales for each region, you could write a calculation like `{FIXED [Region] : SUM([Sales])}`.

Tableau Prep supports the FIXED level of detail expression and uses the syntax `{FIXED [Field1],[Field2] : Aggregation([Field)}`.

LOD expressions have two parts to the equation that are separated by a colon.

- **FIXED [Field]** (required): This is the field or fields that you want to calculate the values for. For example if you wanted to find the total sales for customer and region, you would enter `FIXED [Customer ID], [Region]:`. If you don't select any fields, this is the equivalent to performing the aggregation defined on the right side of the colon and repeating that value for every row.
- **Aggregation ([Field])** (required): Select what you want to calculate and what level of aggregation you want. For example if you want to find the total sales, then enter `SUM ([Sales]`.

When using this feature in Tableau Prep, the following requirements apply:

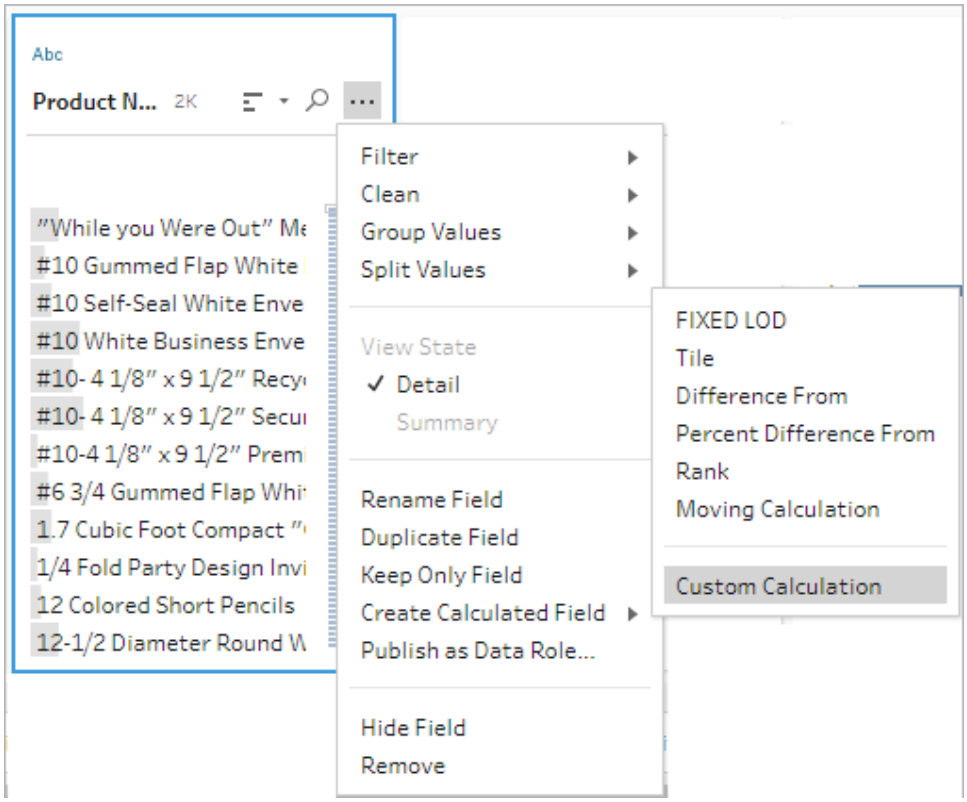
- `INCLUDE` and `EXCLUDE` LOD expressions aren't supported.
- Aggregation calculations are only supported inside an LOD expression. For example, `SUM([Sales])` would not be valid, but `{FIXED [Region] : SUM([Sales])}` is valid.
- Nesting expressions inside an LOD expression isn't supported. For example, `{ FIXED [Region] : AVG( [Sales] ) / SUM( [Profit] )}` isn't supported.
- Combining an LOD expression with another expression isn't supported. For example `[Sales]/{ FIXED [Country / Region]:SUM([Sales])}` isn't supported.

Create Level of Detail (LOD) calculations

To create a level of detail calculation, you can use the Calculation editor to write the calculation yourself or if you want a more guided experience, you can use the Visual Calculation editor where you select your fields and Tableau Prep writes the calculation for you.

Calculation editor

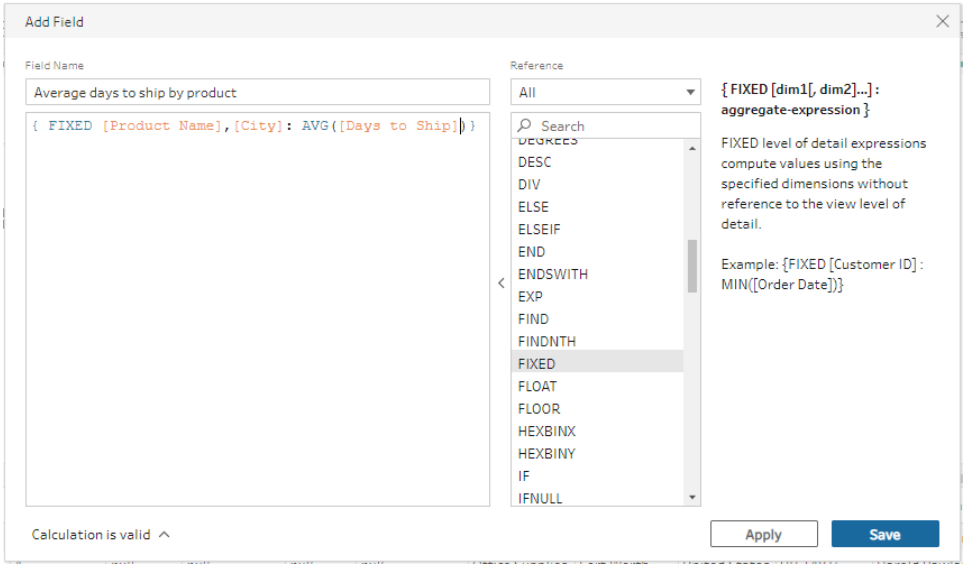
1. In the Profile pane toolbar, click **Create Calculated Field**, or in a profile card or data grid, click the **More options** ··· menu and select **Create Calculated Field** > **Custom Calculation**.



2. In the Calculation editor, enter a name for your calculation and enter the expression.

   For example, to find the average days to ship products by city, create a calculation

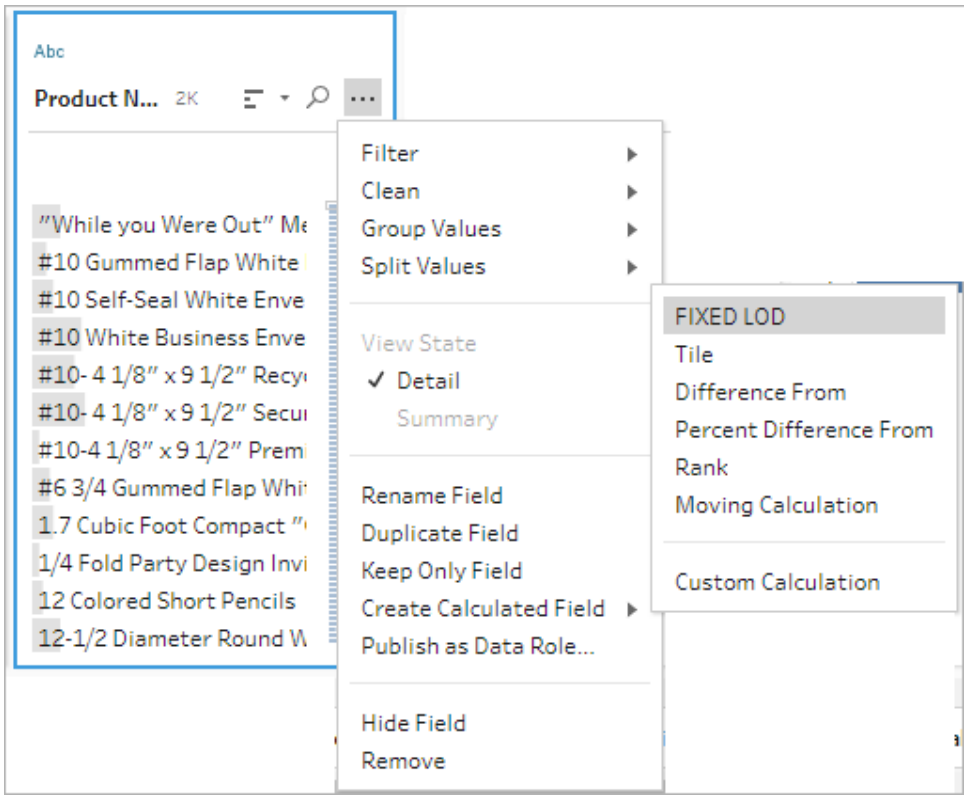
like the one shown below.



Visual calculation editor

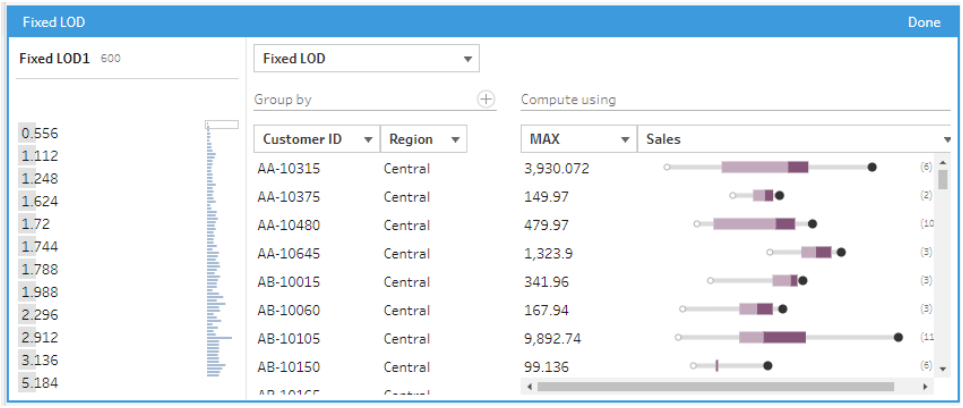
Select fields from a list and Tableau Prep builds the calculation for you as you make your selections. A preview of the results is shown in the left pane so you can see the results of your selections as you go.

1. In a profile card or results pane, click the **More options** ··· menu and select **Create Calculated Field** >**Fixed LOD**.



2. In the Visual Calculation editor, do the following:
   - In the **Group by** section, select the fields that you want to calculate the values for. The field where you selected the **Create Calculated Field** >**Fixed LOD** menu option is added by default. Click the plus ⊕ icon to add any additional fields to your calculation. This populates the left side of the equation, `{FIXED [Field1],[Field2] :.`
   - In the **Compute using** section, select the field that you want to use to calculate your new values. Then select your aggregation. This populates the right side of the equation, `Aggregation([Field)}.`

     A graphic below the field shows the distribution of values and a total count for each value combination. Depending on the type of data, this can be a box plot, range of values or the actual values.



> **Note**: Available aggregation values vary by the data type assigned to the field.

- To remove a field, right-click or Cntrl-click (MacOS) in the drop-down box for the fields in the **Group by** section and select **Remove Field**.
- In the left pane, double-click in the field header and enter a name for your calculation.

3. Click **Done** to add your new calculated field. In the **Changes** pane, you can see the calculation that Tableau Prep generated. Click **Edit** to open the visual calculation editor to make any changes.

Calculate rank or row number

Analytic functions, sometimes referred to as window calculations, enable you to perform calculations across the entire table or a selection of rows (partition) in your data set. For example, when applying a rank to a selection of rows, you would use the following calculation syntax:

```
{PARTITION [field]: {ORDERBY [field]: RANK() }}
```

- **PARTITION** (optional): Designate the rows you want to perform the calculation on. You can specify more than one field, but if you want to use the entire table, omit this part of the

function and Tableau Prep treats all the rows as the partition. For example `{ORDERBY [Sales] : RANK() }`.

- **ORDERBY** (required): Specify one or more fields that you want to use to generate the sequence for the rank.
- **Rank ()** (required): Specify the rank type or ROW_NUMBER () you want to calculate. Tableau Prep supports RANK(), RANK_DENSE(), RANK_MODIFIED(), RANK_PERCENTILE() and ROW_NUMBER() functions.
- **DESC** or **ASC** (optional): Represents descending (DESC) or ascending (ASC) order. By default, rank is sorted in descending order, so you don't need to specify this in the expression. If you want to change the sort order, add **ASC** to the expression.

  You can also include both options in the function. For example, if you wanted to rank a selection of rows, but wanted to sort the rows in ascending order, then apply the rank in descending order, you would include these two options in the expression. For example: `{PARTITION [Country], [State]: {ORDERBY [Sales] ASC,[Customer Name] DESC: RANK() }}`

When using this feature, the following requirements apply:

- Nesting expressions inside a RANK () function isn't supported. For example, `[Sales]/{PARTITION [Country]: {ORDERBY [Sales]: RANK() }} / SUM( [Profit] )}` isn't supported.
- Combining a RANK () function with another expression isn't supported. For example `[Sales]/{PARTITION [Country]: {ORDERBY [Sales]: RANK() }}` isn't supported.

Supported analytic functions

Function	Description	Result
RANK ()	Assigns a whole number rank starting with 1, in ascending or descending order to each row. If rows have the same value, they	<table><tr><th>RANK</th><th>Sales Person</th><th>Region</th><th>Commission</th></tr><tr><td>1</td><td>Lisa</td><td>Central</td><td>15,000</td></tr><tr><td>1</td><td>Sarah</td><td>Central</td><td>15,000</td></tr><tr><td>3</td><td>Barb</td><td>East</td><td>13,000</td></tr><tr><td>4</td><td>Charles</td><td>East</td><td>9,000</td></tr><tr><td>5</td><td>Jorge</td><td>Central</td><td>7,000</td></tr></table>

<table>
<tr><td></td><td>share the rank that is assigned to the first instance of the value. The number of rows with the same rank is added when calculating the rank for the next row, so you may not get consecutive rank values.

Sample calculation: <code>{ORDERBY [Commission] DESC: RANK()}</code></td><td></td></tr>
<tr><td>RANK_DENSE()</td><td>Assigns a whole number rank starting with 1 in ascending or descending order to each row. If rows have the same value, they share the rank that is assigned to the first instance of the value, but no rank values are skipped so you will see consecutive rank values.</td><td><table><tr><th>DENSE RANK</th><th>Sales Person</th><th>Region</th><th>Commission</th></tr><tr><td>1</td><td>Lisa</td><td>Central</td><td>15,000</td></tr><tr><td>1</td><td>Sarah</td><td>Central</td><td>15,000</td></tr><tr><td>2</td><td>Barb</td><td>East</td><td>13,000</td></tr><tr><td>3</td><td>Charles</td><td>East</td><td>9,000</td></tr><tr><td>4</td><td>Jorge</td><td>Central</td><td>7,000</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td>Sample calculation:
<code>{ORDERBY [Commission] DESC: RANK_DENSE()}</code></td><td></td></tr>
<tr><td>RANK_MODIFIED()</td><td>Assigns a whole number rank starting with 1, in ascending or descending order to each row. If rows have the same value, they share the rank that is assigned to the last instance of the value. Rank_Modified is calculated as <code>Rank + (Rank + Number of duplicate rows - 1)</code>.
Sample calculation:
<code>{ORDERBY [Commission] DESC: RANK_MODIFIED()}</code></td><td><table><tr><th>RANK MODIFIED</th><th>Sales Person</th><th>Region</th><th>Commission</th></tr><tr><td>1</td><td>Sarah</td><td>Central</td><td>15,000</td></tr><tr><td>1</td><td>Lisa</td><td>Central</td><td>15,000</td></tr><tr><td>3</td><td>Barb</td><td>East</td><td>13,000</td></tr><tr><td>4</td><td>Charles</td><td>East</td><td>9,000</td></tr><tr><td>5</td><td>Jorge</td><td>Central</td><td>7,000</td></tr></table></td></tr>
</table>

<table>
<tr>
<td>RANK_ PERCENTILE ()</td>
<td>Assigns a percentile rank from 0 to 1 in ascending or descending order to each row. RANK_ PERCENTILE is calculated as <code>(Rank-1)/(Total rows-1)</code>.

Sample calculation: <code>{ORDERBY [Commission] DESC: RANK_ PERCENTILE()}</code>

Note: In the event of a tie, Tableau Prep rounds the rank down, similar to PERCENT_ RANK() in SQL.</td>
<td>
<table>
<tr><th>RANK PERCENTILE</th><th>Sales Person</th><th>Region</th><th>Commission</th></tr>
<tr><td>0</td><td>Sarah</td><td>Central</td><td>15,000</td></tr>
<tr><td>0</td><td>Lisa</td><td>Central</td><td>15,000</td></tr>
<tr><td>0.5</td><td>Barb</td><td>East</td><td>13,000</td></tr>
<tr><td>0.75</td><td>Charles</td><td>East</td><td>9,000</td></tr>
<tr><td>1</td><td>Jorge</td><td>Central</td><td>7,000</td></tr>
</table>
</td>
</tr>
</table>

ROW_ NUMBER()	Assigns a sequential row ID to each unique row. No row number values are skipped. If you have duplicate rows and use this calculation, your results might change each time you run the flow if the order of rows changes. **Sample calculation**: `{ORDERBY [Commission] DESC: ROW_ NUMBER()}`	

ROW NUMBER	Sales Person	Region	Commission
1	Sarah	Central	15,000
2	Lisa	Central	15,000
3	Barb	East	13,000
4	Charles	East	9,000
5	Jorge	Central	7,000

The following example shows a comparison of each of the above functions applied to the same data set.

RANK	DENSE RANK	RANK MODIFIED	RANK PERCENTILE	ROW NUMBER	Sales Person	Region	Commission
1	1	1	0	1	Lisa	Central	15,000
1	1	1	0	2	Sarah	Central	15,000
3	2	3	0.5	3	Barb	East	13,000
4	3	4	0.75	4	Charles	East	9,000
5	4	5	1	5	Jorge	Central	7,000

Create Rank or Row Number calculations

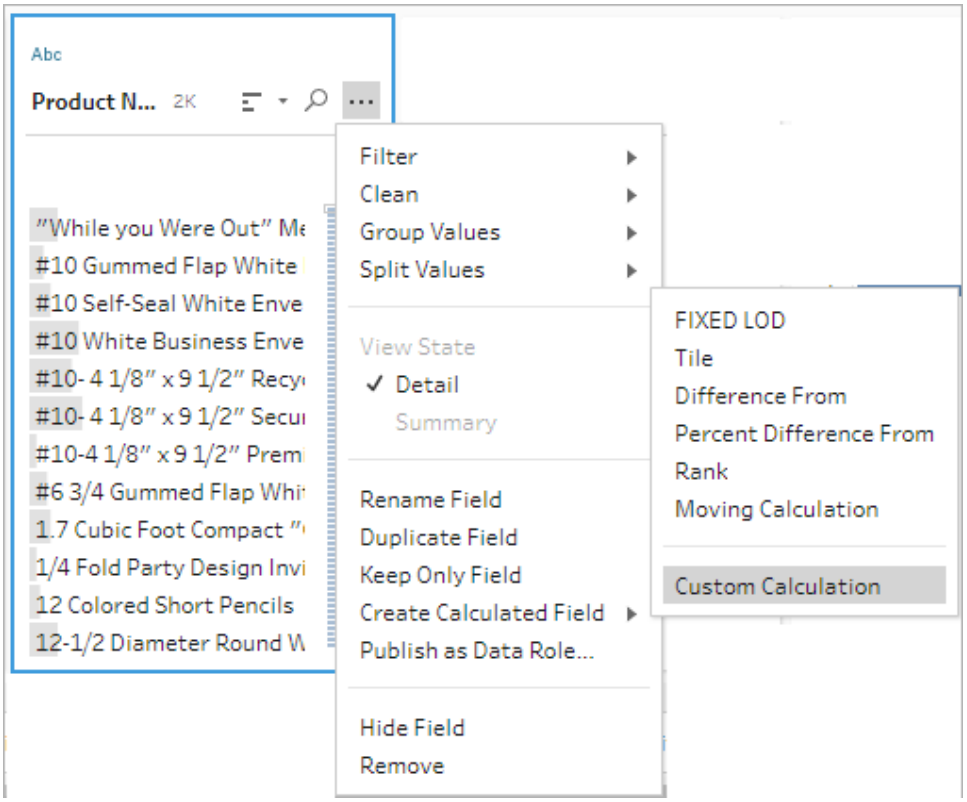
To create a Rank or Row_Number calculations, you can use the Calculation editor to write the calculation yourself or if you want a more guided experience, you can use the Visual Calculation editor where you select your fields and Tableau Prep writes the calculation for you.

Note: ROW_NUMBER () calculations aren't available in the visual calculation editor.

Calculation editor

Use the Calculation editor to create any of the supported RANK () or ROW_NUMBER() calculations. The list of supported analytic calculations is shown in the Calculation editor in the **Reference** drop-down under **Analytic**.

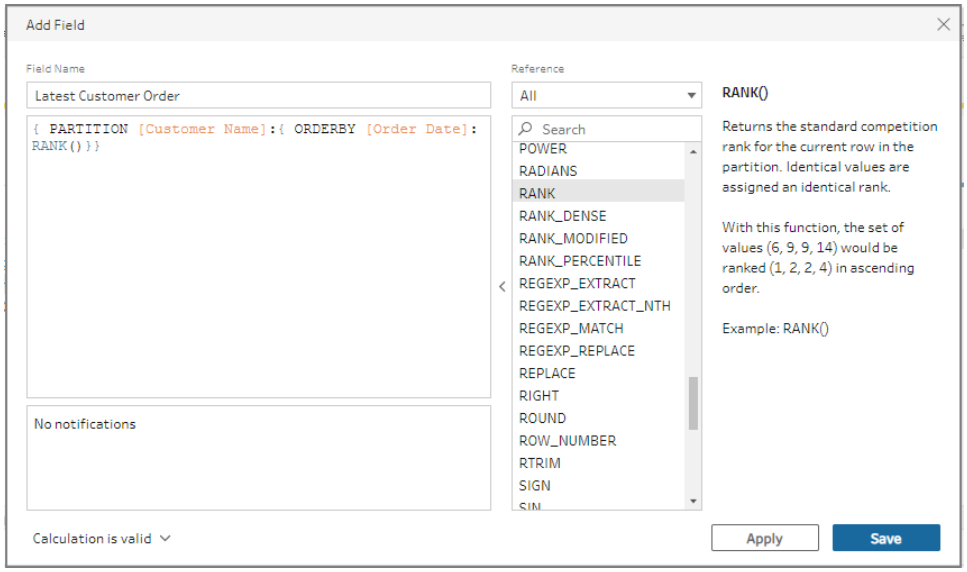
1. In the Profile pane toolbar, click **Create Calculated Field**, or in a profile card or data grid, click the **More options** ··· menu and select **Create Calculated Field** > **Custom Calculation**.



2. In the Calculation editor, enter a name for your calculation and enter the expression.

   For example to find the latest customer order, create a calculation like the one shown

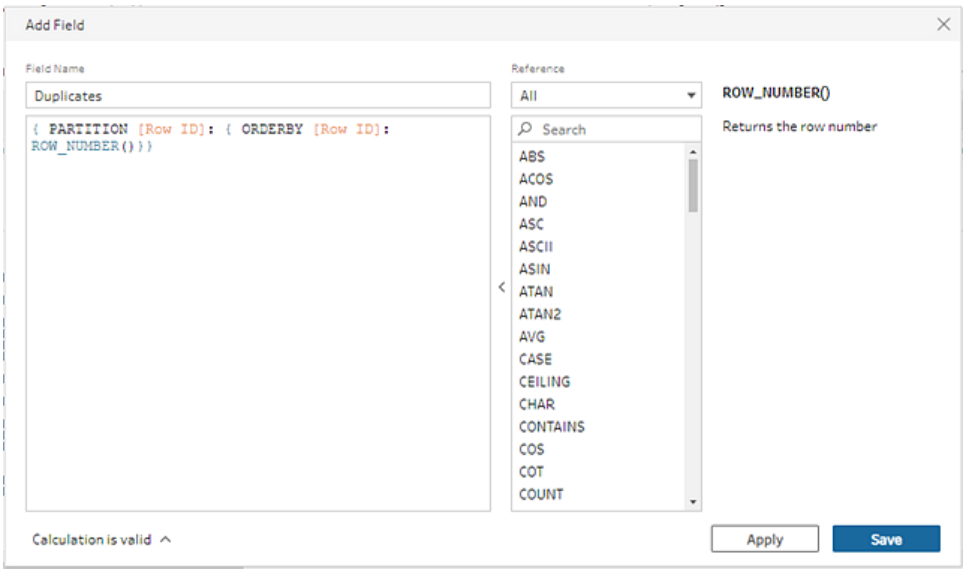
below, then keep only the customer order rows that are ranked with the number **1**.



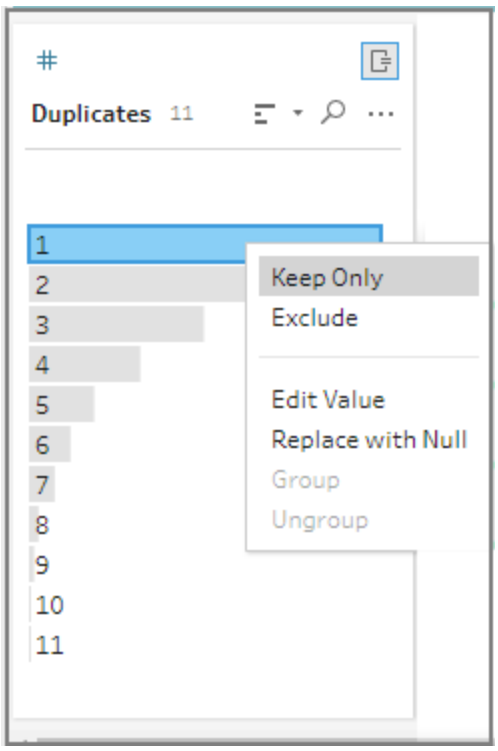
Example: Use ROW_NUMBER to find and remove duplicate values.

This example uses the Superstore sample data set in Tableau Prep Builder to find and remove exact duplicate values for the field **Row ID** using the ROW_NUMBER function.

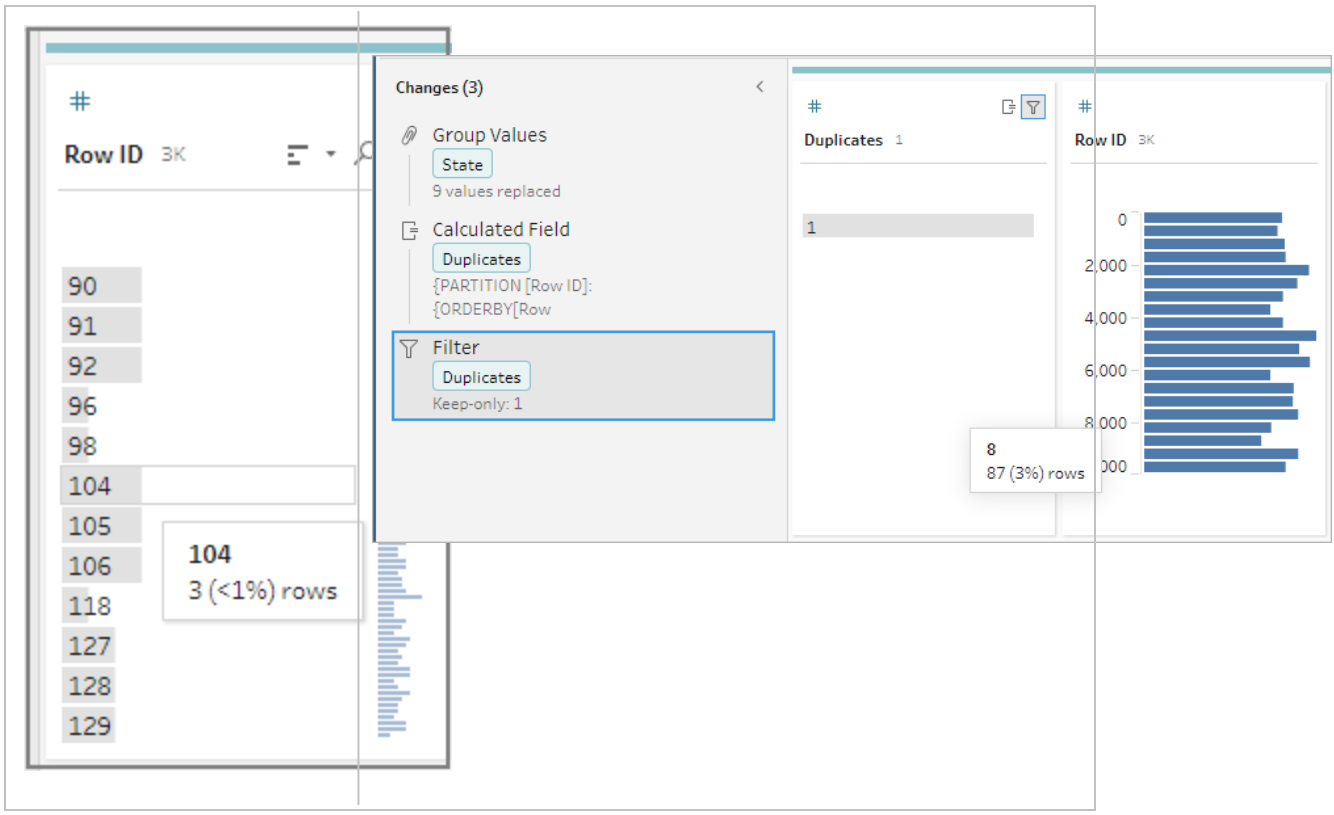
1. Open the **Sample Superstore** flow.
2. In the Flow pane, for the Input step **Orders West**, click on the Clean step **Rename States**.
3. In the toolbar, click **Create Calculated Field**.
4. In the Calculation editor, name the new field "Duplicates", and use the `ROW_NUMBER` function to add a row number to the field **Row ID** using the expression `{PARTITION [Row ID]: {ORDERBY[Row ID]:ROW_NUMBER()}}` and click **Save**.



5. In the new calculated field, right-click or Cmd-click (MacOS) on the field value **1**, then select **Keep Only** from the menu.



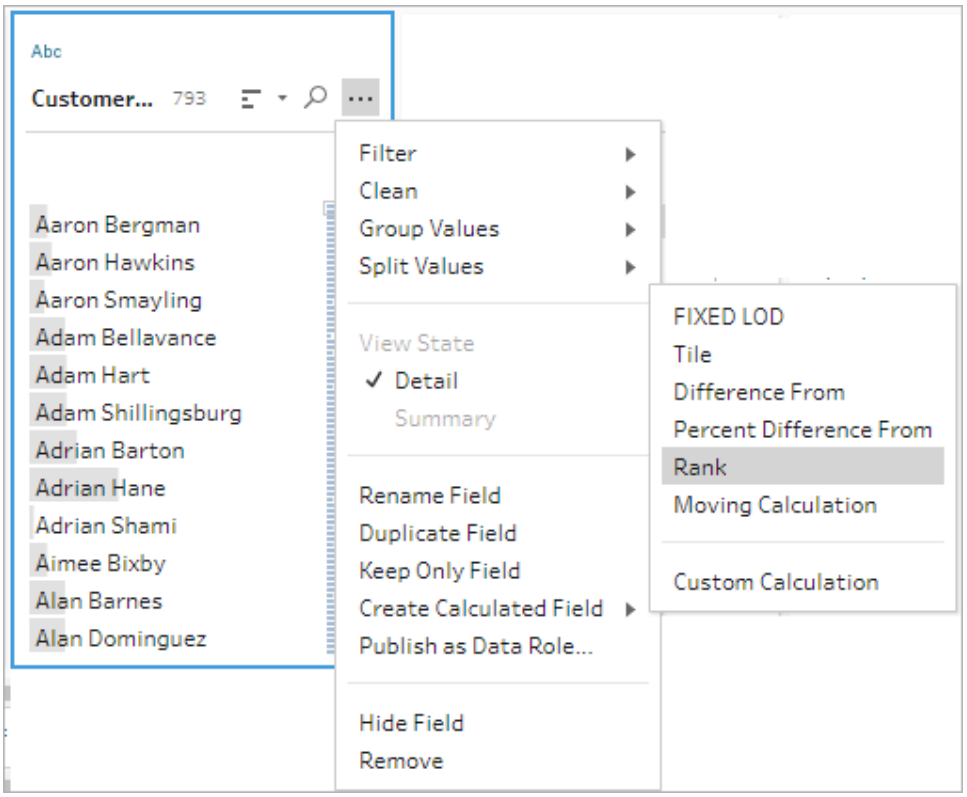
Before	After



Visual Calculation editor

Just like when creating a level of detail calculation, you can use the visual calculation editor to build a rank calculation. Select the fields you want to include in the calculation, then select the fields you want to use to rank the rows and the type of rank you want to calculate. A preview of the results is shown in the left pane so you can see the results of your selections as you go.

1. In a profile card or results pane, click the **More options** ··· menu and select **Create Calculated Field** >**Rank**.



2. In the Visual calculation editor, do the following:

   - In the **Group by** section, select the fields with rows you want to compute values for. This creates the Partition part of the calculation.

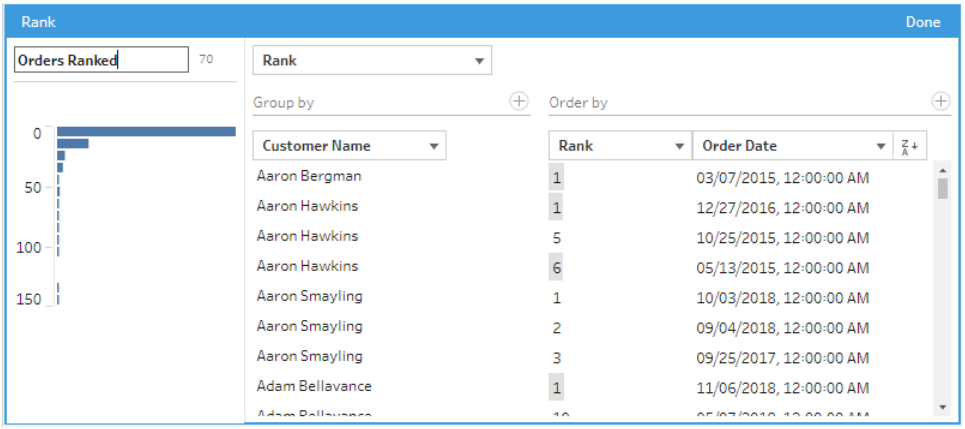
     After you select your first field, click the plus ⊕ icon to add any additional fields to your calculation. If you want to include all rows or remove a selected field, right-click or Cmd-click (MacOS) in the drop-down box for the fields in the **Group by** section and select **Remove Field**.

   - In the **Order by** section, select the fields that you want to use to rank your new values. The field where you selected the **Create Calculated Field** >**Rank** menu option is added by default.

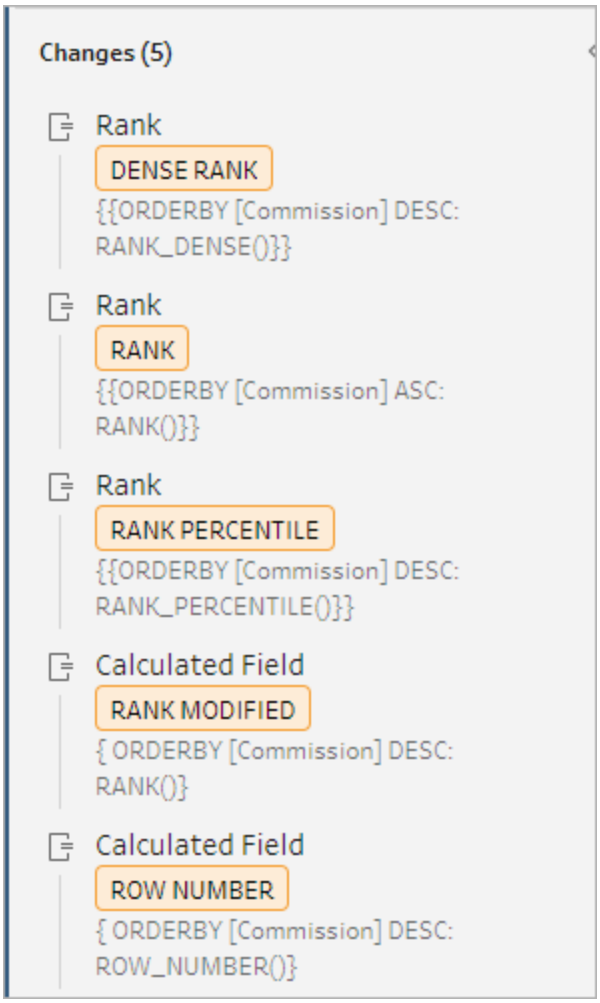
     Click the plus ⊕ icon to add any additional fields to your calculation, then select your **Rank** type. Click the sort icon to change the rank order from descending (DESC) to ascending (ASC).

> **Note**: Rank values vary by the data type assigned to the field.

- In the left pane, double-click in the field header and enter a name for your calculation.



3. Click **Done** to add your new calculated field. In the **Changes** pane, you can see the calculation that Tableau Prep Builder generated. Click **Edit** to open the visual calculation editor to make any changes.



Calculate tiles

Use the Tile feature to distribute rows into a specified number of buckets by creating a calculated field. You select the fields that you want to distribute by, and the number of groups (tiles) to be used. You can also select additional fields for creating partitions where the tiled rows are distributed into groups. Use the Calculation editor to input the syntax manually or use the Visual Calculation editor to select the fields and Tableau Prep writes the calculation for you.

For example, if you have rows of student data and wanted to see which students are in the top 50% and bottom 50%, you can group the data into two tiles.

Student	Subject	Grade
George	English	65
Christina	Science	80
Brian	English	89
Jennifer	Math	55
Daniel	Math	70
Ian	Science	60
Amanda	English	90
Heather	Math	65
Frank	Science	70

The following example shows two groups for the upper and lower half of student grades. The syntax for this method is:

```
{ORDERBY [Grade] DESC:NTILE(2)}
```

Calculation1	Student	Subject	Grade
1	Amanda	English	90
1	Brian	English	89
1	Christina	Science	80
1	Daniel	Math	70
1	Frank	Science	70
2	George	English	65
2	Heather	Math	65
2	Ian	Science	60
2	Jennifer	Math	55

You can also create a partition, where each value of a field is a separate partition, and divide data into groups for each partition.

The following example shows creating partitions for the Subject field. A partition is created for each subject and two groups (tiles) are created for the Grade field. The rows are then distributed evenly into the two groups for the three partitions. The syntax for this method is:

```
{PARTITION [Subject]:{ORDERBY [Grade] DESC:NTILE(2)}}
```

Calculation1	Student	Subject	Grade
1	Daniel	Math	70
1	Heather	Math	65
2	Jennifer	Math	55
1	Christina	Science	80
1	Frank	Science	70
2	Ian	Science	60
1	Amanda	English	90
1	Brian	English	89
2	George	English	65

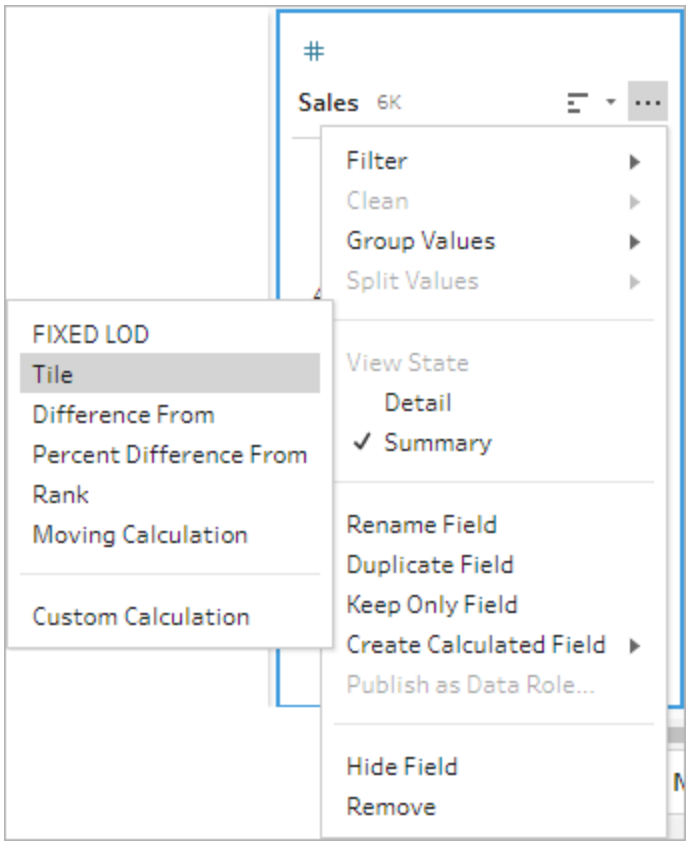
Create Tile calculations

To create tile calculations, you can use the Calculation editor to write the calculation yourself or if you want a more guided experience, you can use the Visual Calculation editor where you select your fields and Tableau Prep writes the calculation for you.

Visual Calculation editor

When you use the visual calculation editor to create a tile calculation, a preview of the results is shown in the left pane.

1. Select a profile card to create a tile calculation.
2. Click on the **More** options menu and select **Create Calculated Field** > **Tile**.



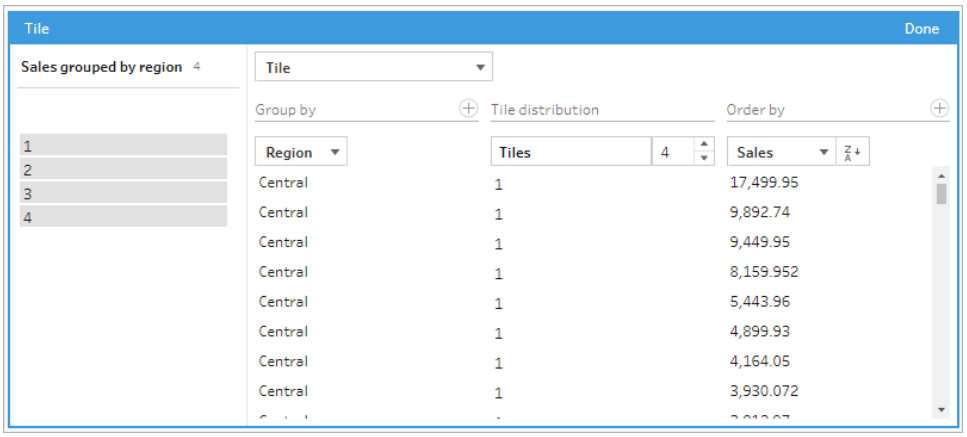
The selected profile card is added as an ORDERBY field.f

3. In the Visual calculation editor, do the following:

   - Select the number of tile groupings you want. The default value for Tiles is 1.
   - In the **Group by** section, select the fields for the rows you want to compute values for. This creates the `PARTITION` part of the calculation. You can have multiple Group-by fields for a single calculation.

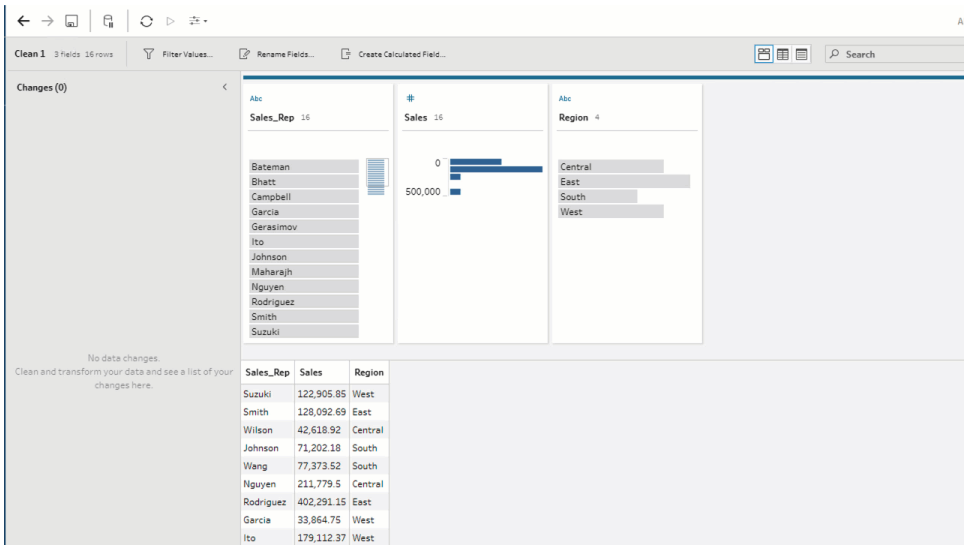
     Click the plus ⊕ icon to add any additional fields to your calculation. If you want to include all rows or remove a selected field, right-click or Cmd-click (MacOS) in the drop-down box for the fields in the **Group by** section and select **Remove Field**.
   - In the left pane, double-click in the field header and enter a name for your calculation.
   - In the **Order by** section, select one or more fields that you want to use to group and distribute your new values. You must have at least one Order by field. The field

where you selected **the Create Calculated Field >Tile** menu option is added by default.



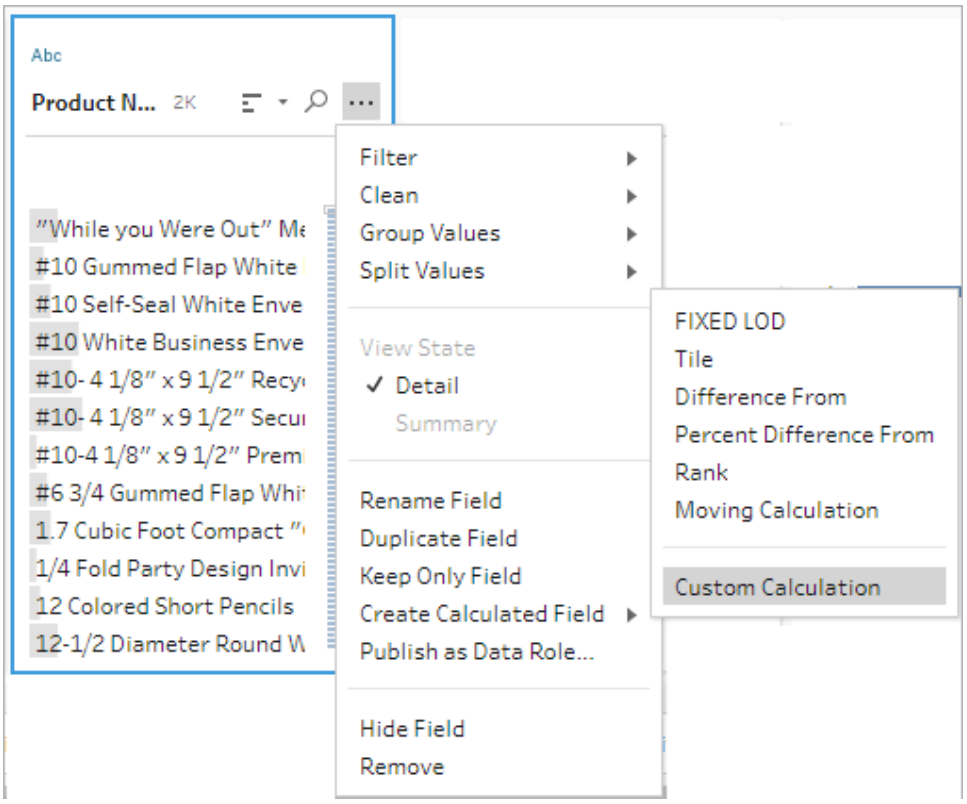
4. To sort the results, do the following:
   - Click on any of the Calculation rows to filter the results for the selected grouping
   - Change the ascending or descending order of the order-by field.
5. Click **Done**to add your new calculated field.
6. In the **Changes** pane, you can see the calculation that Tableau Prep Builder generated. Click **Edit** to open the visual calculation editor to make any changes.

   The following example shows a quartile division of rows. A partition is created based on four US regions and then the Sales field data is evenly grouped into the partitions.

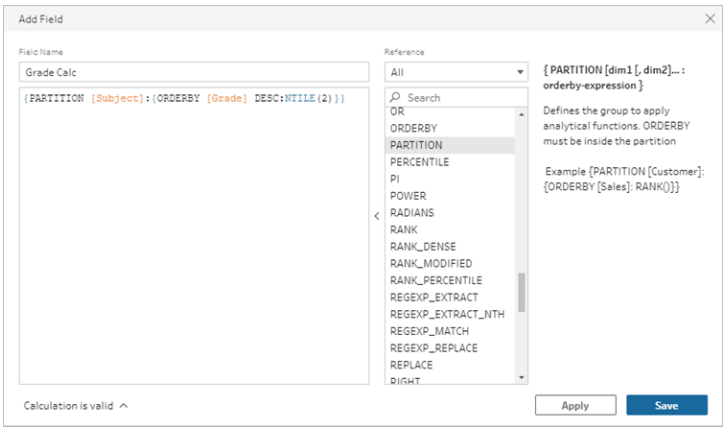


Calculation editor

1. In the Profile pane toolbar, click Create Calculated Field, or in a profile card or data grid, click on the **More options** ··· menu and select **Create Calculated Field** > **Custom Calculation**.



2. In the Calculation editor, enter a name for your calculation and enter the expression. For example, to order rows of students by grades into two groups and then group them by subject, use : `{PARTITION [Subject]:{ORDERBY [Grade] DESC:NTILE(2)}}`.



Tile calculations include the following elements:

- **PARTITION** (optional): A partition clause differs the rows of a result set into partitions where the NTILE() function is used.

- **ORDERBY** (required) The ORDER BY clause defines the distribution of rows in each partition where the NTILE() is used.
- **NTILE** (required): NTILE is the integer into which the rows are divided.

  > **Note**: When all of the rows are divisible by the NTILE clause, the feature divides the rows evenly among the number of tiles. When the number of rows isn't divisible by the NTILE clause, the resulting groups are divided into different sized bins.

- **DESC** or **ASC** (optional): Represents descending (DESC) or ascending (ASC) order. By default, the tile is sorted in descending order, so you don't need to specify this in the expression. If you want to change the sort order, add **ASC** to the expression.

3. Click **Save**.

   The generated field shows the tile grouping (bin) assignments associated with each row in the table.

Calculate Values Across Multiple Rows

> **Note**: Starting in version 2020.4.1, you can create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

Multi-row calculations let you compute values between multiple rows of data in your flow. While similar to table calculations in Tableau, multi-row calculations apply to your entire data set when you run your flow. You can also build on the results using other types of calculations.

> **Note**: When creating a calculated field, don't use the reserved name 'Number of Records'. Using this reserved name will result in a permission-related error.

In Tableau, table calculations only apply to values in your visualisation. While you can build on the result, you must use another table calculation to do so. For more information about using table calculations in Tableau, see Transform Values with Table Calculations in the Tableau help.

Performing table calculations during data preparation can provide greater flexibility when analysing data in Tableau. You can easily reuse the calculation when building your view and the underlying calculation isn't impacted by filtering. Workbook load times for large data sets can be faster as the table calculation isn't recalculated after the query runs.

Tableau Prep currently supports the following multi-row calculations:

- **Difference from**: Computes the difference between the current row value and another value.
- **Percent difference from**: Computes the difference between the current row value and another value as a percentage.
- **Moving calculations**: Returns the sum or average of a numeric field within a flexible set of rows.
- **Fill Down calculations**: Computes and fills in missing data from the rows above.
- **Running calculations**: Computes the sum or average of values in a column.

Use the visual calculation editor to quickly generate the calculation, or write your own custom calculation in the calculation editor.

Calculate Difference From

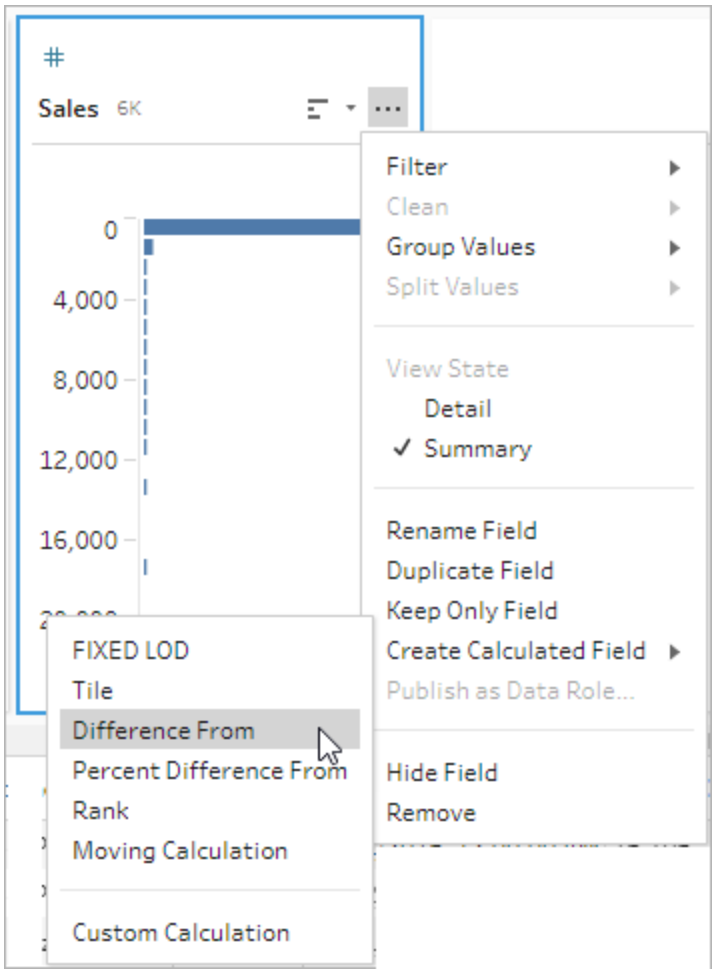
Supported in Tableau Prep Builder 2023.2 and later and on the web in Tableau Cloud. Also supported in Tableau Server 2023.3 and later.

A **Difference From** calculation computes the difference between the current value and a value N rows before or after the current row.

Visual calculation editor

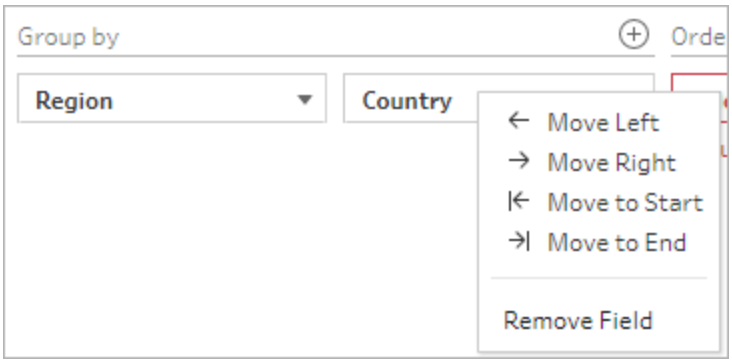
Select fields from a list and Tableau Prep builds the calculation for you as you make your selections. A preview of the new field results is shown in the left pane and you can review the calculation results in the far right of the pane.

1. In a profile card or results pane, click the **More options** ··· menu and select **Create Calculated Field** > **Difference From**.



2. In the **Group by** section, select the fields with rows that you want to include in the calculation. This partitions your table when performing the calculation. To apply the calculation to all rows in the table, accept the default value **Full table**.

   After you select your first field, click the plus ⊕ icon to add any additional **Group by** fields to your partition. To reorder or remove fields, right-click or Ctrl-click (MacOS) and select an action from the menu.

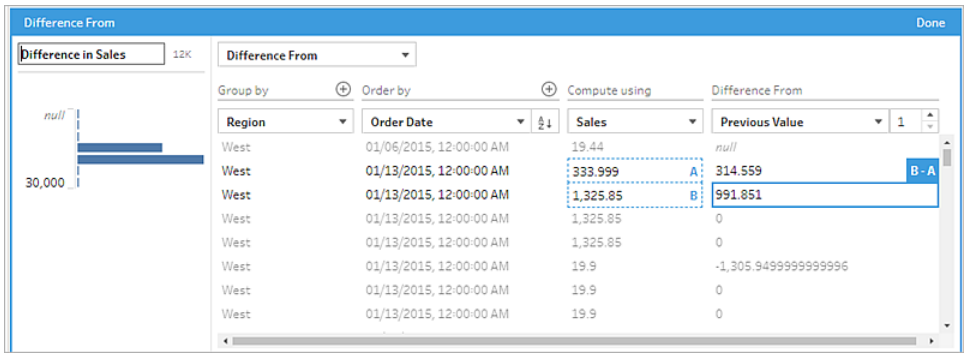


3. In the **Order by** section, select the fields that you want to use as the sort order. This field is used to specify how the `LOOKUP` function orders the rows in your table.

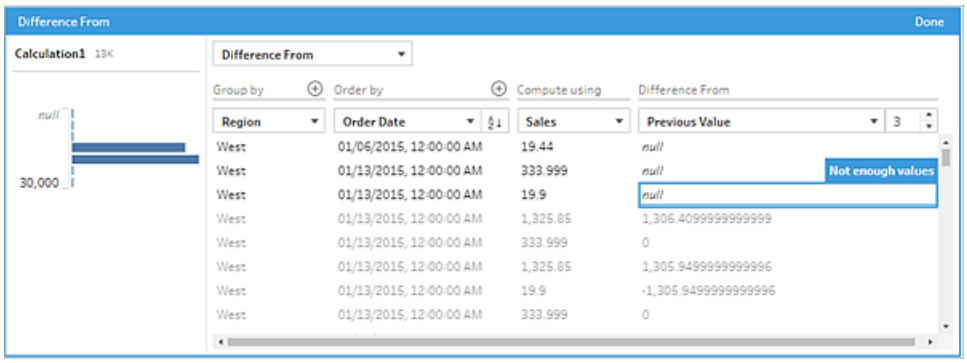
   If the field where you selected the **Create Calculated Field** >**Difference From** menu option is a date or time field, then this field is added by default, but you can change it.

   Click the plus icon to add any additional **Order by** fields to your calculation. Click the sort icon to change the order from ascending (ASC) to descending (DESC). You can also right-click or Ctrl-click (MacOS) and select an action from the menu to reorder or remove fields.
4. In the **Compute using** section, select the field with the values that you want to use to calculate your results.
5. In the **Difference From** section, select the rows to use to calculate the difference. For example select **Previous Value**, 2 to calculate the difference between the current value and a value 2 rows before that value. Annotations highlight the rows used to perform the calculation.

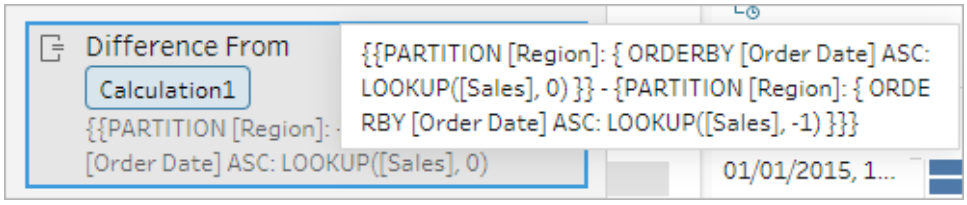
   By default, the calculation preview will show you the first non-null row. However, you can click on any row in the results table and see an updated preview of the selected value.



If the calculation can't be performed with the current settings, the annotation **Not enough values** is shown. To resolve this issue, either select a different current value or change the configuration in the **Difference From** section.



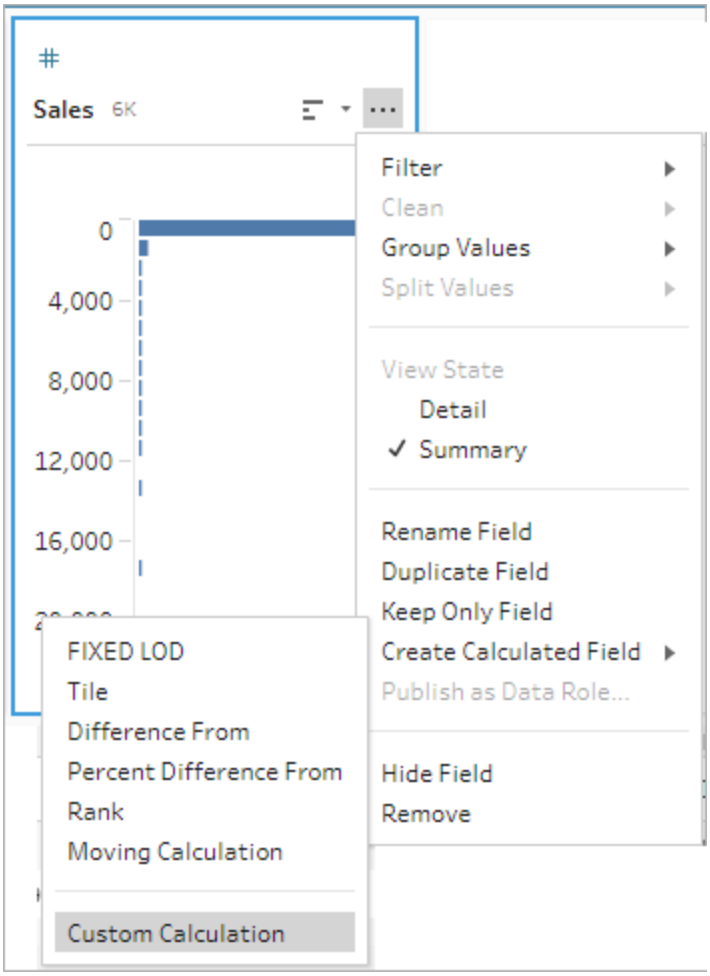
6. In the left pane, double-click in the field header and enter a name for your calculation.
7. Click **Done** to add your new calculated field. In the **Changes** pane, you can see the calculation that Tableau Prep generated. Click **Edit** to open the visual calculation editor to make any changes.



Calculation editor

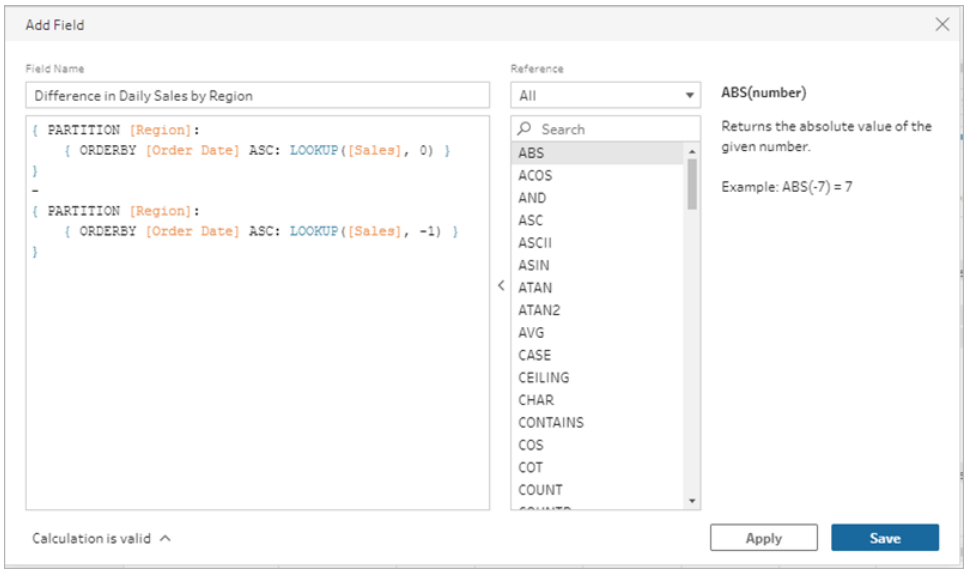
If you want to write your own calculation to calculate the difference between two values, use the `LOOKUP` function in the Calculation editor.

1. In the Profile pane toolbar, click **Create Calculated Field**, or in a profile card or data grid, click the **More options** ··· menu and select **Create Calculated Field** > **Custom Calculation**.



2. In the Calculation editor, enter the expression. For example, to find the difference between current sales and the previous day's sales by region, create a calculation like the one shown below.

```
{PARTITION [Region]:{ ORDERBY [Order Date]ASC:LOOKUP([Sales],0)}}
-
{ PARTITION [Region]:{ ORDERBY [Order Date]ASC:LOOKUP
([Sales],1)}}
```



3. Enter a name for your calculation, and click **Save**.

Calculate Percent Difference From

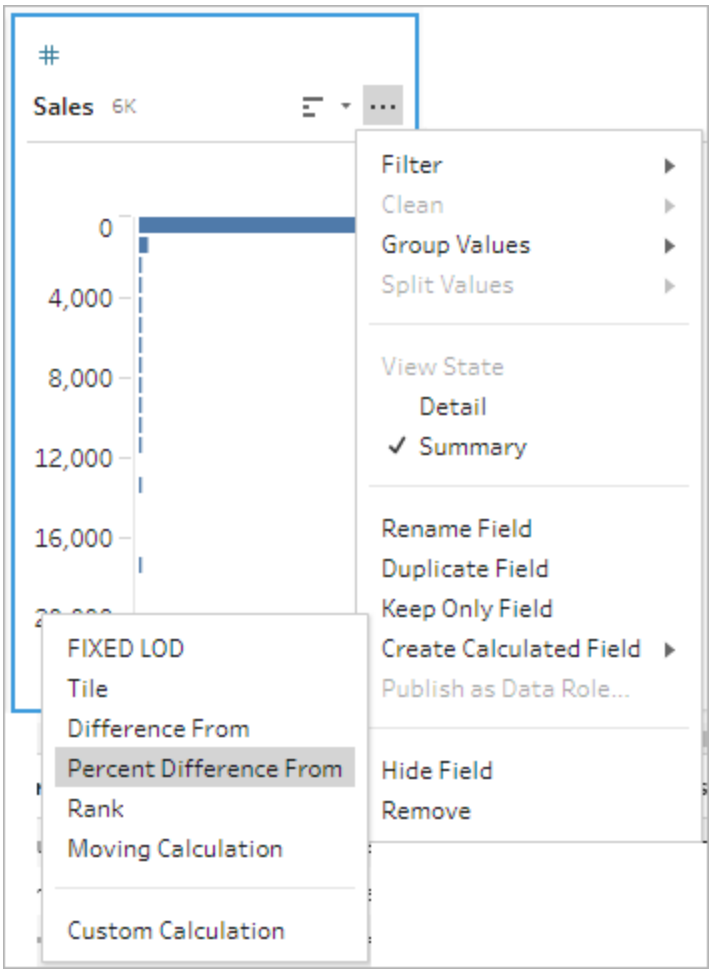
Supported in Tableau Prep Builder 2023.2 and later and on the web in Tableau Cloud. Also supported in Tableau Server 2023.3 and later.

A **Percent Difference From** calculation computes the difference between the current value and a value N rows before or after the current row as a percentage. For example `Value1-Value2/Value2`.

Visual Calculation editor

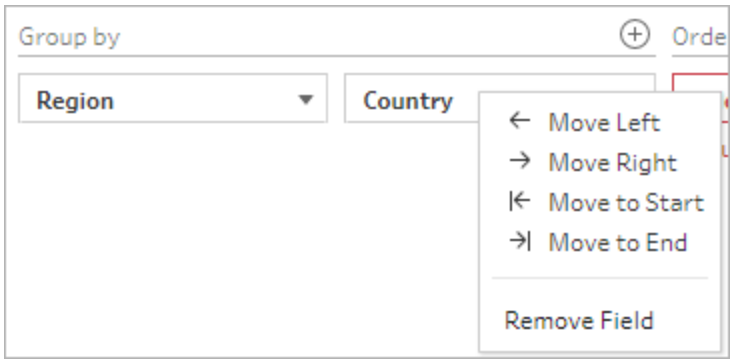
Select fields from a list and Tableau Prep builds the calculation for you as you make your selections. A preview of the new field results is shown in the left pane and you can review the calculation results in the far right of the pane.

1. In a profile card or results pane, click the **More options** ··· menu and select **Create Calculated Field** > **Percent Difference From**.



2. In the **Group by** section, select the fields with rows that you want to include in the calculation. This partitions your table when performing the calculation. To apply the calculation to all rows in the table, accept the default value **Full table**.

   After you select your first field, click the plus ⊕ icon to add any additional **Group by** fields to your partition. To reorder or remove fields, right-click or Ctrl-click (MacOS) and select an action from the menu.

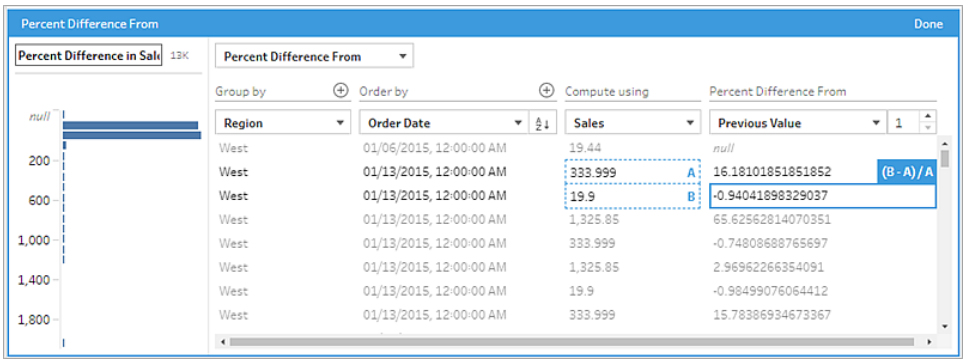


3. In the **Order by** section, select the fields that you want to use as the sort order. This field is used to specify how the `LOOKUP` function orders the rows in your table.

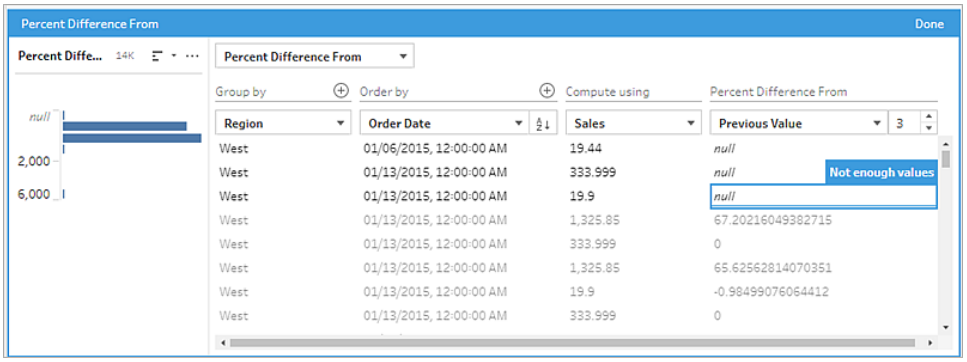
   If the field where you selected the **Create Calculated Field** > **Percent Difference From** menu option is a date or time field, then this field is added by default, but you can change it.

   Click the plus icon to add any additional **Order by** fields to your calculation. Click the sort icon to change the order from ascending (ASC) to descending (DESC). You can also right-click or Ctrl-click (MacOS) and select an action from the menu to reorder or remove fields.
4. In the **Compute using** section, select the field with the values that you want to use to calculate your results.
5. In the **Percent Difference From** section, select the rows to use to calculate your result. For example select **Previous Value**, 2 to calculate the percent difference between the current value and a value 2 rows before that value. Annotations highlight the rows used to perform the calculation.

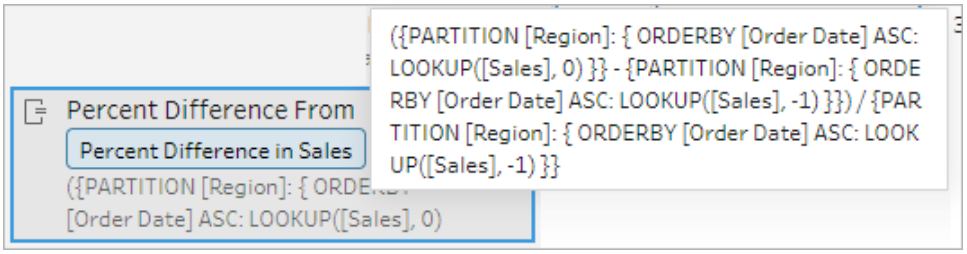
   By default, the calculation preview will show you the first non-null row. However, you can click on any row in the results table and see an updated preview of the selected value.



If the calculation can't be performed with the current settings, you will see the annotation **Not enough values**. To resolve this, either select a different current value or change the configuration in the **Percent Difference From** section.



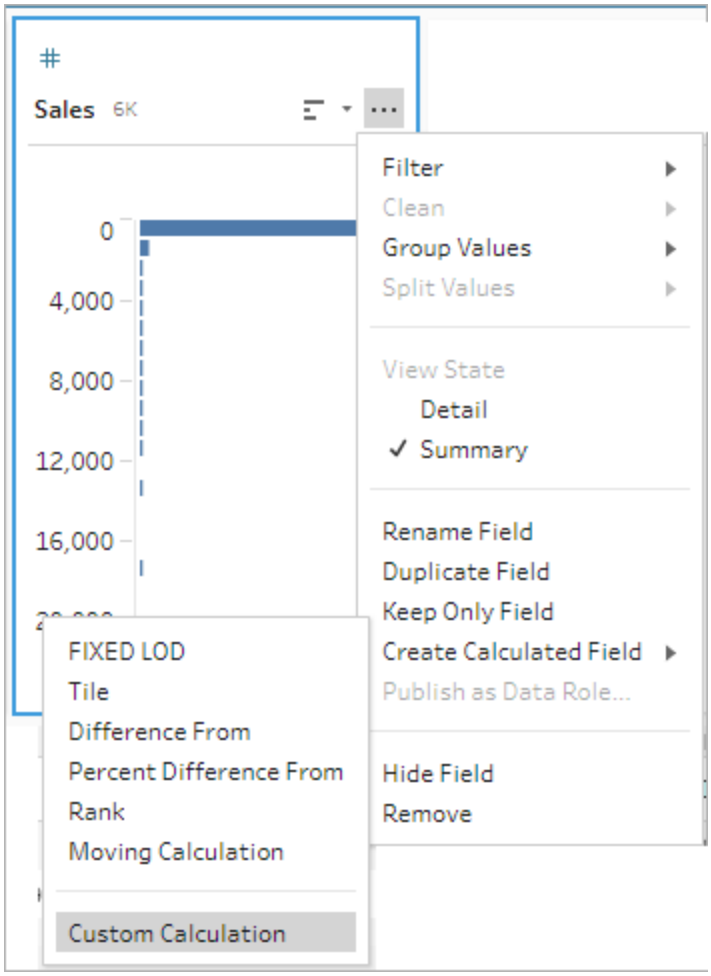
6. In the left pane, double-click in the field header and enter a name for your calculation.
7. Click **Done** to add your new calculated field. In the **Changes** pane, you can see the calculation that Tableau Prep generated. Click **Edit** to open the visual calculation editor to make any changes.



Calculation editor

If you want to write your own calculation to calculate the percent difference between two values, use the `LOOKUP` function in the Calculation editor.

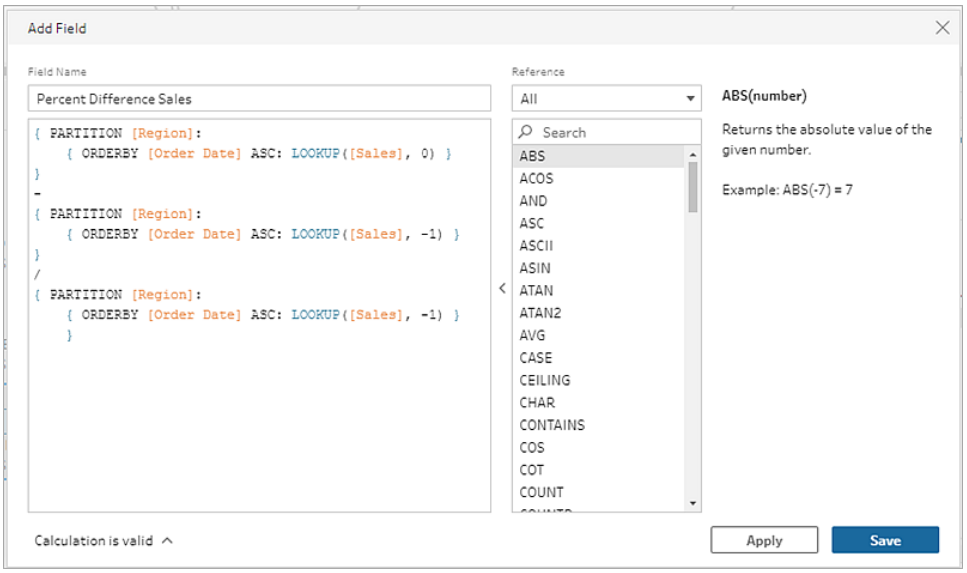
1. In the Profile pane toolbar, click **Create Calculated Field**, or in a profile card or data grid, click the **More options** ··· menu and select **Create Calculated Field** > **Custom Calculation**.



2. In the Calculation editor, enter the expression. For example, to find the percent difference between current sales and previous days sales by region, create a calculation like the one shown below.

```
{ PARTITION [Region]:{ ORDERBY [Order Date]ASC:LOOKUP
([Sales],0)}}
-
{ PARTITION [Region]:{ ORDERBY [Order Date]ASC:LOOKUP([Sales],-
1)}}
```

```
/
{ PARTITION [Region]:{ ORDERBY [Order Date]ASC:LOOKUP([Sales],-
1)}}
```



3. Enter a name for your calculation, and click **Save**.

Calculate Moving Average or Sum

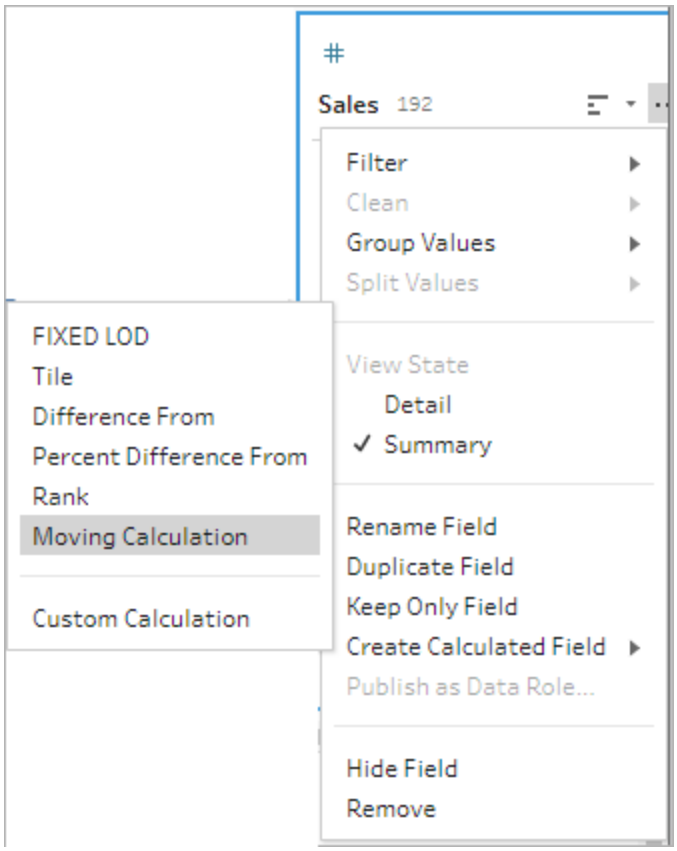
Supported in Tableau Prep Builder 2023.2 and later and on the web in Tableau Cloud. Also supported in Tableau Server 2023.3 and later.

Create a moving calculation to better understand trends in your data and reduce overall fluctuations. In Tableau Prep you can calculate a moving average or sum across a specified number of values before or after the current value. For example tracking the three month moving average of sales per region.

Visual Calculation editor

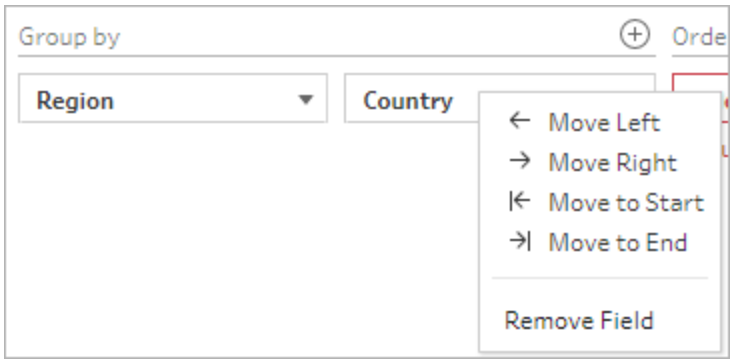
Select fields from a list and Tableau Prep builds the calculation for you as you make your selections. A preview of the new field results is shown in the left pane and you can review the calculation results in the far right of the pane.

1. In a profile card or results pane, click the **More options** ··· menu and select **Create Calculated Field** > **Moving Calculation**.



2. In the **Group by** section, select the fields with rows that you want to include in the calculation. This partitions your table when performing the calculation. To apply the calculation to all rows in the table, accept the default value **Full table**

   After you select your first field, click the plus ⊕ icon to add any additional **Group by** fields to your calculation. To reorder or remove fields, right-click or Ctrl-click (MacOS) and select an action from the menu.

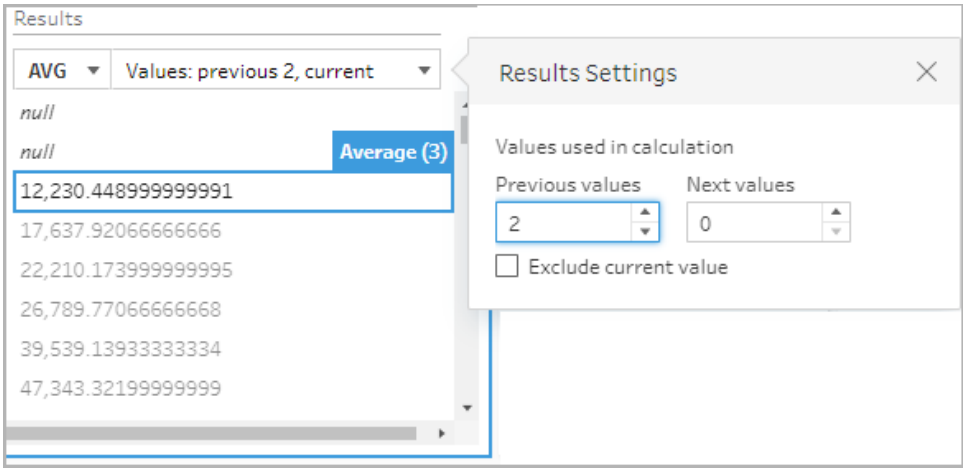


3. In the **Order by** section, select the fields that you want to use as the sort order. This field is used to specify how the `LOOKUP` function orders the rows in your table.

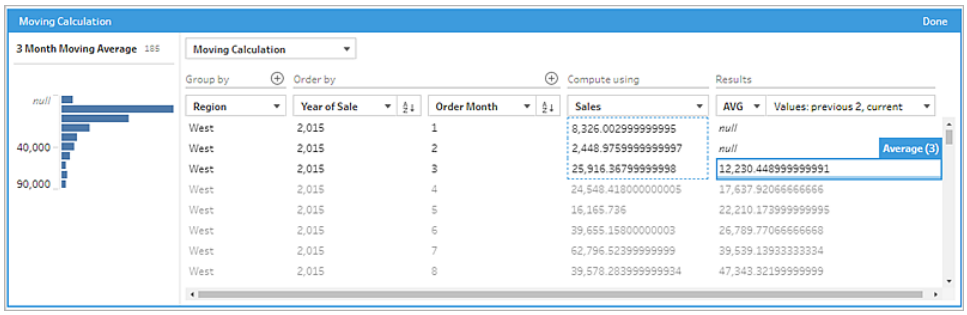
   If the field where you selected the **Create Calculated Field** > **Moving Calculation** menu option is a date or time field, then this field is added by default, but you can change it.

   Click the plus icon to add any additional **Order by** fields to your calculation. Click the sort icon to change the order from ascending (ASC) to descending (DESC). You can also right-click or Ctrl-click (MacOS) and select an action from the menu to reorder or remove fields.
4. In the **Compute using** section, select the field with the values that you want to use to calculate your results.
5. In the **Results** section, select the aggregation you want to perform (sum or average), the number of rows to include in the calculation, and whether to include the current row or exclude it.

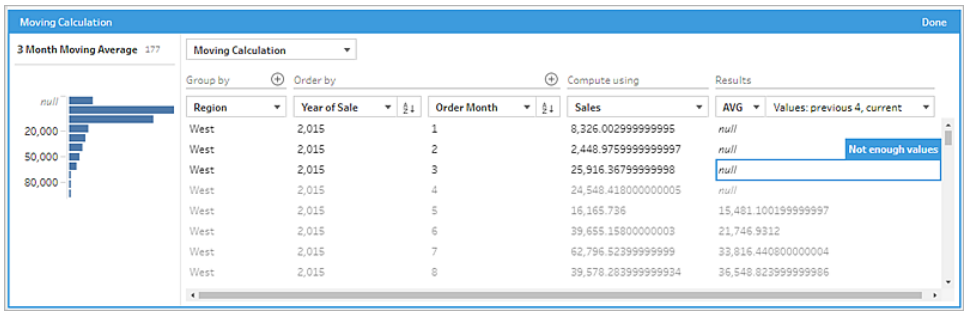
   To change the results setting, click the drop-down in the **Values** field. For example, to calculate the moving average of sales across the current month and the previous 2 months, set the **Previous values** to 2 and close the dialog.



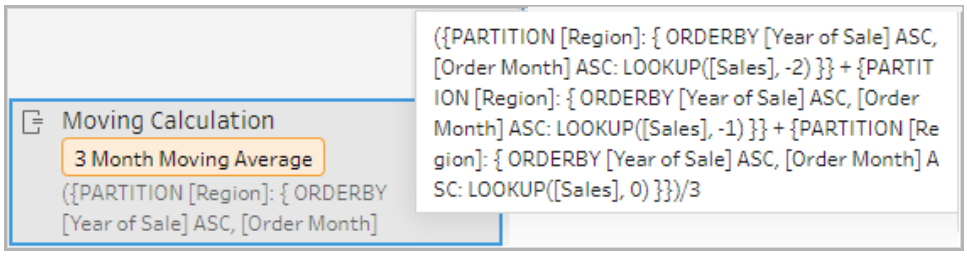
6. By default, the calculation preview shows you the first non-null row. However, you can click on any row in the results table and see an updated preview of the selected value. Annotations highlight the rows used to perform the calculation.



If the calculation can't be performed with the current settings, you will see the annotation **Not enough values**. To resolve this, click the drop-down in the **Values** field to change the configuration in the **Results Settings**.



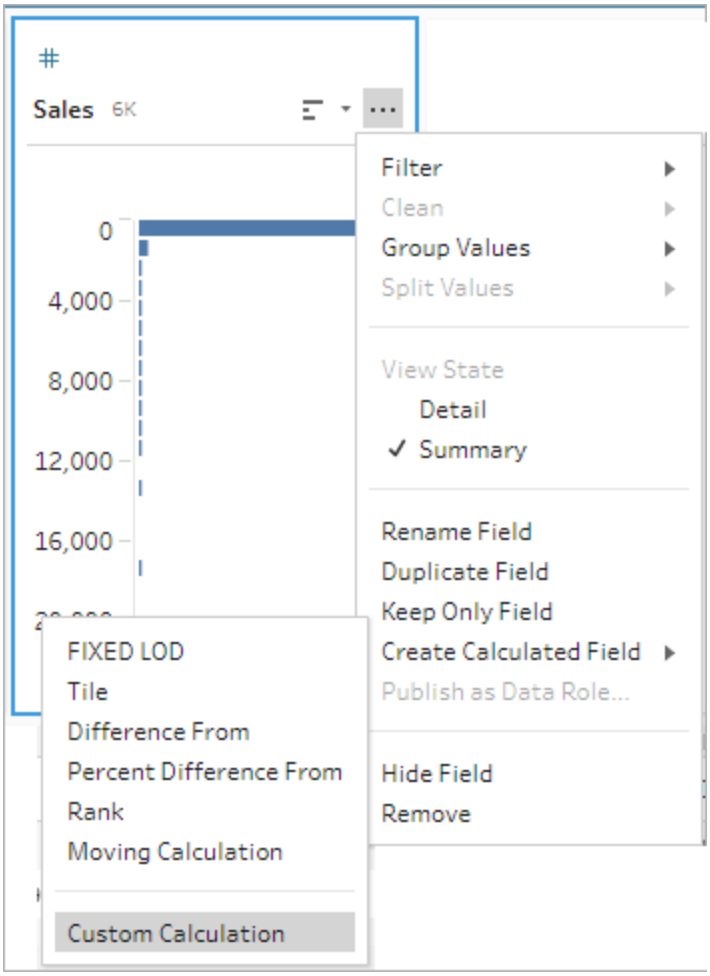
7. In the left pane, double-click in the field header and enter a name for your calculation.
8. Click **Done** to add your new calculated field. In the **Changes** pane, you can see the calculation that Tableau Prep generated. Click **Edit** to open the visual calculation editor to make any changes.



Calculation editor

If you want to write your own calculation to calculate the moving average or sum, use the `LOOKUP` function in the Calculation editor.

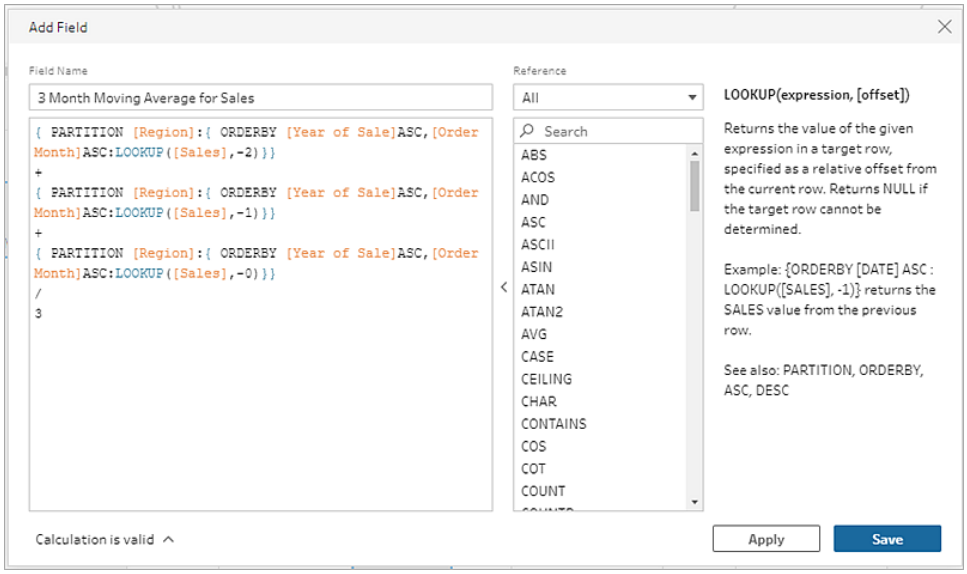
1. In the Profile pane toolbar, click **Create Calculated Field**, or in a profile card or data grid, click the **More options** ··· menu and select **Create Calculated Field** > **Custom Calculation**.



2. In the Calculation editor, enter the expression. For example, to find the three month moving average of sales per region, create a calculation like the one shown below.

> **Note**: This example assumes that the data set is at the correct level of detail, one row for each month. If your data set is not at the correct level of detail, consider using an aggregation step to change this before applying the calculation.

```
{ PARTITION [Region]:{ ORDERBY [Year of Sale]ASC,[Order
Month]ASC:LOOKUP([Sales],-2)}}
+
{ PARTITION [Region]:{ ORDERBY [Year of Sale]ASC,[Order
Month]ASC:LOOKUP([Sales],-1)}}
+
{ PARTITION [Region]:{ ORDERBY [Year of Sale]ASC,[Order
Month]ASC:LOOKUP([Sales],-0)}}
/
3
```



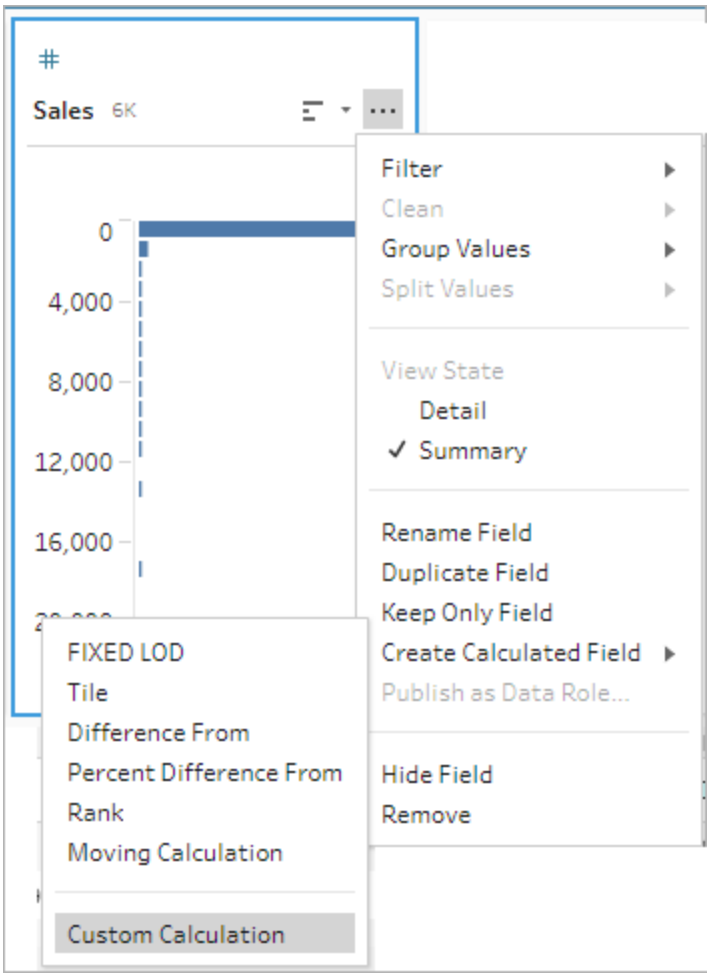
3. Enter a name for your calculation, and click **Save**.

Get Previous Value

Supported in Tableau Prep Builder 2023.2 and later and on the web in Tableau Cloud. Also supported in Tableau Server 2023.3 and later.

If you need to create a field with the value from a previous row, you can create a custom calculation using the `LOOKUP` function.

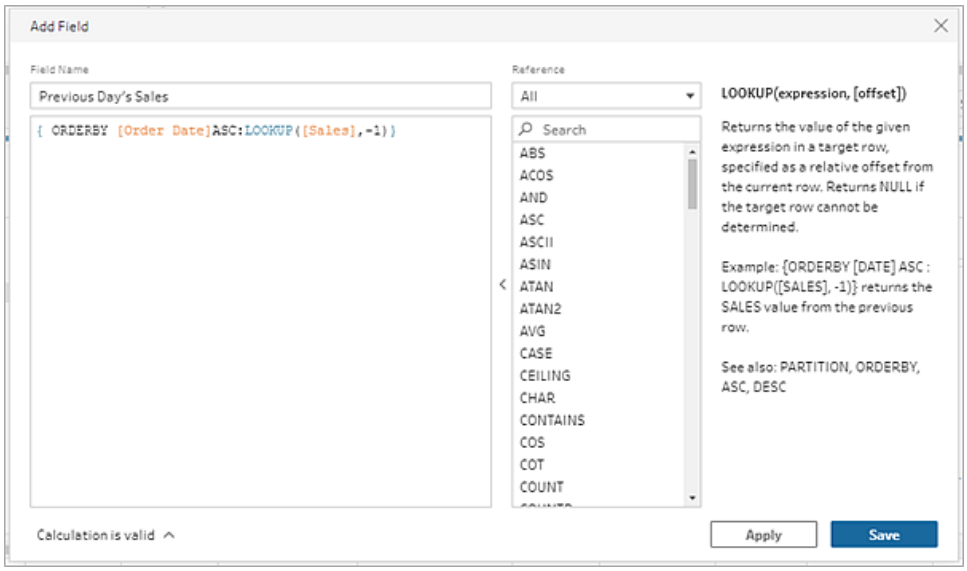
1. In the Profile pane toolbar, click **Create Calculated Field**, or in a profile card or data grid, click the **More options** ··· menu and select **Create Calculated Field** > **Custom Calculation**.



2. In the Calculation editor, enter the expression. For example, to find the previous sales value by order date, create a calculation like the one shown below.

> **Note**: This example assumes that the data set is at the correct level of detail, one row for each day. If your data set is not at the correct level of detail, consider using an aggregation step to change this before applying the calculation.

```
{ ORDERBY [Order Date]ASC:LOOKUP([Sales],-1)}
```



3. Enter a name for your calculation, and click **Save**.

Fill Down Calculation

Supported in Tableau Prep Builder 2023.2 and later and on the web in Tableau Cloud. Also supported in Tableau Server 2023.3 and later.

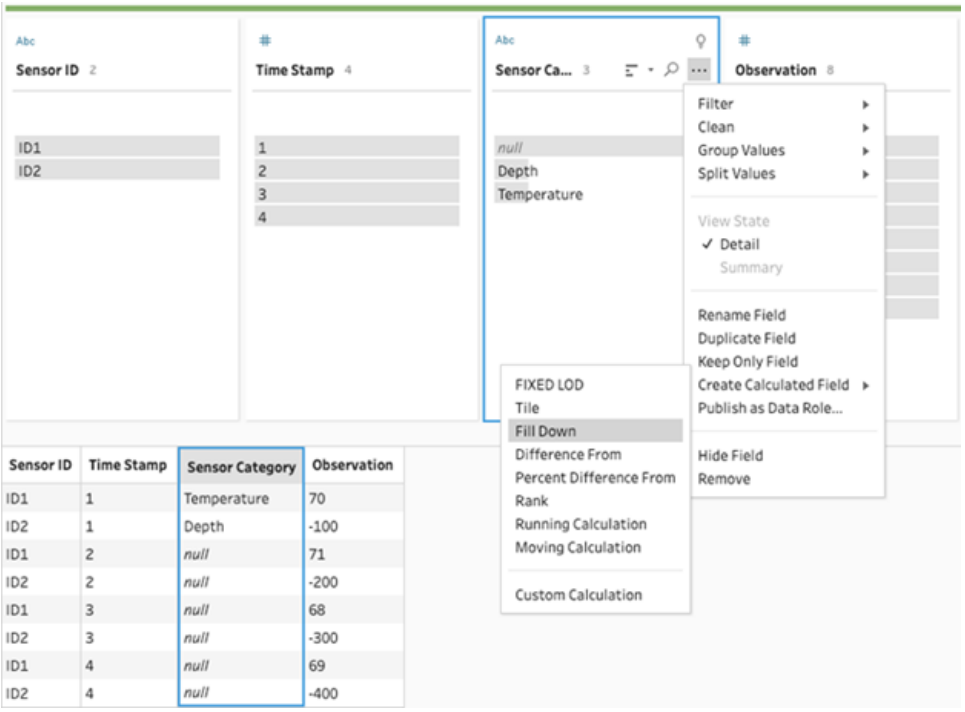
A Fill Down calculation computes and fills in missing data so that null values are replaced with the values from the rows above, allowing for the data to be easily visualised. The example below shows the results of a fill down calculation by replacing the null values from the Student field with values based on the rows above.

Student	Subject	Grade
Amanda	English	90
	Math	89
	Science	80
Daniel	English	70
	Math	70
	Science	65
Ian	English	65
	Math	60
	Science	55

Student	Subject	Grade
Amanda	English	90
Amanda	Math	89
Amanda	Science	80
Daniel	English	70
Daniel	Math	70
Daniel	Science	65
Ian	English	65
Ian	Math	60
Ian	Science	55

Prep Visual calculation editor

1. In a profile card or results pane, click the **More options** ··· menu, and select **Create Calculated Field** > **Fill Down**.



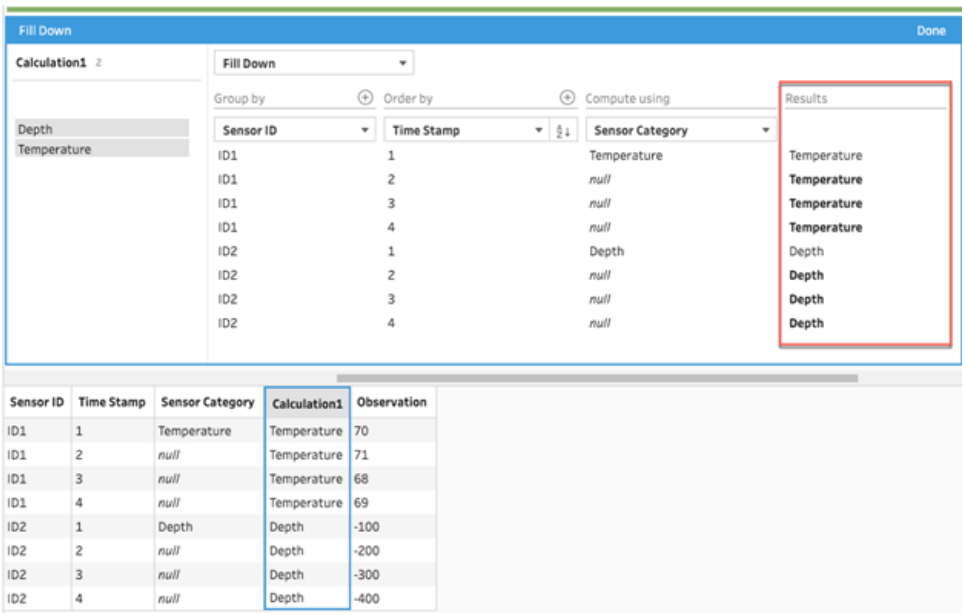
2. In the **Group by** section, you can select fields to partition the data when performing the calculation. If you don't want your data to be partitioned, accept the default value of **Full table**.

   After you select your first field, you can add additional Group by fields to your partition by clicking the plus ⊕ icon. To reorder or remove fields, right-click or Ctrl-click (MacOS) and select an action from the menu.

3. In the **Order by** section, select one or more fields that you want to use for the sort order. This field is used to specify the order of rows in a table.

   You can also add a source row number in the input step of text, CSV or Excel files to sort the data in the data source order.

4. Click the plus icon to add any additional Order by fields to your calculation. You must choose at least one field. Click the sort icon to change the order from ascending (ASC) to descending (DESC). You can also right-click or Ctrl-click (MacOS) and select an action from the menu to reorder or remove fields.
5. In the **Compute using** section, select the field with the values that you want to fill down. The **Results** field shows the filled down values in bold.



6. In the left pane, double-click in the field header and enter a name for your calculation.
7. Click **Done** to add your new calculated field. In the Changes pane, you can see the calculation that Tableau Prep generated. Click **Edit** to open the visual calculation editor to make any changes.

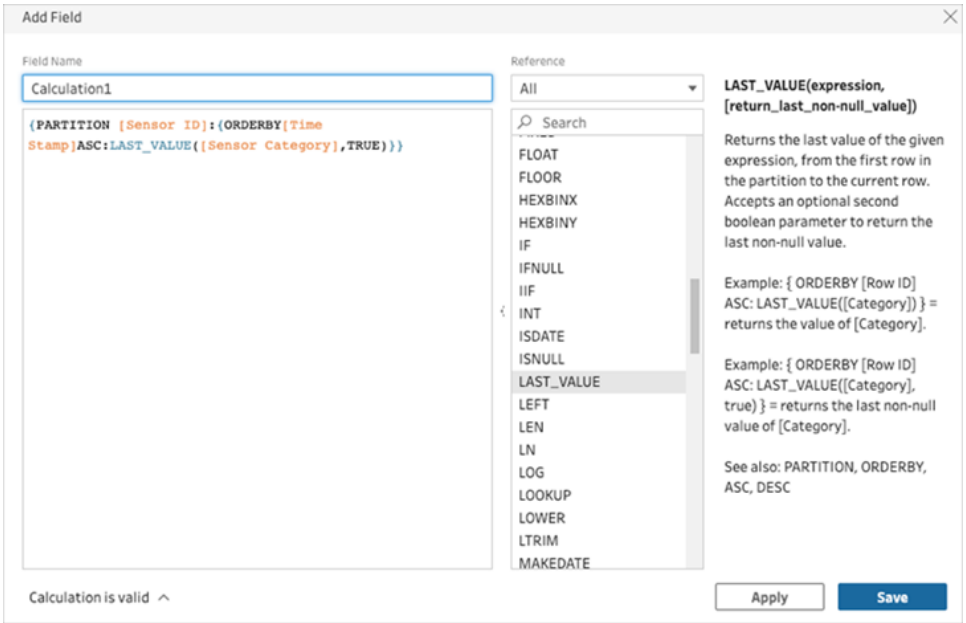
Calculation editor

1. In the Profile pane toolbar, click **Create Calculated Field**, or in a profile card or data grid, click the **More options** ··· menu and select **Create Calculated Field** > **Custom Calculation**.
2. In the Calculation editor, enter the expression.
3. Enter a name for your calculation, and click **Save**.

Examples

Fill down example with a partition: Based on the following data, you can fill down the missing values in the Sensor Category field using the `LAST_VALUE` function. In this example, *Sensor ID* is used to partition the data, *Time Stamp* is used to order the data, and *Sensor Category* is used to compute the data.

Sensor ID	Time Stamp	Sensor Category	Observation
ID1	1	Temperature	70
ID2	1	Depth	-100
ID1	2		71
ID2	2		-200
ID1	3		68
ID2	3		-300
ID1	4		69
ID2	4		-400

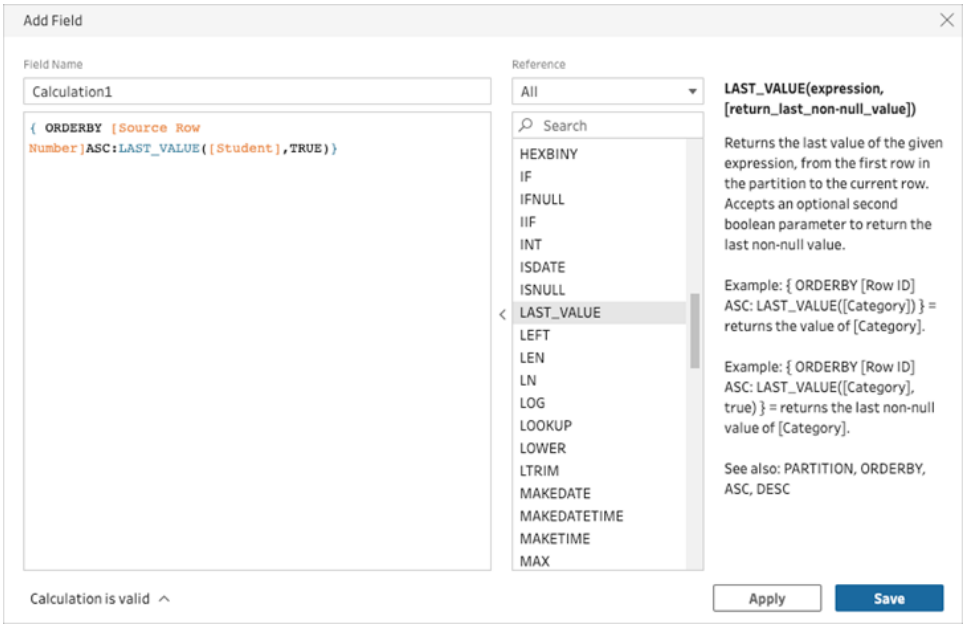


The fill down results is highlighted in the calculation pane.

Calculation1	Sensor ID	Time Stamp	Sensor Category	Observation
Temperature	ID1	1	Temperature	70
Temperature	ID1	2	*null*	71
Temperature	ID1	3	*null*	68
Temperature	ID1	4	*null*	69
Depth	ID2	1	Depth	-100
Depth	ID2	2	*null*	-200
Depth	ID2	3	*null*	-300
Depth	ID2	4	*null*	-400

Fill down example without a partition: Based on the following data, you can fill down the missing values of the *Student* field. The *Source Row Number* field has been added in an input step and is used to order the data in the source order. *Student* is used to compute the data.

Student	Subject	Grade
Amanda	English	90
	Math	89
	Science	80
Daniel	English	70
	Math	70
	Science	65
Ian	English	65
	Math	60
	Science	55



The fill down results is highlighted in the calculation pane.

Calculation1	Source Row Number	Student	Subject	Grade
Amanda	1	Amanda	English	90
Amanda	2	*null*	Math	89
Amanda	3	*null*	Science	80
Daniel	4	Daniel	English	70
Daniel	5	*null*	Math	70
Daniel	6	*null*	Science	65
Ian	7	Ian	English	65
Ian	8	*null*	Math	60
Ian	9	*null*	Science	55

Running Calculation

Supported in Tableau Prep Builder 2023.2 and later and on the web in Tableau Cloud. Also supported in Tableau Server 2023.3 and later.

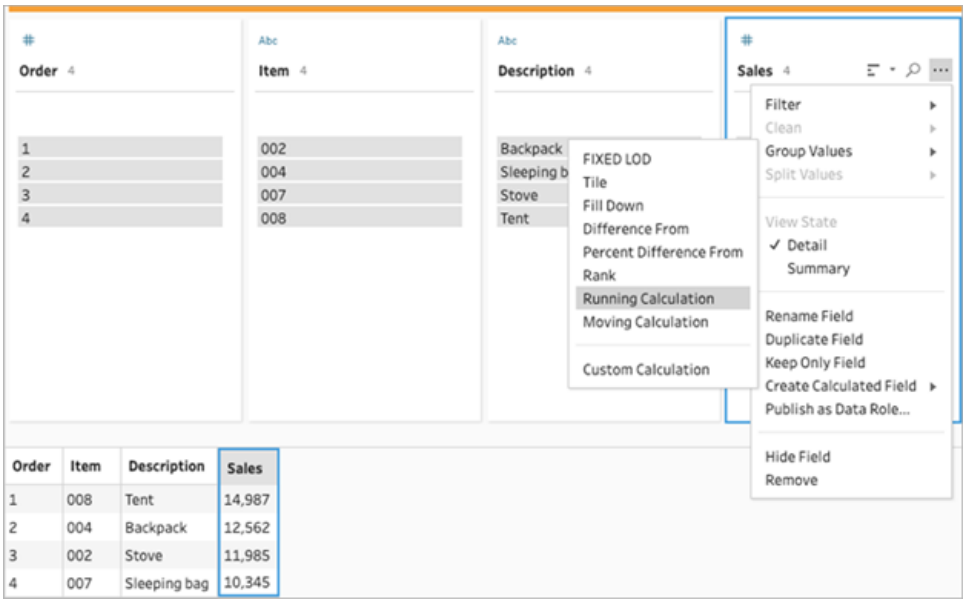
A Running Calculation computes the cumulative sum or average of a value and all previous values in the column. The example below shows the results of a running calculation for the sum of Sales over time.

Order	Item	Description	Sales
1	008	Tent	$14,987.00
2	004	Backpack	$12,562.00
3	002	Stove	$11,985.00
4	007	Sleeping bag	$10,345.00

Order	Item	Description	Sales	Running Sum of Sales
1	008	Tent	$14,987.00	$14,987.00
2	004	Backpack	$12,562.00	$27,549.00
3	002	Stove	$11,985.00	$39,534.00
4	007	Sleeping bag	$10,345.00	$49,879.00

Prep Visual calculation editor

1. In a profile card or results pane, click the **More options** ··· menu, and select **Create Calculated Field** > **Fill Down**.



2. In the Group by section, you can select fields to partition the data when performing the calculation. If you don't want your data to be partitioned, accept the default value of **Full table**.

   After you select your first field, you can add additional Group by fields to your partition by clicking the plus ⊕ icon. To reorder or remove fields, right-click or Ctrl-click (MacOS) and select an action from the menu.

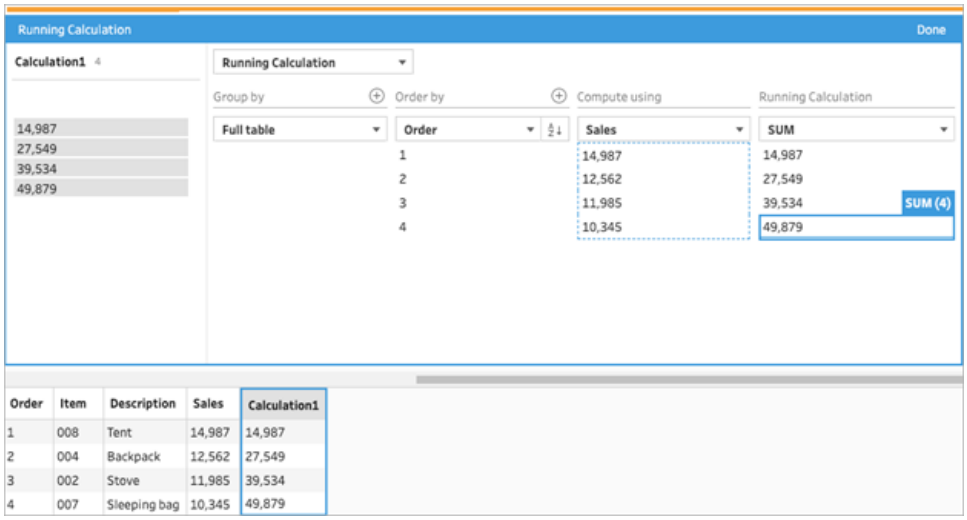
3. In the **Order by** section, select one or more fields that you want to use for the sort order. This field is used to specify the order of rows in a table.

   You can also add a source row number in the input step of text, CSV or Excel files to sort the data in the data source order.

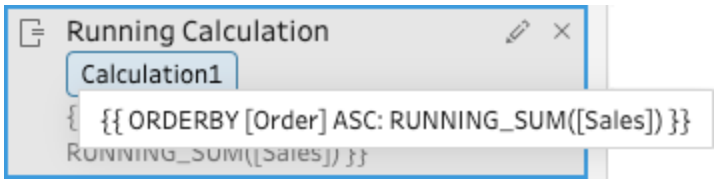
4. Click the plus ⊕ icon to add any additional Order by fields to your calculation. You must choose at least one field. Click the sort icon to change the order from ascending

(ASC) to descending (DESC). You can also right-click or Ctrl-click (MacOS) and select an action from the menu to reorder or remove fields.

5. In the **Results** section, select the aggregation you want to perform (SUM or AVG), for a running sum or a running average
6. Click the rows in the results table to see an updated preview of the selected value. Annotations highlight the rows used to perform the calculation..



7. In the left pane, double-click in the field header and enter a name for your calculation.
8. Click **Done** to add your new calculated field. In the Changes pane, you can see the calculation that Tableau Prep generated. Click **Edit** to open the visual calculation editor to make any changes.

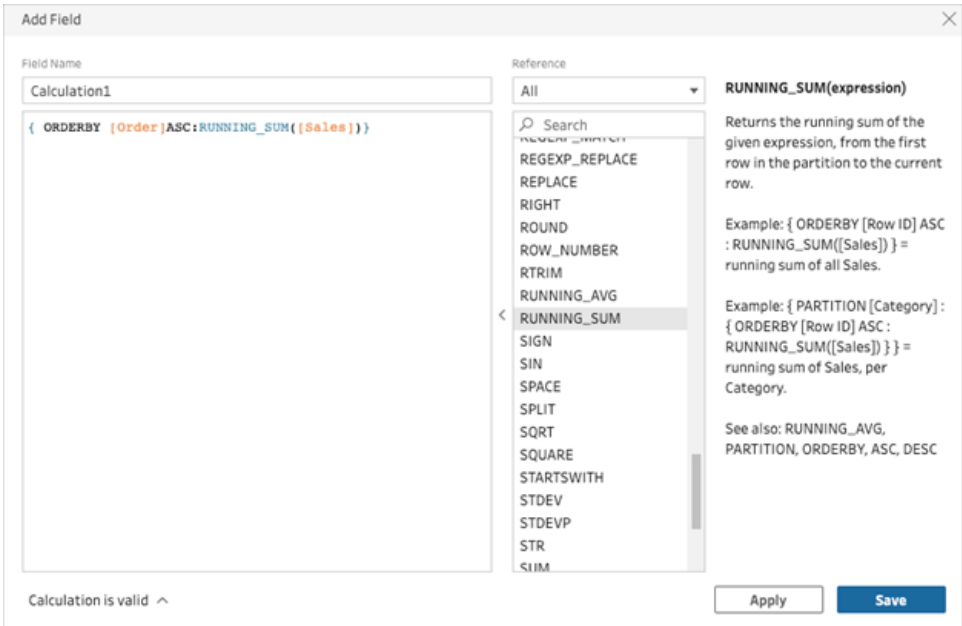


Calculation editor

1. In the Profile pane toolbar, click **Create Calculated Field**, or in a profile card or data grid, click the **More options** ··· menu and select **Create Calculated Field** > **Custom Calculation**.
2. In the Calculation editor, enter the expression. For example, based on the following data, you can calculate the total of *Sales* over time using the `RUNNING_SUM` function. The data

isn't partitioned and is grouped using **Full table**. The *Order* field is used to sort in chronological order, and *Sales* is used to compute the data.

Order	Item	Description	Sales
1	008	Tent	$14,987.00
2	004	Backpack	$12,562.00
3	002	Stove	$11,985.00
4	007	Sleeping bag	$10,345.00



3. Enter a name for your calculation, and click **Save**.

   The running sum results are highlighted in the calculation pane.

Calculation1	Order	Item	Description	Sales
14,987	1	008	Tent	14,987
27,549	2	004	Backpack	12,562
39,534	3	002	Stove	11,985
49,879	4	007	Sleeping bag	10,345

Use Einstein Copilot

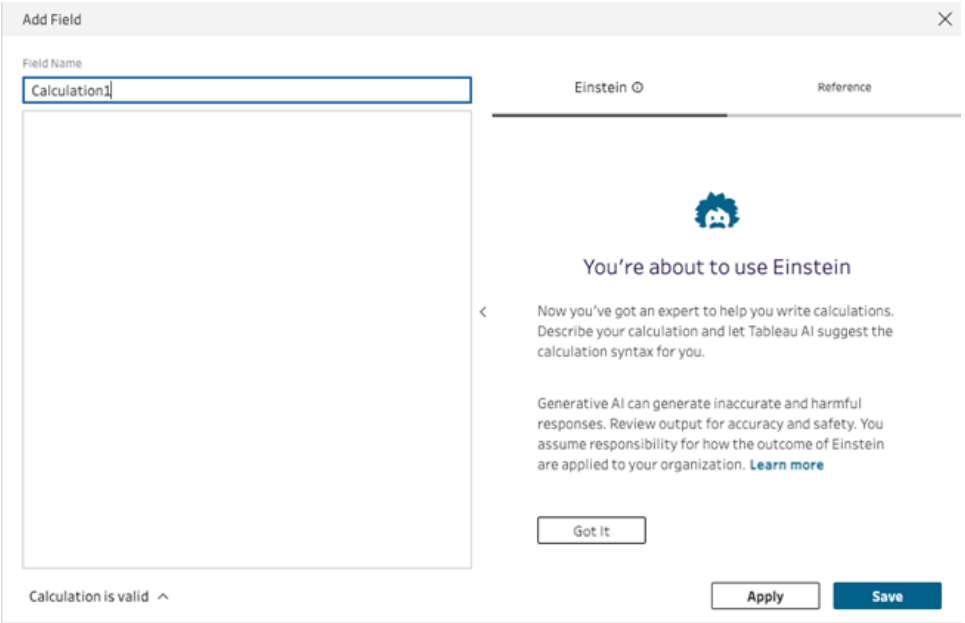
Supported in Tableau Prep Web Authoring version 2024.2.

With Einstein Copilot, simply describe a calculation and Einstein will assess the formula. With one click, you can add powerful calculated fields for your Prep flows. Previously, creating formulas for calculated fields required an understanding of objects and fields, available functions and various limitations based on data types, formula types and more. Einstein greatly simplifies the task of creating custom functions. Einstein Copilot allows technical and non-technical users to easily create calculations using plain language. With the ability to quickly and efficiently create both row-level and summary formula fields from natural language descriptions, you can perform data preparation, making it easier and faster to combine, shape and clean data for analysis.

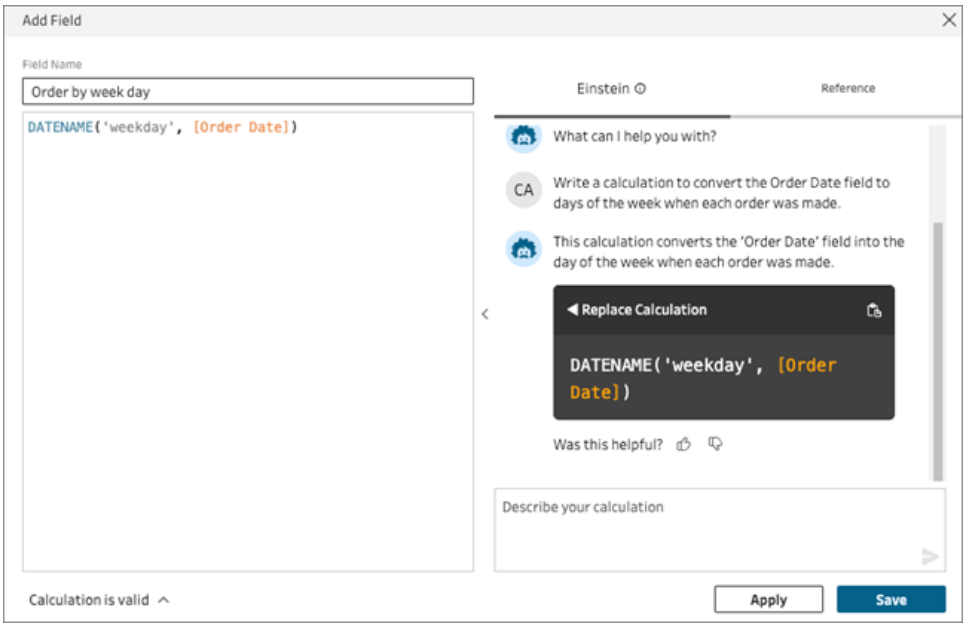
Note: You must have Tableau+ and a connection to a Salesforce org with Einstein generative AI set up to use Tableau Prep generative AI features. For more information, see Turn On Tableau AI for Your Site.

Create Calculated Fields

1. Click on a clean step in the Flow pane.
2. In the Profile pane toolbar, click **Create Calculated Field**, or in a profile card or data grid, click the **More options** ··· menu and select **Create Calculated Field** > **Custom Calculation**.



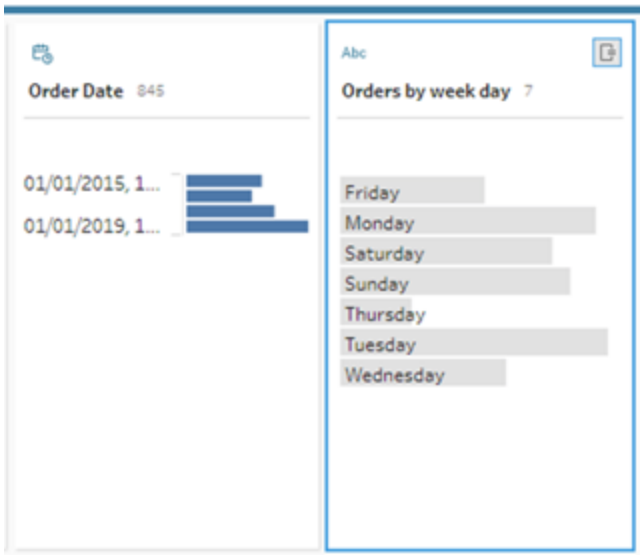
3. Provide a name for the field and in the Einstein text entry box, describe the calculation that you want. Provide as much detail as possible to try to get better results.
4. When your request is ready, click the arrow in the text box to pass it to Einstein.



Note: If Einstein can't provide a valid calculation, you can provide feedback.

5. Evaluate the output to make sure that it's what you want.

6. Click **Replace Calculation** to add the calculation to the editor. You can then edit the calculation and customise if needed. You can also add additional prompts to refine the calculation and then replace the original calculation.
7. Click **Apply** to create a field based on the calculation.
8. Click **Save** to save the field and close the calculation editor.

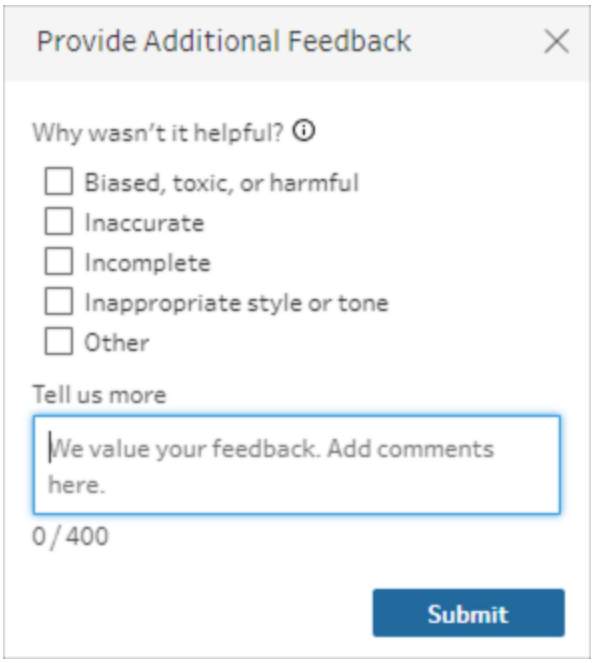


Guidelines for Writing Instructions

- Stick to basic instructions that result in calculation syntax.
- Use a verb-noun structure.
    - Preferred: Write a calculation to extract the characters from the field…
    - Not preferred: For extracting characters, how can I change the way…
- Use proper grammar and spelling.

Help Einstein Improve

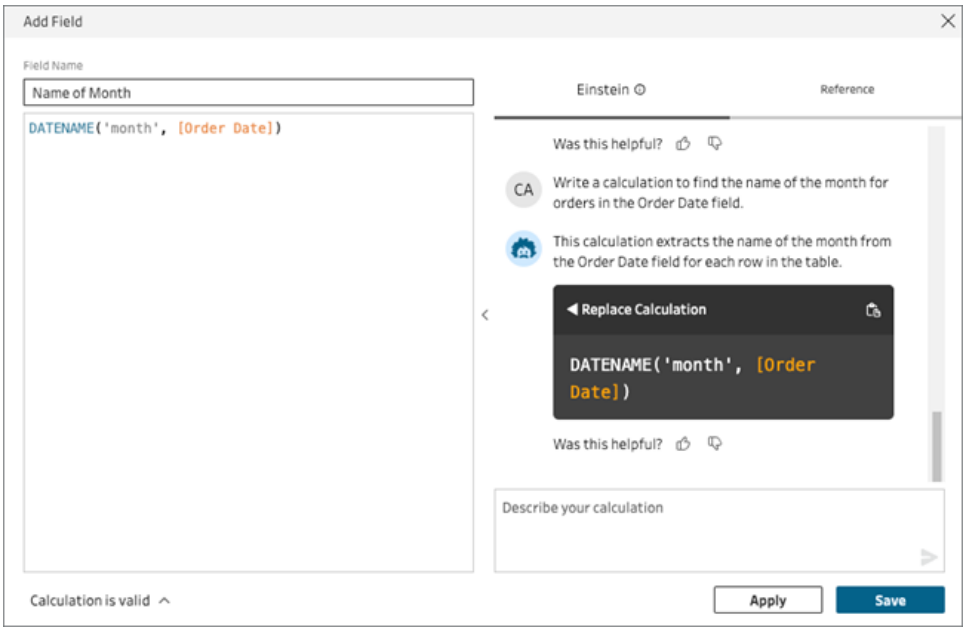
Einstein is still learning how to build accurate Tableau Prep calculations. Some calculations created by Einstein won't be written the way you expected. When this happens, your feedback about what's wrong with the calculation helps Einstein continue to improve and build increasingly accurate calculations.

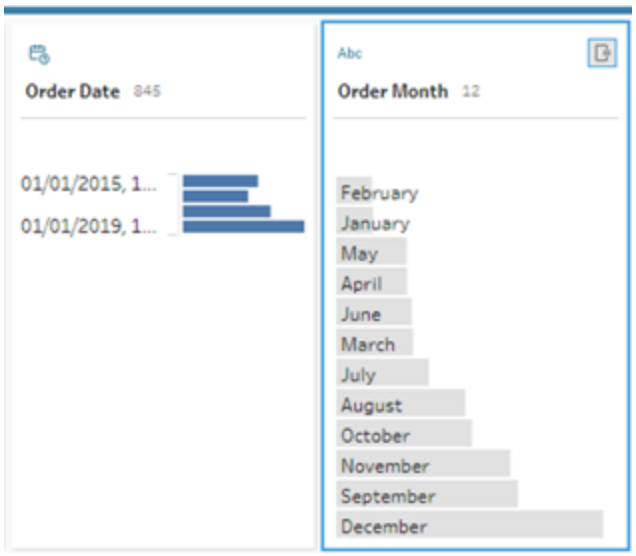


Examples

Example Calculation 1:

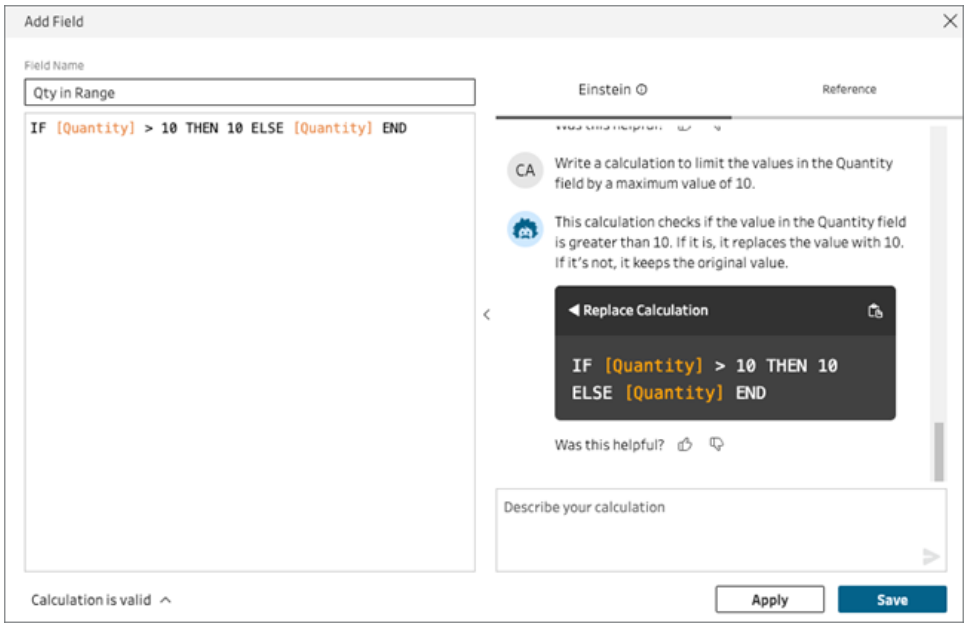
Prompt: *Write a calculation to find the name of the month for orders in the Order Date field.*





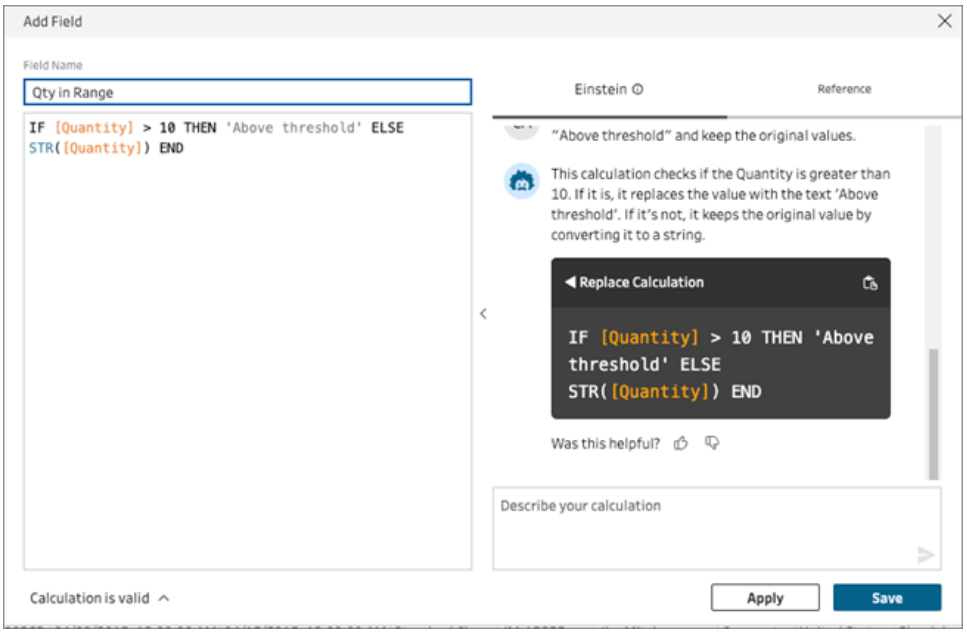
Example Calculation 2:

Prompt: *Write a calculation to limit the values in the Quantity field by a maximum value of 10.*



In this example, an additional prompt is added to refine the calculations to keep values that are above the threshold.

Prompt: *If the quantity is above the 10, replace it with the text "Above threshold" and keep the original values.*



Pivot Your Data

Note: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

Sometimes analysing 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.

However you may also have scenarios where your data tables are tall and narrow and are too normalised to properly analyse. For example, a sales department that tracks advertising spend in two columns, one called **Advertising** that contains rows for **radio**, **television** and **print** and one column for total spent. In this type of scenario, to analyse this data as separate measures you would need to pivot that row data to columns.

But what about pivoting larger data sets or data that changes frequently over time? You can use a wildcard pattern match to search for fields that match the pattern and automatically pivot the data.

Use one of the following options when pivoting your data:

- Pivot columns to rows
- Use wildcard search to instantly pivot fields based on a pattern match (Tableau Prep Builder version 2019.1.1 and later and on the web).
- Pivot rows to columns (Tableau Prep Builder version 2019.1.1 and later and on the web).

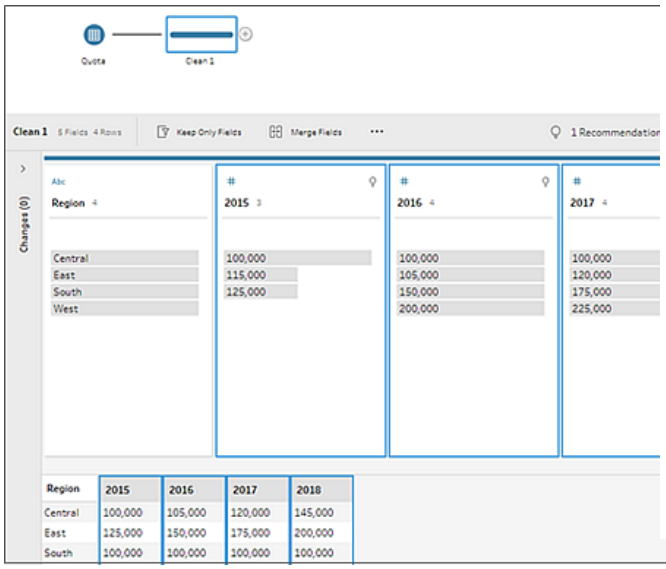
No matter how you pivot your fields, you can interact directly with the results and perform any additional cleaning operations to get your data looking just the way you want it. You can also use Tableau Prep's smart default naming feature to automatically rename your pivoted fields and values.

Pivot columns to rows

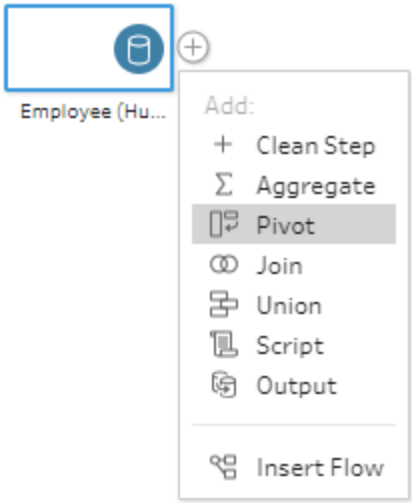
Use this pivot option to go from wide data to tall data. Pivot from columns to rows on one or more groups of fields. Select the fields that you want to work with and pivot the data from columns to rows.

1. Connect to your data source.
2. Drag the table that you want to pivot to the **Flow** pane.
3. Do one of the following:
    - **Tableau Prep Builder Version 2019.4.2 and later and on the web**: In the Profile pane, select the fields that you want to pivot, then right-click or Ctrl-click (MacOS) and select **Pivot Columns to Rows** from the menu. If using this option, skip to step 7.
    - **All versions**: Click the plus icon, and select **Add Pivot** from the context menu.

Select Fields (Tableau Prep Builder version 2019.4.2 and later and on the web)



Flow Step Menu (all versions)



4. (Optional) In the **Fields** pane, enter a value in the **Search** field to search the field list for fields to pivot.
5. (Optional) Select the **Automatically rename pivoted fields and values** tick 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.
6. Select one or more fields from the left pane, and drag them to the **Pivot1 Values** column in the **Pivoted Fields** pane.
7. (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. Your results appear immediately in both the **Pivot Results** pane and the data grid.

   > **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.

8. If you didn't enable the default naming option or if Tableau Prep couldn't automatically detect a name, edit the names of the fields. You can also edit the names of the original fields in this pane to best describe the data.

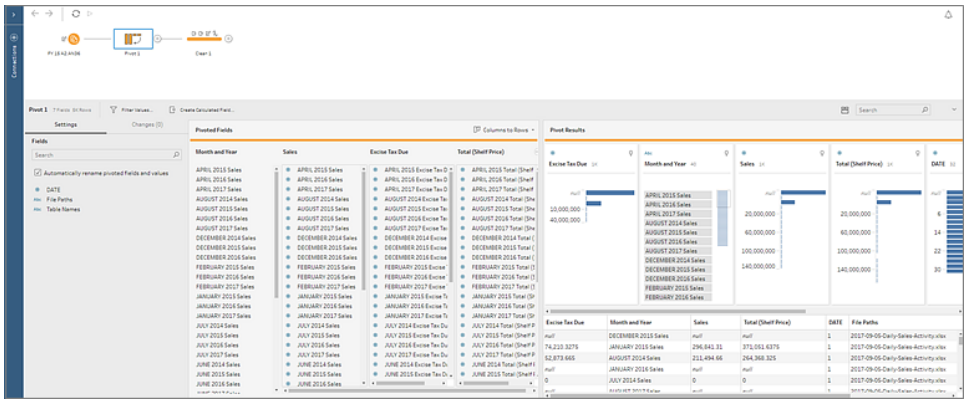
9. (Optional) Rename the new Pivot step to keep track of your changes. For example "Pivot months".
10. To refresh your pivot data when data changes, run your flow. If new fields are added to your data source that need to be added to the pivot, manually add them to the pivot.

Example: Pivoting on multiple fields

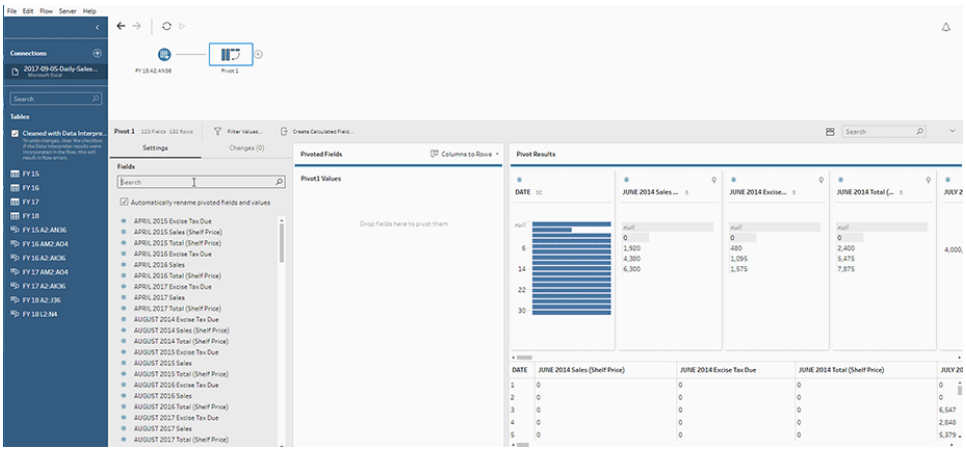
This example shows a spreadsheet for pharmaceutical sales, taxes and totals by month and year.

	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.

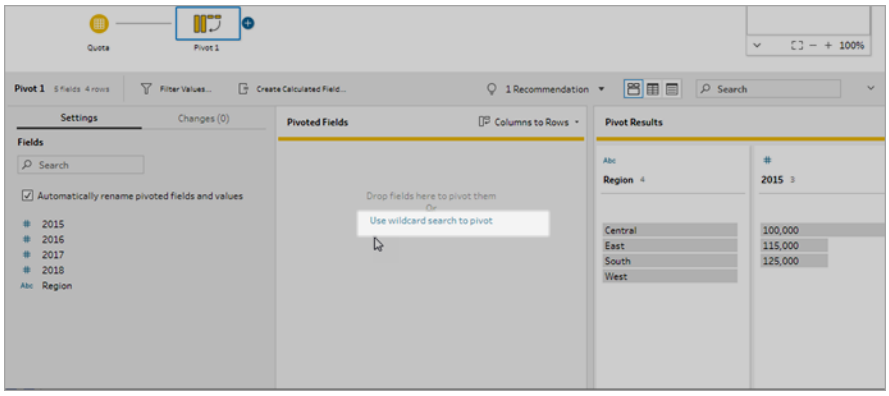


Use wildcard search to pivot

If you work with large data sets or if your data frequently changes over time, starting in Tableau Prep Builder version 2019.1.1 and on the web, you can use a wildcard search when pivoting columns to rows to instantly pivot your data based on a wildcard pattern match.

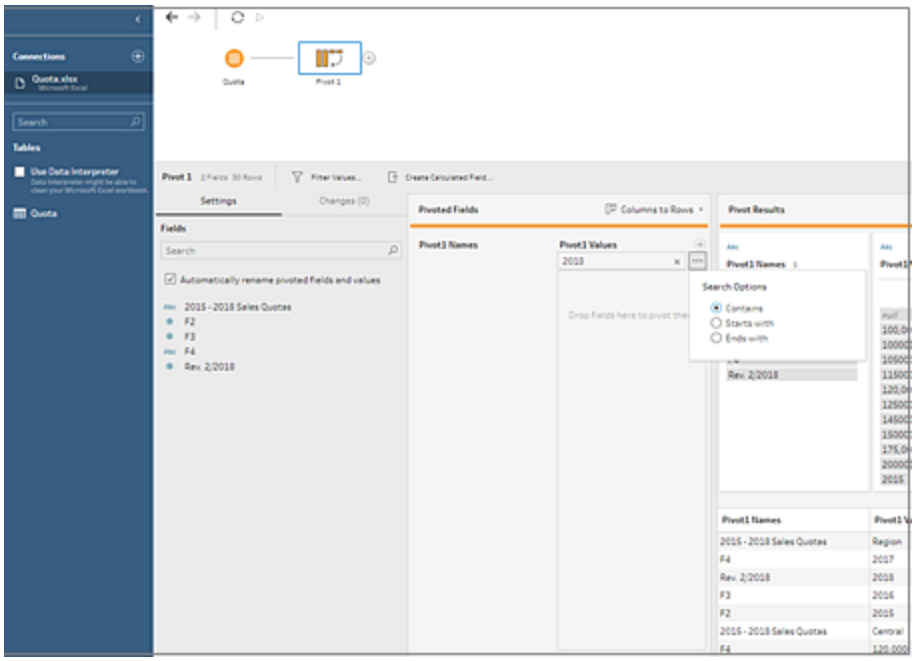
If new fields are added or removed that match the pattern, Tableau Prep detects the schema change when the flow is run and the pivot results are automatically updated.

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. In the **Pivoted Fields** pane, click on the link **Use wildcard search to pivot** .



5. Enter a value or partial value that you want to search for. For example, enter **Sales_** to match fields that are labelled as **sales_2017**, **sales_2018** and **sales_2019**.

   Do not use asterisks to match the pattern unless they are part of the field value that you are searching for. Instead, click the **Search Options** button to select how you want to match the value. Then press **Enter** to apply the search and pivot the matching values.



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.

7. If you didn't enable the default naming option or if Tableau Prep couldn't automatically detect a name, edit the names of the fields.
8. To refresh your pivot data when data changes, run your flow. Any new fields added to your data source that match the wildcard pattern are automatically detected and added to the pivot.
9. If the results aren't what you expect, try one of the following options:
    - Enter a different value pattern in the Search field and press enter. The pivot will automatically refresh and show the new results.
    - Manually drag additional fields to the **Pivot1 Values** column in the **Pivoted Fields** pane. You can also remove fields that were added manually by dragging them off the **Pivot1 Values** column and dropping them in the **Fields** pane.

    > **Note**: Fields that were added from the wildcard search results can't be removed by dragging them off the **Pivot1 Values** column. Instead try using a more specific pattern to match the search results you are looking for.

Pivot rows to columns

In Tableau Prep Builder version 2019.1.1 and later and on the web, pivot rows to columns if your data is too normalised and you need to create new columns – going from tall data to wider data.

For example if you have advertising costs for each month that includes all advertising types in one column, if you pivot the data from rows to columns you can then have a separate column for each advertising type instead, making the data easier to analyse.

Month	Cost	Advertising Type
January	12187	Radio
January	45698	Television
January	9762	Print
February	4598	Print
February	23987	Television
February	8907	Radio
March	10987	Print

Month	Radio	Television	Print
January	12187	45698	9762
February	8907	23987	4598
March	Null	Null	10987

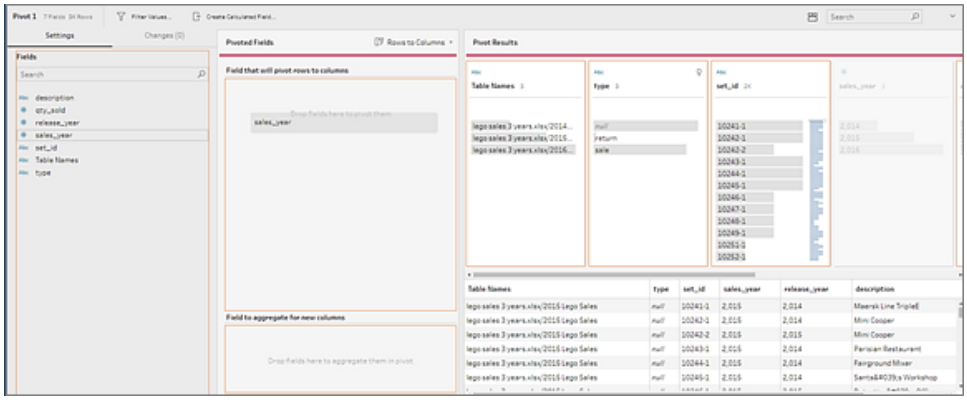
You can select one field to pivot on. The field values for that field are then used to create the new columns. Then, select a field to use to populate the new columns. These field values are aggregated and you can select the type of aggregation to apply.

Because aggregation is applied, pivoting columns back to rows won't reverse this pivot action. To reverse a row to column pivot type, you will need to undo the action. Either click the **Undo** button on the top menu, remove the fields from the **Pivoted Fields** pane or delete the pivot step.

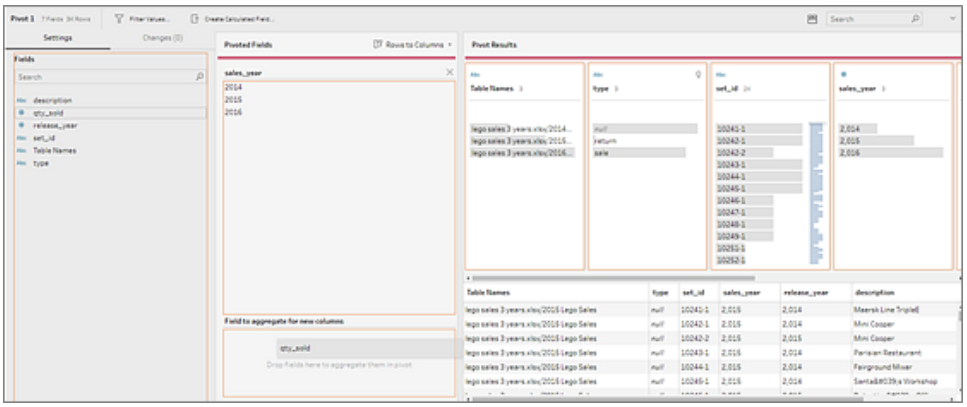
1. Connect to your data source.
2. Drag the table that you want to pivot to the **Flow** pane.
3. Click the plus icon, and select **Add Pivot** from the context menu.
4. In the **Pivoted Fields** pane, select **Rows to Columns** from the drop-down list.
5. (Optional) In the **Fields** pane, enter a value in the **Search** field to search the field list for fields to pivot
6. Select a field from the left pane, and drag it to the **Field that will pivot rows to columns** section in the **Pivoted Fields** pane.

   > **Note**: If the field you want to pivot on has a data type of date or datetime, you will need to change the data type to string to pivot it.

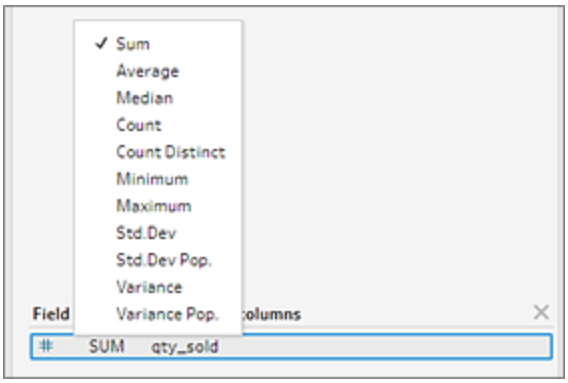
   The values in this field will be used to create and name the new columns. You can change the column names in the **Pivot Results** pane later.



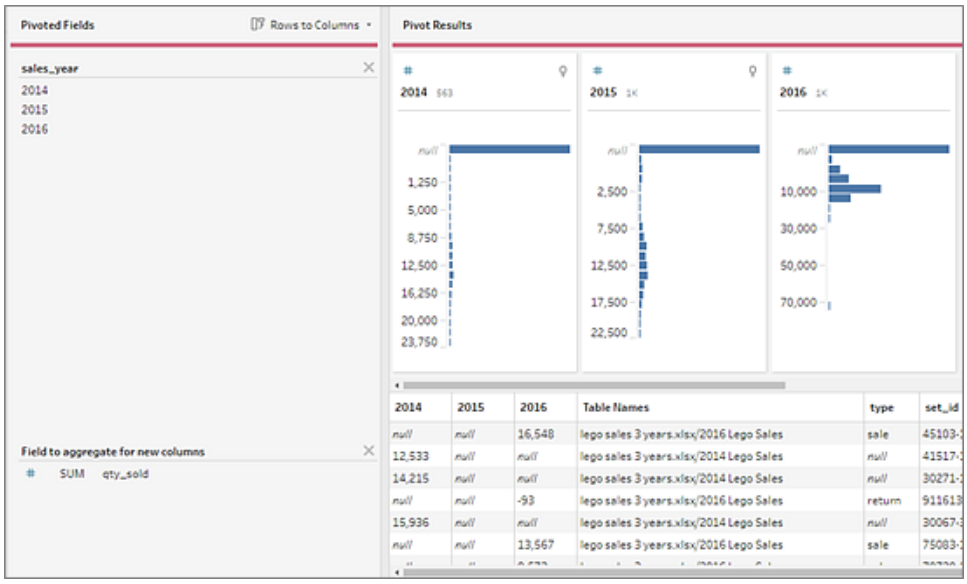
7. Select a field from the left pane and drag it to the **Field to aggregate for new columns** section in the **Pivoted Fields** pane. The values in this field are used to populate the new columns created from the previous step.



A default aggregation type is assigned to the field. Click the aggregation type to change it.



8. In the **Pivot Results** pane, review the results and apply any cleaning operations to the new columns that you created.

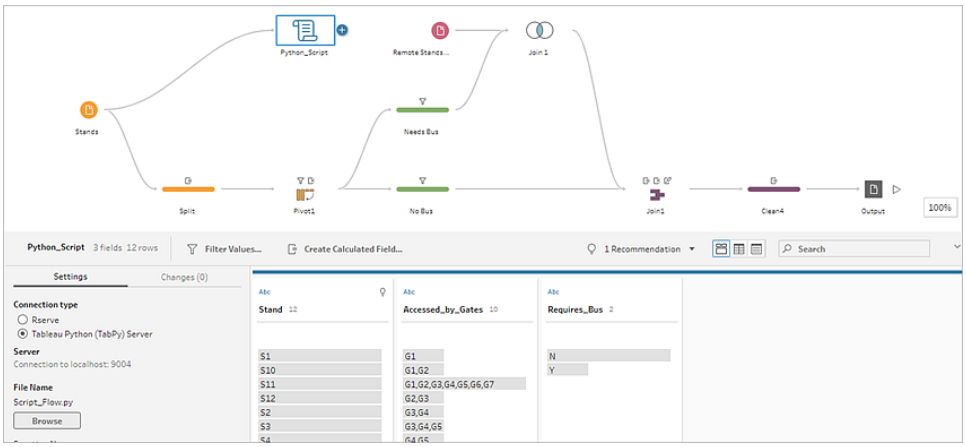


9. If the field being pivoted has a change in its row data, right-click or Ctrl-click (MacOS) on the Pivot step in the flow pane and select **Refresh**.

Use R and Python scripts in your flow

Starting in version 2019.3.1, you can use R and Python scripts to perform more complex cleaning operations or incorporate predictive modelling data into your flow. Data is passed from the flow as input through the R or Python script step, then returned as output data that you can continue cleaning using the features and functions of Tableau Prep Builder.

> **Note**: Connecting to scripts as an input step for your flow is not yet supported. Also, script steps are not yet supported for flows authored or published to Tableau Cloud.



Configure your Rserve server or Tableau Python (TabPy) server and add a script step to your flow. Tableau Prep passes the data to Rserve for R or Tableau Python server (TabPy) for Python and returns the resulting data back to the flow in the form of a table. You can continue to apply cleaning operations to the results and generate your output for analysis.

When you create your script, you will need to include a function that specifies a data frame as an argument of the function. If you want to return different fields than what you input, you'll need to include a getOutputSchema function in your script that defines the output and data types. Otherwise, the output will use the fields from the input data.

If you author or edit flows in Tableau Server (version 2020.4.1 and later) that include script steps, Tableau Server must also have a connection to an Rserve or TabPy server to run script steps. For information about how to configure R or Python to use in your flows and how to create your scripts, see **Use R (Rserve) scripts in your flow** below or **Use Python scripts in your flow** on page 350.

Use R (Rserve) scripts in your flow

> **Disclaimer:** This topic includes information about a third-party product. Please note that while we make every effort to keep references to third-party content accurate, the information we provide here might change without notice as R and Rserve changes. For the most up-to-date information, please consult the R and Rserve documentation and support.

R is an open source software programming language and a software environment for statistical computing and graphics. To extend the functionality of Tableau Prep Builder, you can create

scripts in R to use in your flow that run through an Rserve server to produce output that you can further work with in your flow.

For example, you might want to add statistical modelling data or forecasting data to the data that you already have in your flow using a script in R, then use the power of Tableau Prep Builder to clean the resulting data set for analysis.

To include R scripts in your flow, you need to configure a connection between Tableau Prep Builder and an Rserve server. Then you can use R scripts to apply supported functions to data from your flow using R expressions. After you enter the configuration details and point Tableau Prep Builder to the file and function that you want to use, data is securely passed to the Rserve server, the expressions are applied, and the results are returned as a table (R data.frame) that you can clean or output as needed.

You can run flows that include script steps in Tableau Server as long as you have configured a connection to your Rserve server. Running flows with script steps in Tableau Cloud, isn't currently supported. To configure Tableau Server, see **Configure Rserve Server for Tableau Server** below.

Prerequisites

To include R script steps in your flow, install R and configure a connection to an Rserve server.

Resources

- Download and Install R. Download and install the most current version of R for Linux, Mac or Windows.
- R Implementation notes (community post). Install and configure a connection to R and Rserve for Windows.
- Install and configure Rserve: Instructions for general installation and configuration for all platforms.
    - Rserve for Windows (release notes): This topic covers limitations when installing Rserve locally on Windows.

Configure Rserve Server for Tableau Server

Use the following instructions to configure a connection between your Rserve server and Tableau Server.

- **Version 2019.3 and later**: You can run published flows that include script steps in Tableau Server.

- **Version 2020.4.1 and later**: You can create, edit, and run flows that include script steps in Tableau Server.
- **Tableau Cloud**: Creating or running flows with script steps isn't currently supported.

1. Open the TSM command line.
2. Enter the following commands to set the host address, port values and connect timeout:

   ```
   tsm security maestro-rserve-ssl enable --connection-type
   {maestro-rserve-secure/maestro-rserve} --rserve-host <Rserve IP
   address or host name> --rserve-port <Rserve port> --rserve-
   username <Rserve username> --rserve-password <Rserve password> --
   rserve-connect-timeout-ms <RServe connect timeout>
   ```

   - Select `{maestro-rserve-secure}` to enable a secure connection or `{maestro-rserve}` to enable an unsecured connection.
   - If you select `{maestro-rserve-secure}`, specify the certificate file `-cf<certificate file path>` in the command line.
   - Specify the `--rserve-connect-timeout-ms <RServe connect timeout>` in milliseconds. For example `--rserve-connect-timeout-ms 900000`.
3. To disable the Rserve connection enter the following command

   ```
   tsm security maestro-rserve-ssl disable
   ```

Additional Rserve configuration (optional)

You can create a file named Rserv.cfg to set default configuration values to customise Rserve and place it in the `/etc/Rserve.conf` installation location. To improve stability with the Rserve server and Tableau Prep Builder, you can add additional values to your Rserve configuration. When you launch Rserve you can refer to this file to apply your configuration options. For example:

- Windows: `Rserve(args="--RS-conf C:\\folder\\Rserv.cfg")`
- MacOS and Linux: `Rserve(args=" --no-save --RS-conf ~/Documents/Rserv.cfg")`

The following example shows some additional options you can include in your `Rserve.conf` configuration file:

```
# If your data includes characters other than ASCII, make it explicit
that data should be UTF8 encoded.
```

```
encoding utf8
# Disable interactive behavior for Rserve or Tableau Prep Builder will
stall when trying to run the script as it waits for an input response.
interactive no
```

For information about setting up an Rserve.conf file, see the Advanced Rserve configuration section in the R Implementation notes (community post).

Create your R script

When you create your script, include a function that specifies a data frame as an argument of the function. This will call your data from Tableau Prep Builder. You will also need to return the results in a data frame using supported data types.

For example:

```
postal_cluster <- function(df) {
  out <- kmeans(cbind(df$Latitude, df$Longitude), 3, iter.max=10)
  return(data.frame(Latitude=df$Latitude, Longitude=df$Longitude,
Cluster=out$cluster))
 }
```

The following data types are supported:

Data type in Tableau Prep Builder	Data type in R
String	Standard UTF-8 string
Decimal	Double
Int	Integer
Bool	Logical
Date	String in ISO_DATE format “YYYY-MM-DD” with optional zone offset. For example, “2011-12-03+01:00” is a valid date.
DateTime	String in ISO_DATE_TIME format “YYYY-MM-DDT:HH:mm:ss” with optional zone offset. For example, “2011-12-03T10:15:30+01:00” is a vslid date.

> **Note**: Date and DateTime must always be returned as a valid string. Native Date (DateTime) types in R aren't supported as returned values but can be used in the script.

If you want to return different fields than what you input, you'll need to include a getOutputSchema function in your script that defines the output and data types. Otherwise, the output will use the fields from the input data, which are taken from the step that is prior to the script step in the flow.

Use the following syntax when specifying the data types for your fields in the getOutputSchema:

Function in R	Resulting data type
prep_string ()	String
prep_decimal ()	Decimal
prep_int ()	Integer
prep_bool ()	Boolean
prep_date ()	Date
prep_datetime ()	DateTime

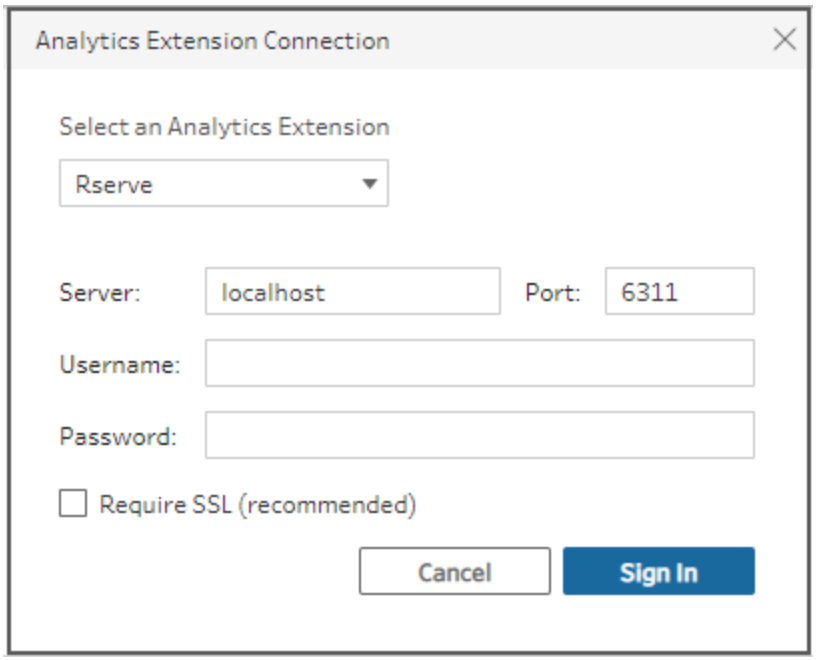
The following example shows the getOutputSchema function for the postal_cluster script:

```
getOutputSchema <- function() {
  return (data.frame (
    Latitude = prep_decimal (),
    Longitude = prep_decimal (),
    Cluster = prep_int ())));
}
```

Connect to your Rserve server

Important: Starting in Tableau Prep Builder version 2020.3.3, you can configure your server connection once from the top **Help** menu instead of setting up your connection per flow in the Script step by clicking **Connect to Rserve Server** and entering your connection details. You will need to reconfigure your connection using this new menu for any flows that were created in an older version of Tableau Prep Builder that you open in version 2020.3.3.

1. Select **Help** > **Settings and Performance** > **Manage Analytics Extension Connection**.
2. In the **Select an Analytics Extension** drop-down list, select **Rserve**.



3. Enter your credentials:
   - Port 6311 is the default port for plaintext Rserve servers.
   - Port 4912 is the default port for SSL-encrypted Rserve servers.
   - If the server requires credentials, enter a Username and Password.
   - If the server uses SSL encryption, tick the **Require SSL** box, then click the **Custom configuration file** link to specify a certificate for the connection.

     Note: Tableau Prep Builder doesn't provide a way to test the connection. If there is a problem with the connection, an error message shows when you try and run the flow.

Add a script to your flow

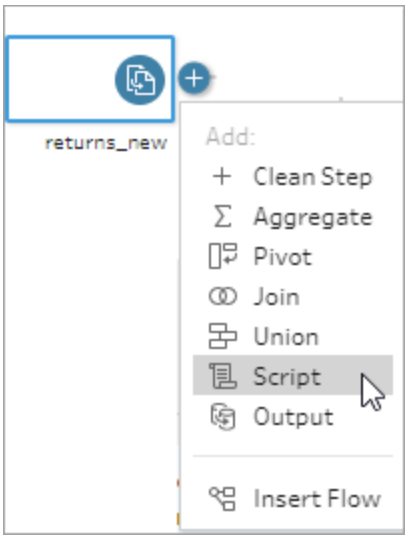
Start your Rserve server then complete the following steps:

1. Open Tableau Prep Builder click the **Add connection** button.

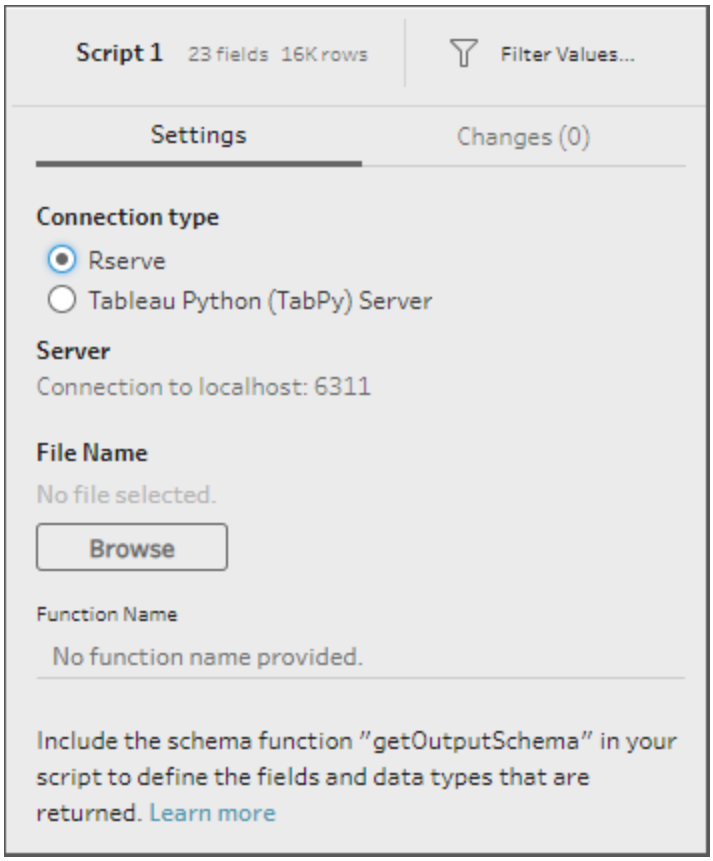
   In web authoring, from the **Home** page, click **Create** > **Flow** or from the **Explore** page, click **New** > **Flow**. Then click **Connect to Data**.

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.

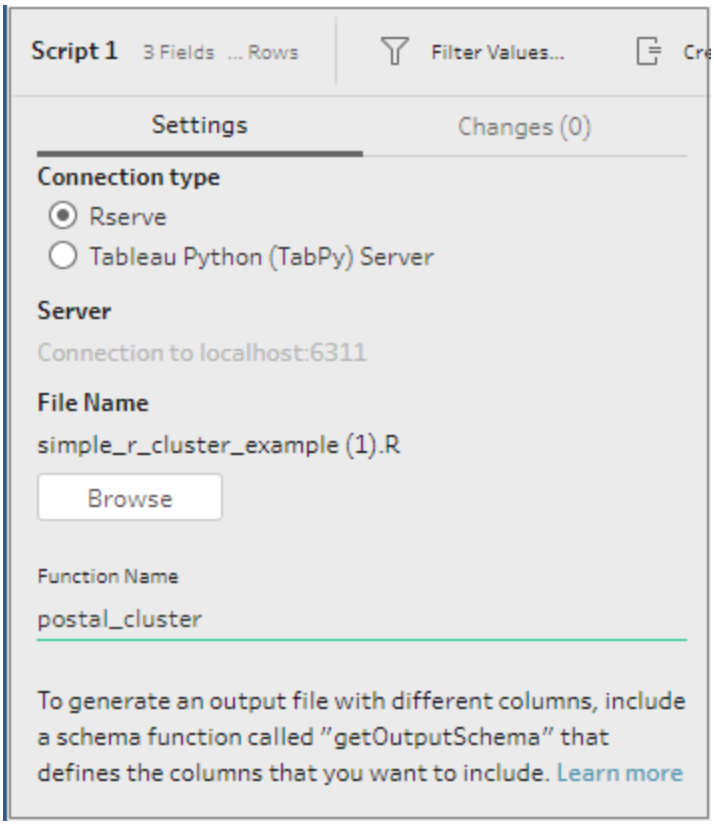
3. Click the plus icon, and select **Add Script** from the context menu.



4. In the Script pane, under **Connection type** , select **Rserve**.



5. In the **File Name** section, click **Browse** to select your script file.
6. Enter the **Function Name** then press **Enter** to run your script.



Use Python scripts in your flow

Disclaimer: This topic includes information about a third-party product. Please note that while we make every effort to keep references to third-party content accurate, the information we provide here might change without notice as python changes. For the most up-to-date information, please consult the python documentation and support.

Python is a widely used high-level programming language for general-purpose programming. By sending Python commands to an external service through Tableau Prep Builder, you can easily extend your data preparation options by performing actions like adding row numbers, ranking fields, filling down fields and performing other cleaning operations that you might otherwise do using calculated fields.

To include Python scripts in your flow, you need to configure a connection between Tableau and a TabPy server. Then you can use Python scripts to apply supported functions to data from your flow using a pandas dataframe. When you add a script step to your flow and specify the configuration details, file and function that you want to use, data is securely passed to the TabPy

server, the expressions in the script are applied, and the results are returned as a table that you can clean or output as needed.

You can run flows that include script steps in Tableau Server as long as you have configured a connection to your TabPy server. Running flows with script steps in Tableau Cloud, isn't currently supported. To configure Tableau Server, see **Configure the Tableau Python (TabPy) server for Tableau Server** below.

For information about how to configure sites on Tableau Server with analytics extensions for workbooks, see Configure Connections with Analytics Extensions.

Prerequisites

To include Python scripts in your flow, complete the following setup. Creating or running flows with script steps in Tableau Cloud isn't currently supported.

1. Download and install Python. Download and install the most current version of Python for Linux, Mac or Windows.
2. Download and install the Tableau Python server (TabPy). Follow the installation and configuration instructions for installing TabPy. Tableau Prep Builder uses TabPy to pass data from your flow through TabPy as the input, applies your script, then returns the results back to the flow.
3. Install Pandas. Run `pip3 install pandas`. You must use a pandas data frame in your scripts to integrate with Tableau Prep Builder.

Configure the Tableau Python (TabPy) server for Tableau Server

If you plan to publish, create, edit and run flows that include script steps in Tableau Server, you will need to configure a connection between your TabPy server and Tableau Server.

- **Version 2019.3 and later**: You can run published flows that include script steps in Tableau Server.
- **Version 2020.4.1 and later**: You can create, edit and run flows that include script steps in Tableau Server.
- **Tableau Cloud**: Creating or running flows with script steps isn't currently supported.

1. Open the TSM command line/shell .
2. Enter the following commands to set the host address, port values and connect timeout:

```
tsm security maestro-tabpy-ssl enable --connection-type {maestro-
tabpy-secure/maestro-tabpy} --tabpy-host <TabPy IP address or
```

host name> --tabpy-port <TabPy port> --tabpy-username <TabPy username> --tabpy-password <TabPy password> --tabpy-connect-timeout-ms <TabPy connect timeout>

- Select {maestro-tabpy-secure} to enable a secure connection or {maestro-tabpy} to enable an unsecured connection.
- If you select {maestro-tabpy-secure}, specify the certificate file -cf<certificate file path> in the command line.
- Specify the --tabpy-connect-timeout-ms <TabPy connect timeout> in milliseconds. For example --tabpy-connect-timeout-ms 900000.

3. To disable the TabPy connection enter the following command

```
tsm security maestro-tabpy-ssl disable
```

Create your python script

When you create your script, include a function that specifies a pandas (pd.DataFrame) as an argument of the function. This will call your data from Tableau Prep Builder. You will also need to return the results in a pandas (pd.DataFrame) using supported data types.

For example to add encoding to a set of fields in a flow, you could write the following script:

```
def encode(input):
  le = preprocessing.LabelEncoder()
  Return pd.DataFrame({
    'Opportunity Number' : input['Opportunity Number'],
    'Supplies Subgroup Encoded' : le.fit_transform(input['Supplies
Subgroup']),
    'Region Encoded' : le.fit_transform(input['Region']),
    'Route To Market Encoded' : le.fit_transform(input['Route To Mar-
ket']),
    'Opportunity Result Encoded' : le.fit_transform(input
['Opportunity Result']),
    'Competitor Type Encoded' : le.fit_transform(input['Competitor
Type']),
    'Supplies Group Encoded' : le.fit_transform(input['Supplies
Group']),
})
```

The following data types are supported:

Data type in Tableau Prep Builder	Data type in Python
String	Standard UTF-8 string
Decimal	Double
Int	Integer
Bool	Boolean
Date	String in ISO_DATE format “YYYY-MM-DD” with optional zone offset. For example, “2011-12-03” is a valid date.
DateTime	String in ISO_DATE_TIME format “YYYY-MM-DDT:HH:mm:ss” with optional zone offset. For example, “2011-12-03T10:15:30+01:00” is a vslid date.

Note: Date and DateTime must always be returned as a valid string.

If you want to return different fields than what you input, you'll need to include a get_output_schema function in your script that defines the output and data types. Otherwise, the output will use the fields from the input data, which are taken from the step that is prior to the script step in the flow.

Use the following syntax when specifying the data types for your fields in the get_output_schema:

Function in Python	Resulting data type
prep_string()	String
prep_decimal()	Decimal
prep_int()	Integer
prep_bool()	Boolean
prep_date()	Date
prep_datetime()	DateTime

Square brackets can be used to access rows from a DataFrame. Single brackets output a Pandas Series, while a double bracket will output a Pandas DataFrame.

```
def create_key(df):
return pd.DataFrame({
"Key": ['12345']
})
```

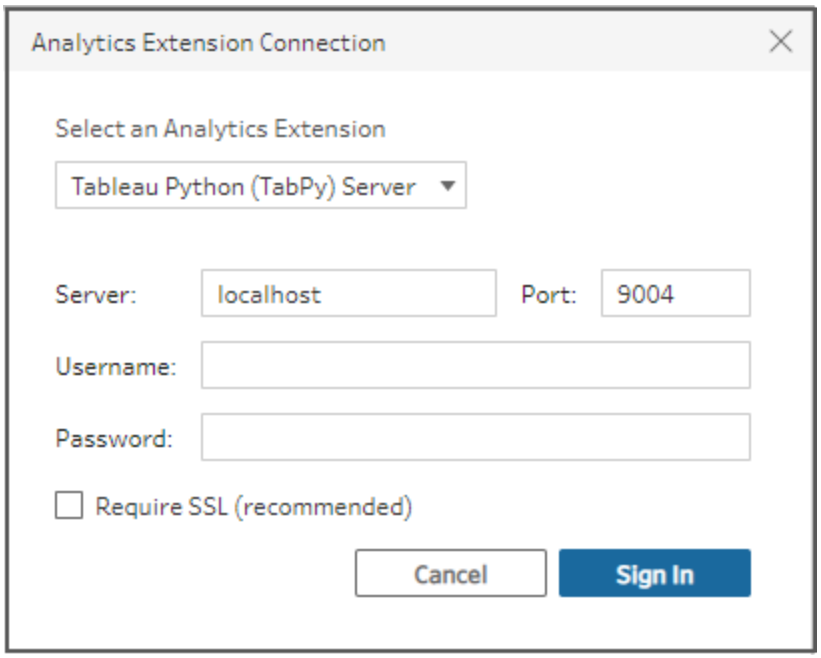
The following example shows the get_output_schema function added to the field encoding python script:

```
def get_output_schema():
  return pd.DataFrame({
    'Opportunity Number' : prep_int(),
    'Supplies Subgroup Encoded' : prep_int(),
    'Region Encoded' : prep_int(),
    'Route To Market Encoded' : prep_int(),
    'Opportunity Result Encoded' : prep_int(),
    'Competitor Type Encoded' : prep_int(),
    'Supplies Group Encoded' : prep_int()
})
```

Connect to your Tableau Python (TabPy) server

Important: Starting in Tableau Prep Builder version 2020.3.3, you can configure your server connection once from the top **Help** menu instead of setting up your connection per flow in the Script step by clicking **Connect to Tableau Python (TabPy) Server** and entering your connection details. You will need to reconfigure your connection using this new menu for any flows that were created in an older version of Tableau Prep Builder that you open in version 2020.3.3.

1. Select **Help** > **Settings and Performance** > **Manage Analytics Extension Connection**.
2. In the **Select an Analytics Extension** drop-down list, select **Tableau Python (TabPy) Server**.



3. Enter your credentials:
   - Port 9004 is the default port for TabPy.
   - If the server requires credentials, enter a username and password.
   - If the server uses SSL encryption, select the Require SSL tick box, then click the **No custom configuration file specified** link to select a certificate for the connection. This is your SSL server certificate file.

     > **Note**: Tableau Prep Builder doesn't provide a way to test the connection. If there is a problem with the connection an error message shows.

Add a script to your flow

Start your TabPy server then complete the following steps:

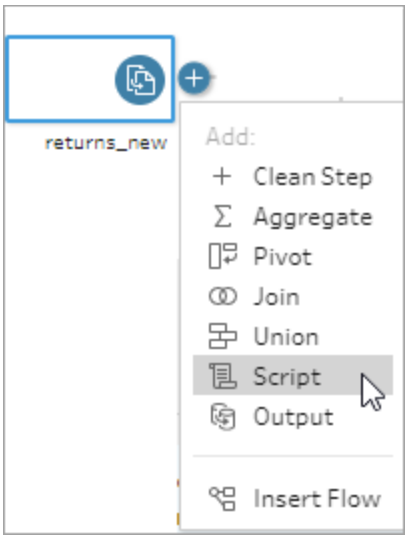
> **Note**: TabPy requires tornado package version 5.1.1 to run. If you receive the error 'tornado.web' has no attribute 'asynchronous' when trying to start TabPy, from the command line run `pip list` to check the version of tornado that was installed. If you have a different version installed, download the tornado package version 5.1.1. Then run `pip uninstall tornado` to uninstall your current version, then run `pip install tornado==5.1.1` to install the required version.

1. Open Tableau Prep Builder and click the **Add connection** button.

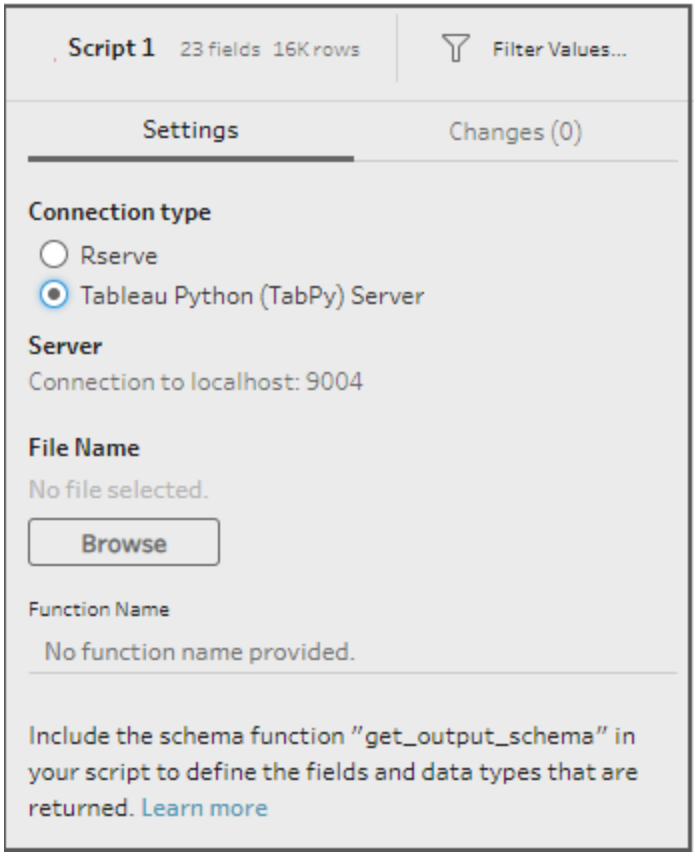
   In web authoring, from the **Home** page, click **Create** > **Flow** or from the **Explore** page, click **New** > **Flow**. Then click **Connect to Data**.

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.

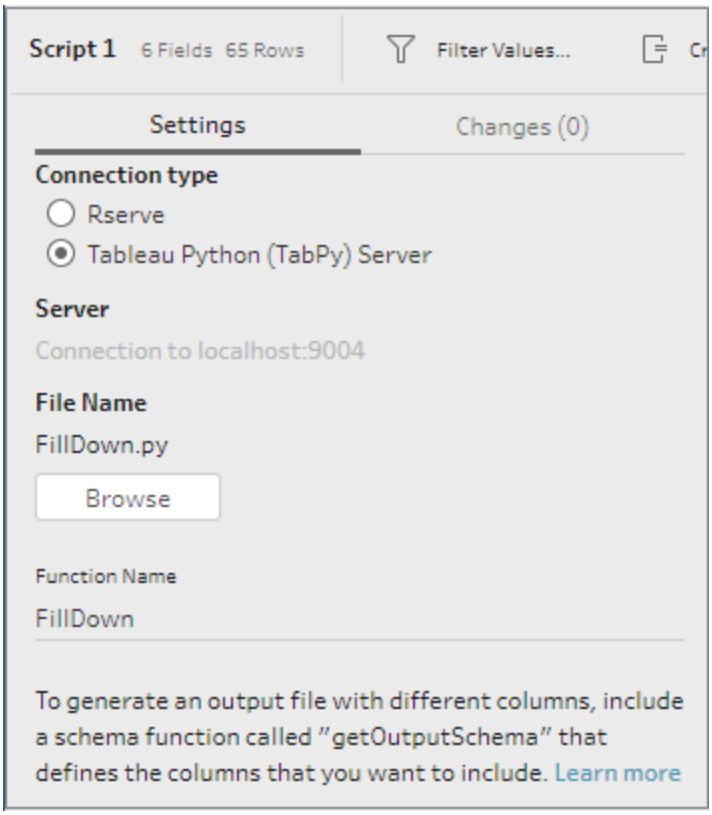
3. Click the plus icon, and select **Add Script** from the context menu.



4. In the Script pane, in the **Connection type** section, select **Tableau Python (TabPy) Server**.



5. In the **File Name** section, click **Browse** to select your script file.
6. Enter the **Function Name** then press **Enter** to run your script.


Script 1
6 Fields 65 Rows
Filter Values...
Settings
Changes (0)
Connection type
Rserve
Tableau Python (TabPy) Server
Server
Connection to localhost:9004
File Name
FillDown.py
Browse
Function Name
FillDown
To generate an output file with different columns, include a schema function called "getOutputSchema" that defines the columns that you want to include. Learn more

Aggregate, Join or Union Data

Aggregate, join, or union your data to group or combine data for analysis.

> **Note**: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

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 **Aggregate** option to create a step to group and aggregate data. Whether data is aggregated or grouped depends on the data type (string, number or date).

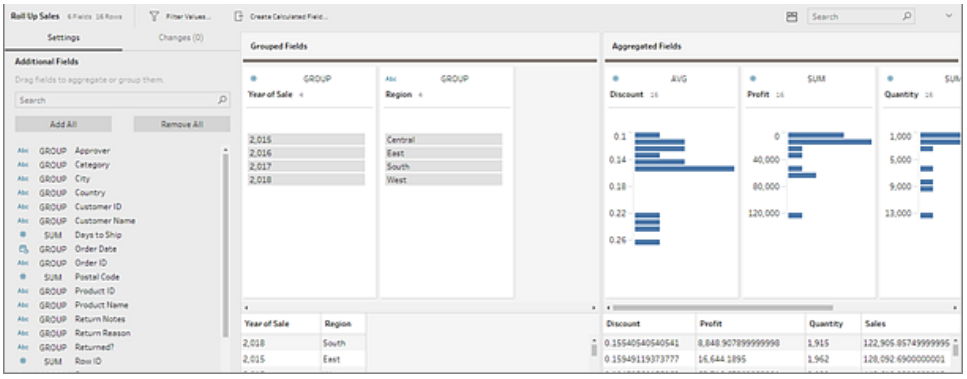
1. In the **Flow** pane, click the plus icon, and select **Aggregate**. A new aggregation step displays in the **Flow** pane and the **Profile** pane updates to show the aggregate and group profile.
2. Drag fields from the left pane to the **Grouped Fields** pane (the fields that make the row) or to the **Aggregated Fields** pane (the data that will be aggregated and presented at the level of the grouped fields).

   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.

- Apply certain cleaning operations to fields. For more information abut which cleaning options are available, see **About cleaning operations** on page 227.

The following example would show the aggregated sum of profit and quantity, as well as average discount by region and year of sale.



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.

Any cleaning operations that are made to the fields are tracked in the **Changes** pane.

Join your data

The data that you want to analyse 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.

Tableau Prep supports the following join types:

Join Type	Description
Left	For each row, includes all values from the left table and corresponding matches from the right table. When a value in the left table doesn't have a cor-

	responding match in the right table, you see a null value in the join results.
Inner	For each row, includes values that have matches in both tables.
Right	For each row, includes all values from the right table and corresponding matches from the left table. When a value in the right table doesn't have a corresponding match in the left table, you see a null value in the join results.
leftOnly	For each row, includes only values from the left table that don't match any values from the right table. Field values from the right table show as null values in the join results.
rightOnly	For each row, includes only values from the right table that don't match any values from the left table. Field values from the left table show as null values in the join results.
notInner	For each row, includes all of the values from the right and the left table that don't match.
Full	For each row, includes all values from both tables. When a value from either table doesn't have a match with the other table, you see a null value in the join results.

To create a join, do the following:

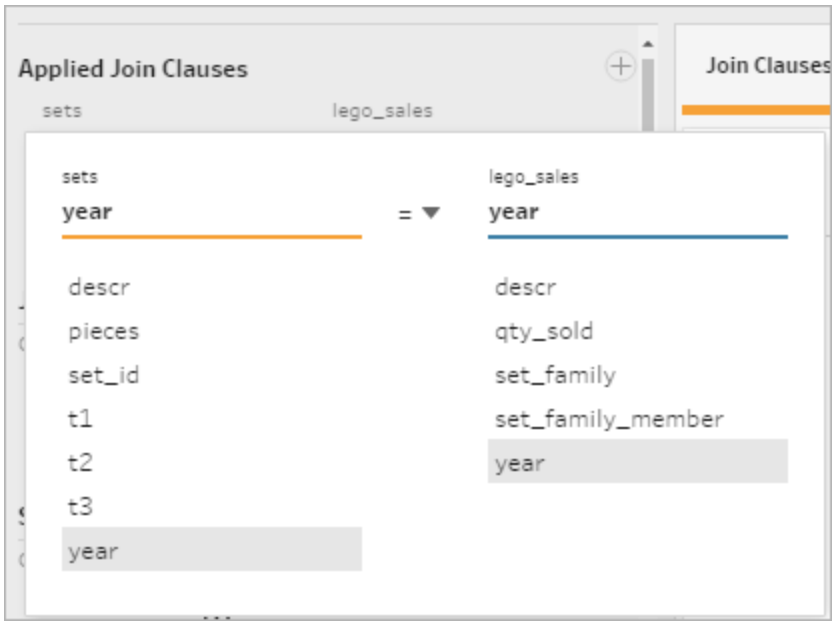
1. Join two tables using one of the following methods:
    - Add at least two tables to the **Flow** pane, then select and drag the related table to the other table until the **Join** option displays.

- Click the ⊕ icon and select **Join** from the menu, then manually add the other input to the join and add the join clauses.

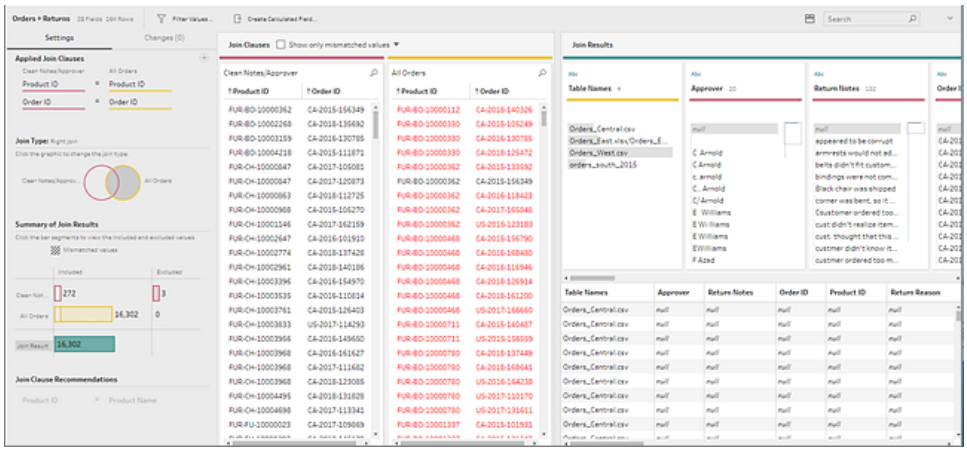
  > **Note**: If you connect to a table that has table relationships defined and includes related fields, you can select **Join** and select from a list of related tables. Tableau Prep creates the join based on the fields that make up the relationship between the two tables.
  >
  > For more information about connectors with table relationships, see **Join data in the Input step** on page 141.

A new join step is added to the flow and the profile pane updates to show the join profile.

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. You can also click the recommended join clauses shown under **Join Clause Recommendations** to add the clause to the list of applied join clauses.

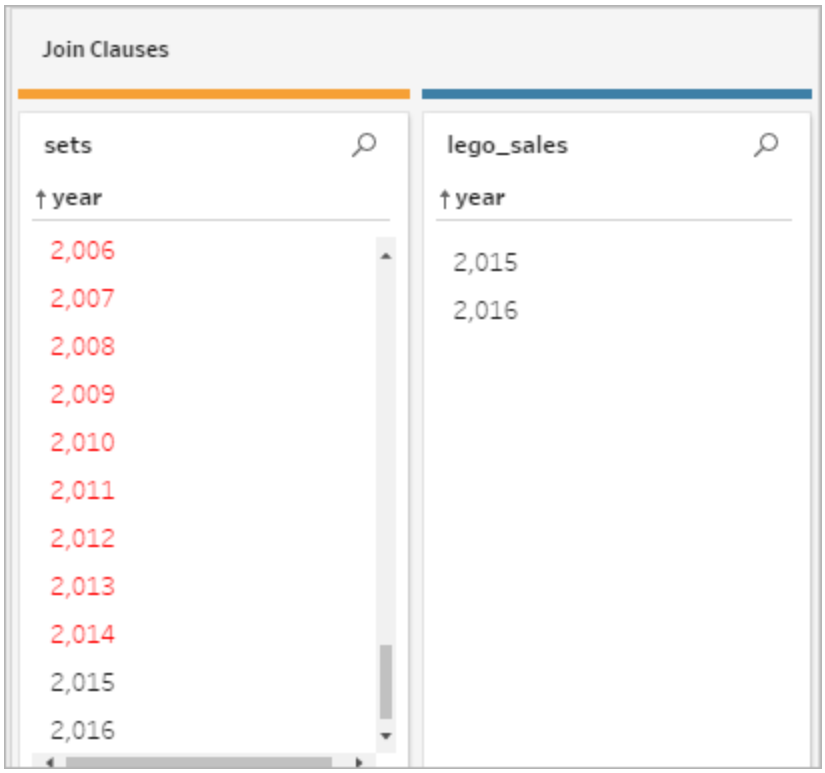


Inspect the results of the join

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, leftOnly, rightOnly, notInner or full 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.

Common join issues

If you don't see the results you expect after joining your data, you may need to do some additional cleaning on your field values. The following issues will result in Tableau Prep reading the values as not matching and exclude them from the join:

- Different capitalisation: My Sales and my sales
- Different spelling: Hawaii and Hawai'i
- Mispelling or data entry errors: My Company Health and My Company Heath
- Name changes: John Smith and John Smith Jr.
- Abbreviations: My Company Limited and My Company Ltd
- Extra separators: Honolulu and Honolulu (Hawaii)
- Extra spaces: This includes extra space between characters, tabbed spaces or extra

leading or trailing spaces

- Inconsistent use of periods: Returned, not needed. and Returned, not needed.

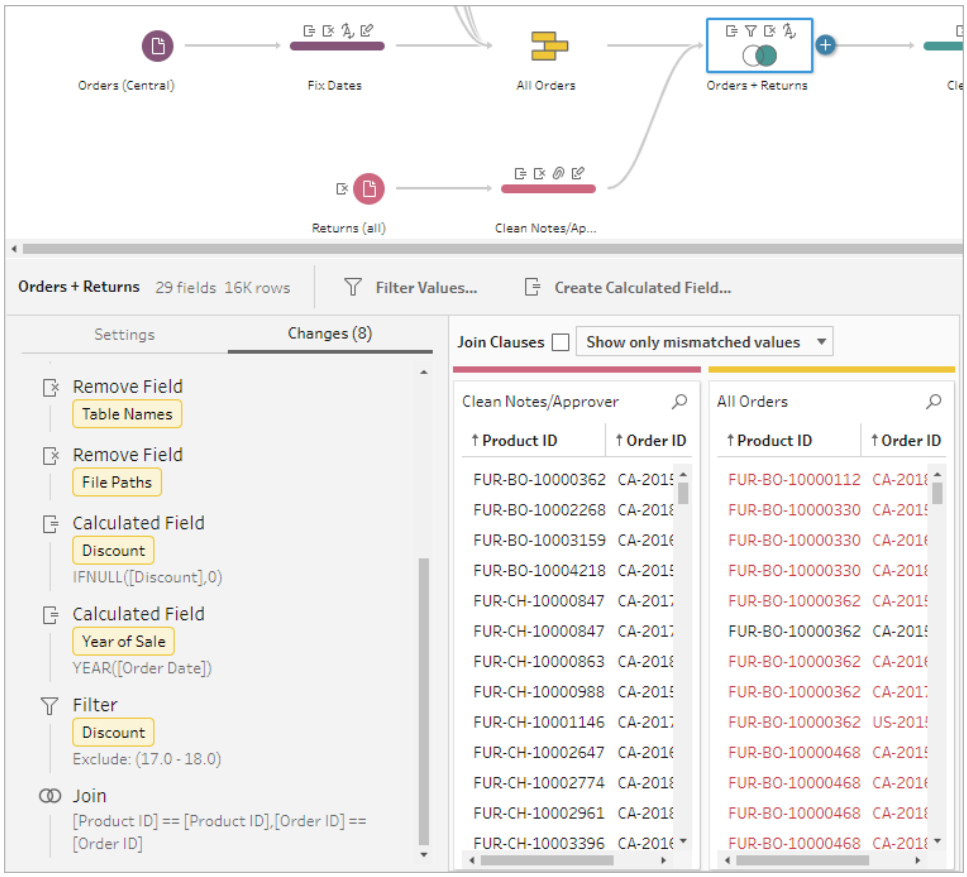
The good news is that if your field values have any of these issues, you can fix the field values directly in the **Join Clauses** or work with excluded values by clicking in the **Excluded** bars in the **Summary of Join Results** and use the cleaning operations in the profile card menu.

For more information about the different cleaning options available in the Join step, see **About cleaning operations** on page 227.

Fix mismatched fields and more

You can fix mismatched fields right in the join clause. Double-click or right-click on 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 tracked and added to the **Changes** pane right in the Join step.

You can also select multiple values to keep, exclude or filter in the Join Clauses panes, or apply other cleaning operations in the Join Results pane. Depending on which fields you change and where they are in the join process, your change is applied either before or after the join to give you the corrected results.



For more information about cleaning fields see **Apply cleaning operations** on page 231.

Union your data

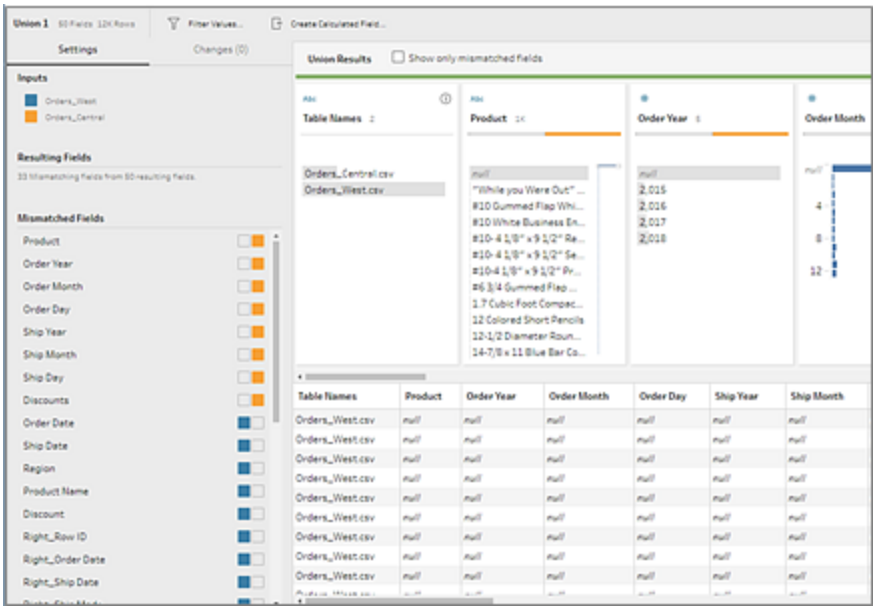
Union is a method for combining data by appending rows of one table onto another table. For example, you might want to add new transactions in one table to a list of past transactions in another table. Make sure the tables you union have the same number of fields, the same field names, and the fields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

Tip: To maximise performance a single union can have a maximum of 10 inputs. If you need to union more than 10 files or tables, try unioning files in the Input step. For more information about this type of union, see **Union files and database tables in the Input step** on page 130.

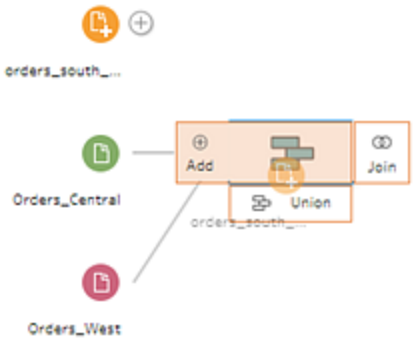
Similar to a join, you can use the union operation anywhere in the flow.

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 **Union** option. You can also click the ⊕ icon and select **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.

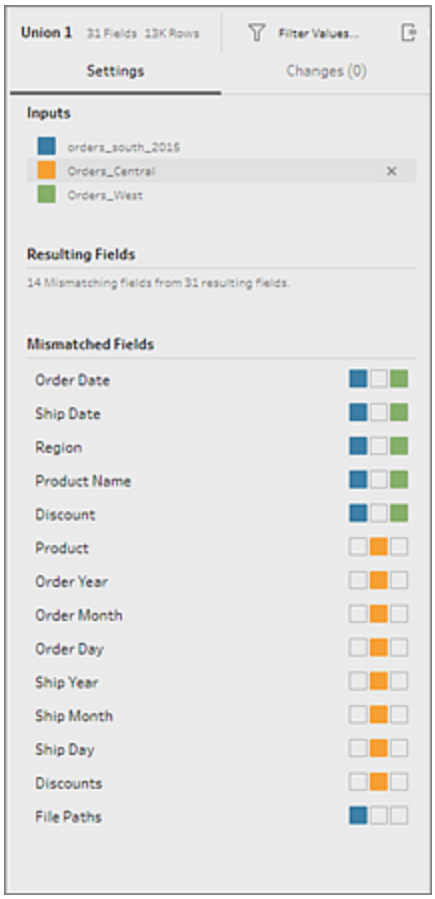


3. In the union profile, review the metadata about the union. You can remove tables from the union as well as see details about any mismatched fields.

Inspect the results of the union

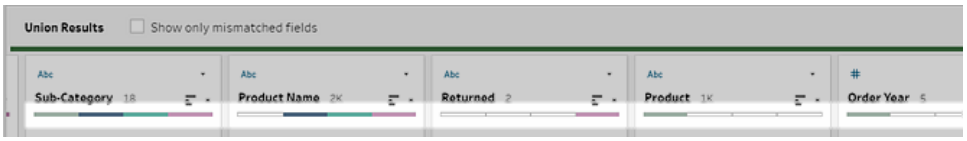
After you create a union, inspect the results of the union to validate that the data in the union is what you expect. To validate your unioned data, check the following areas:

- **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.

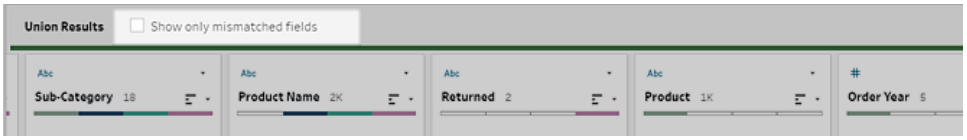


- **Review the colours for each field**: Next to each field listed in the Union summary and above each field in the union profile, is a set of colours. The colours correspond to each table in the union.

  If all table colours show for that field, then the union performed correctly for that field. A missing table colour 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, tick the **Show only mismatched fields** 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** below 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.

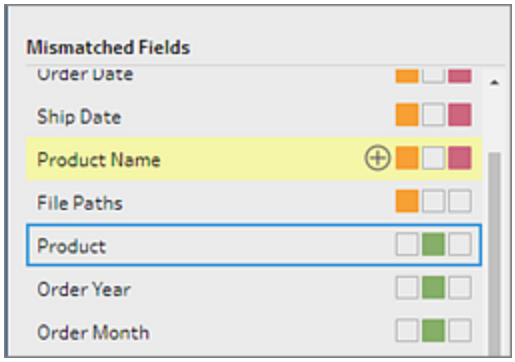
- **Corresponding fields have different names**: If corresponding fields between tables have different names, you can use union recommendations, manually merge fields in the **Mismatched Fields** list, or rename the field in the union profile to merge the mismatched fields together.

  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.

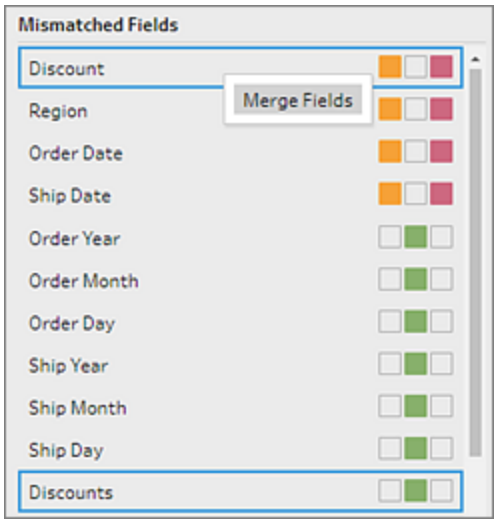
2. Hover on the highlighted field and click the plus button to merge the fields.



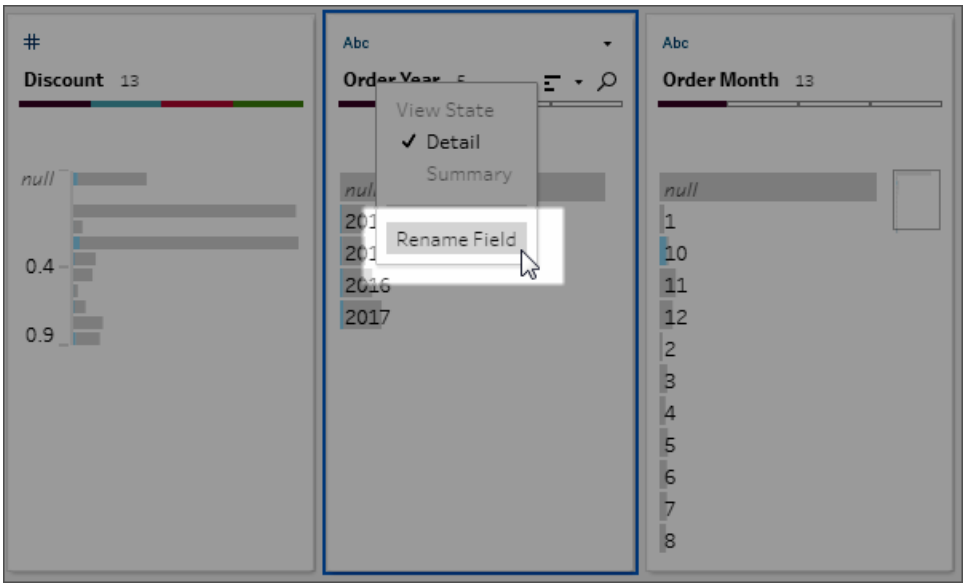
To manually merge fields in the **Mismatched Fields** list, do the following:

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

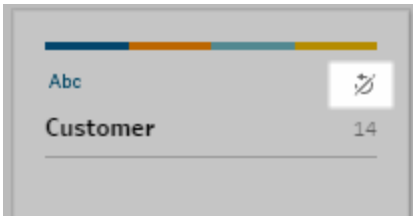
   If you see **No options available** when you right-click on the field, this is because the fields are not eligible to merge. For example trying to merge two fields from the same input.
3. Click **Merge Fields** to merge the selected fields.



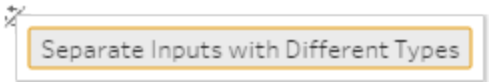
To rename the field in the union profile pane, right-click on the field name and click **Rename Field**.



- **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 changes the data type of one of the fields so that 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 or Ctrl-click (MacOS) 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 facing 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 Output step.

For information about how to merge fields in the same file, see **Merge fields** on page 243.

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.

Add Einstein Discovery Predictions to your flow

Supported in Tableau Prep Builder version 2021.1.3 and later and on the web in Tableau Cloud and Tableau Server version 2021.2.0 and later.

Use Einstein Discovery-powered models to bulk score predictions for the data in your flow. Predictions can help you make better informed decisions and take actions to improve your business outcomes.

When applying these models, a new field for predicted outcomes (in the form of probability scores or estimated averages) is automatically added to your flow. You can also add top predictors and top improvements fields to your flow data by selecting these options when applying your model. Top predictors show factors that contributed most significantly to the prediction. Top improvements show suggested actions to take to improve the predicted outcome.

For example, to predict employee retention, you could build a model using historical data (where you already know the outcome) in Einstein Discovery, then apply that model to the data set in your flow and generate the predicted outcome. Prediction results are applied at the row-level, helping you dive deep into your analysis in Tableau.

If you need to apply multiple models to your data set, you can include multiple prediction steps in your flow. Each prediction step applies a single prediction model to the flow. Starting in version 2021.2, you can sign into multiple Einstein Discovery servers in a single flow to choose the models you need. Prior versions restrict you to a single Einstein Discovery server per flow.

> **Note**: You must have a Salesforce licence and user account that is configured to access Einstein Discovery to use this feature. See **Prerequisites** on the facing page for more information.

What is Einstein Discovery?

Einstein Discovery augments your business intelligence with statistical modelling and supervised machine learning to identify, surface and visualise insights into your business data. It

quickly sifts through millions of rows of data to find important correlations, predict outcomes and suggest ways to improve those predicted outcomes.

For more information about Einstein Discovery, see Getting Started with Discovery, and Explain, Predict and Take Action with Einstein Discovery in Salesforce help. You can also expand your knowledge with the Gain Insight with Einstein Discovery trail in Trailhead.

> **Note**: Einstein Discovery in Tableau is powered by salesforce.com. Consult your agreement with salesforce.com for applicable terms.

Prerequisites

To configure and use Einstein Discovery predictions in your flow, you need certain licences, access and permissions in Salesforce and Tableau.

Salesforce Requirements

requirement	description
Salesforce licence	One of the following licences: • Einstein Discovery in Tableau licence • Tableau CRM Plus licence • Einstein Predictions licence These licences are available for an extra cost.
Salesforce user account	Account that is configured to access Einstein Discovery. If you use the Einstein Discovery in Tableau licence, your user account must have the **View Einstein Discovery Recommendations Via Connect API** system permission assigned to it. If you use either the Tableau CRM Plus licence or Einstein Predictions licence: • To get predictions using already deployed Einstein Discovery models, the account must have the **View Einstein Discovery Recommendations** system permission assigned to it.

requirement	description
	• To build, deploy and manage predictions in Einstein Discovery, the account must have the **Manage Einstein Discovery permission** assigned to it. To configure user accounts, see Set Up Einstein Discovery in Salesforce help.
Administrator settings	Salesforce administrators will need to: • **Tableau Prep Extensions**: Configure Salesforce to create a connected app for Tableau Server (basic). Required for Tableau Server only.

Tableau Prep Requirements

requirement	description
Tableau Prep licence and permissions	Creator licence. As a creator you need to be able to sign into the Salesforce org account to access prediction definitions and add models to your flow.
Tableau user account	In Tableau Server and Tableau Cloud version 2021.2 and later, users can save Salesforce user account credentials along with their Tableau user account. For more information about connecting to Salesforce data see **Connect to Salesforce data** on page 77.
Administrator settings	Tableau Server administrators will need to configure Tableau Server to integrate with Einstein Discovery for Tableau Prep. For more information, see Configure Einstein Discovery Integration in the Tableau Server help.

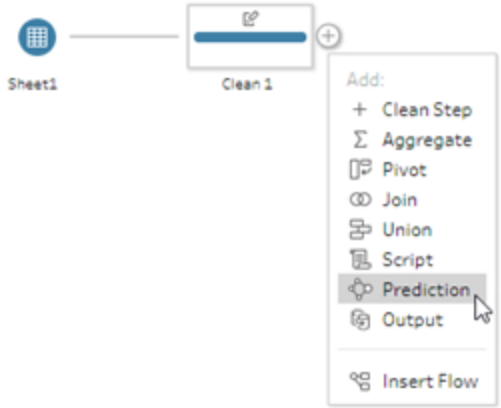
Add prediction data to your flow

> **Note**: In version 2021.1.4 and earlier, flows that include prediction steps can only be run manually in Tableau Prep Builder.

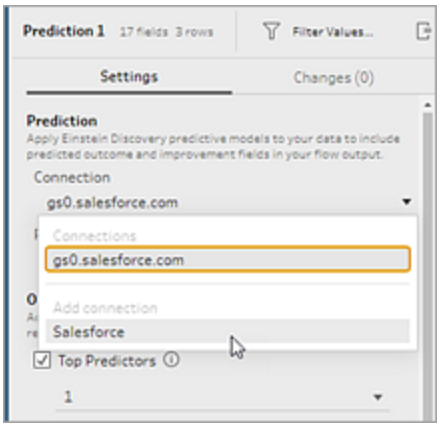
To apply Einstein Discovery predictions to your flow, you will need:

- Access to a Salesforce org.
- Access to Tableau Prep Builder version 2021.1.3 or later.
- If authoring or running flows on the web, access to Tableau Cloud or Tableau Server version 2021.2 or later has been enabled for Einstein Discovery predictions.
- Einstein Discovery prediction models that are deployed in Salesforce.
- Source data in Tableau Prep with fields that match the model fields required by the Einstein Discovery prediction model.

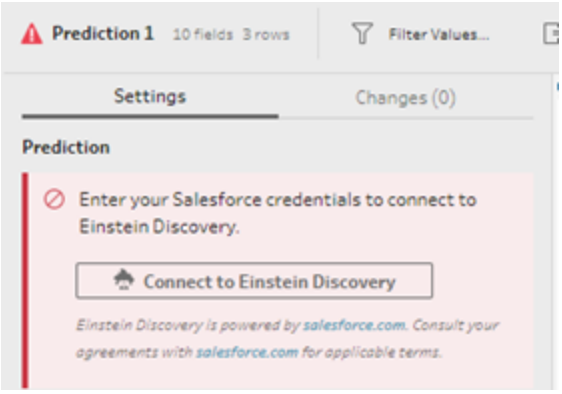
1. Open Tableau Prep and connect to a data source.
2. Apply any cleaning operations as needed.
3. Click the plus icon and select **Prediction** from the **Add** menu.



4. In the **Prediction** pane on the **Settings** tab, do one of the following, depending on your version:
   - **Version 2021.2 and later**: In the **Connection** drop-down, connect to your Salesforce server or select your Salesforce server from the list if you already have a connection established.

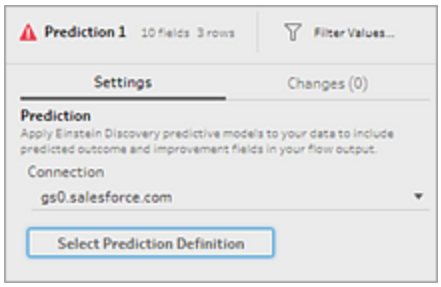


- **Version 2021.1.4 and earlier**: Click **Connect to Einstein Discovery**.

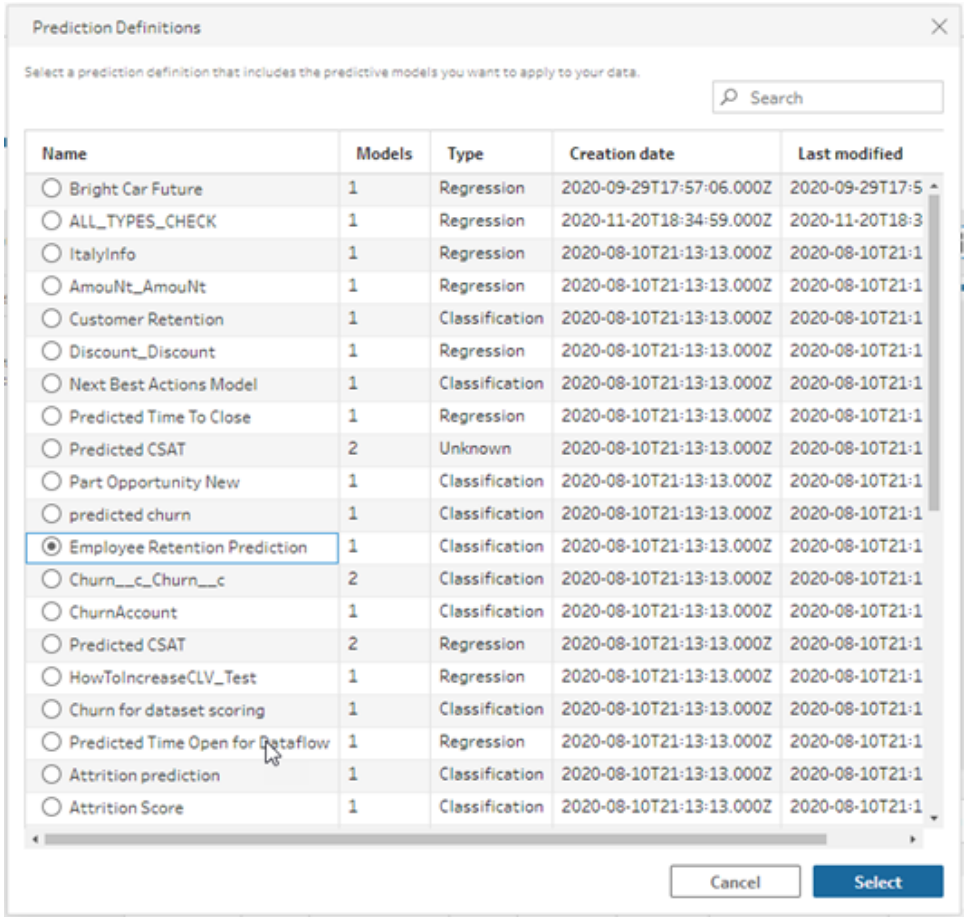


When connecting for the first time, a web page opens, asking you to sign in to your Salesforce account using your Salesforce credentials. After you sign in, a web page opens asking if you want to let Tableau access your Salesforce data. Click **Allow** to continue, and then close the resulting tab in your browser.

5. Click **Select Prediction Definition**. This opens the list of deployed models that you have access to. The models are built and deployed in Salesforce using Einstein Discovery. For more information about predictive models see, About Models in Salesforce help.



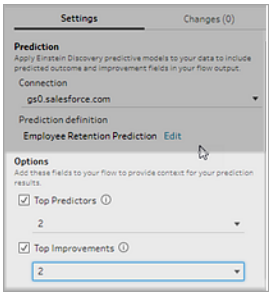
6. In the **Prediction Definitions** dialog, select the prediction definition that maps to your data set. To generate predicted outcomes using your flow data, all fields in the model must map to a corresponding flow field.



Name	Models	Type	Creation date	Last modified
Bright Car Future	1	Regression	2020-09-29T17:57:06.000Z	2020-09-29T17:5
ALL_TYPES_CHECK	1	Regression	2020-11-20T18:34:59.000Z	2020-11-20T18:3
ItalyInfo	1	Regression	2020-08-10T21:13:13.000Z	2020-08-10T21:1
AmouNt_AmouNt	1	Regression	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Customer Retention	1	Classification	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Discount_Discount	1	Regression	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Next Best Actions Model	1	Classification	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Predicted Time To Close	1	Regression	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Predicted CSAT	2	Unknown	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Part Opportunity New	1	Classification	2020-08-10T21:13:13.000Z	2020-08-10T21:1
predicted churn	1	Classification	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Employee Retention Prediction	1	Classification	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Churn__c_Churn__c	2	Classification	2020-08-10T21:13:13.000Z	2020-08-10T21:1
ChurnAccount	1	Classification	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Predicted CSAT	2	Regression	2020-08-10T21:13:13.000Z	2020-08-10T21:1
HowToIncreaseCLV_Test	1	Regression	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Churn for dataset scoring	1	Classification	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Predicted Time Open for Dataflow	1	Regression	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Attrition prediction	1	Classification	2020-08-10T21:13:13.000Z	2020-08-10T21:1
Attrition Score	1	Classification	2020-08-10T21:13:13.000Z	2020-08-10T21:1

7. In the **Options** section, select up to 3 top predictors and improvements to include in your flow data. This is supplemental data you can add to your flow.

- Top predictors indicate which factors contributed the most to the predicted outcome.
- Top improvements suggest actions to take to improve the predicted outcome.



8. In the **Map Fields** section, map your flow fields to your model fields.
   - All model fields must be mapped to a corresponding flow field.
   - Field names that match exactly are automatically mapped.
   - You can't map the same flow field to multiple model fields.
   - Model and flow field data types must match.

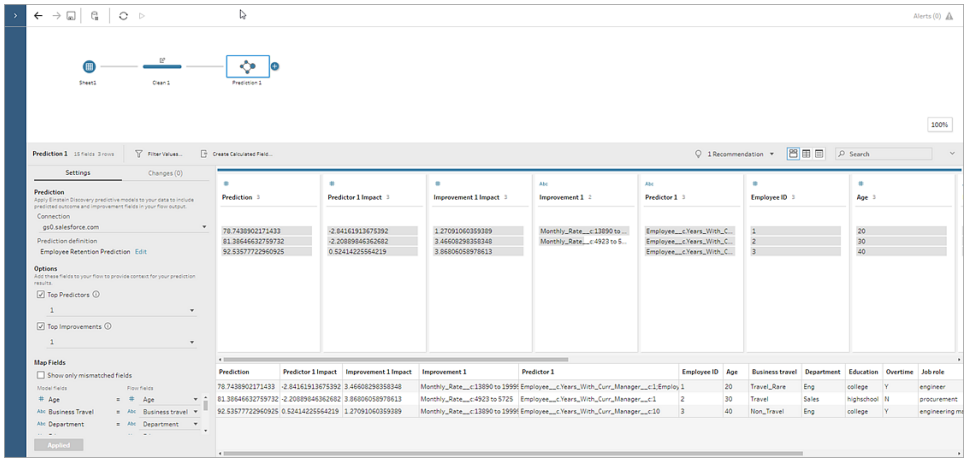
     If your flow field is assigned to a different data type, you'll need to change it to match the data type assigned to the model field.

     To change the data type, in the **Map Fields** section, simply click the data type for the flow field, then select the new data type in the menu. You can then change the data type back in a subsequent cleaning step.

     For more information about changing data types, see **Review the data types assigned to your data** on page 163.

9. To apply your settings and run the model against your data, click **Apply**. The prediction results show in the profile pane and data grid.

   If you change any settings, you can click **Apply** again to re-run the model with your changes. If you leave the **Prediction** step before clicking **Apply**, the model won't run and your changes will be lost.

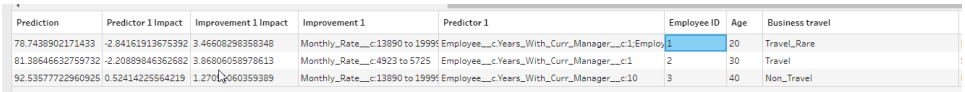


Reviewing your results

After you apply the predictive model to your flow data you can generate your flow output and use the new data source to analyse the predicted outcomes at the row level in Tableau. To understand the results of the prediction model, let's look at an example.

In this topic, we applied the Employee Retention Prediction model to our employee data in Tableau Prep to get a probability score that an employee will stay with the company.

This gave us the following results:



Prediction	Predictor 1 Impact	Improvement 1 Impact	Improvement 1	Predictor 1	Employee ID	Age	Business travel
78.7438902171433	-2.84161913675392	3.46608298358348	Monthly_Rate__c:13890 to 1999	Employee__c:Years_With_Curr_Manager__c:1;Employ	1	20	Travel_Rare
81.38646632759732	-2.20889846362682	3.86806058978613	Monthly_Rate__c:4923 to 5725	Employee__c:Years_With_Curr_Manager__c:1	2	30	Travel
92.53577722960925	0.52414225564219	1.27091060359389	Monthly_Rate__c:13890 to 1999	Employee__c:Years_With_Curr_Manager__c:10	3	40	Non_Travel

Let's look at what these results tell us for Employee 2:

Question	Prediction	Where is this?
How likely will this employee stay?	Einstein Discovery predicts that there is an 81.38% chance that they will stay.	Prediction field

What factors impact this result?	The years with the current manager reduces the chance that this employee will stay by 2.2%.	Predictor 1 field (top predictor) Predictor 1 Impact (percent impact of the top predictor)
What can improve this predicted outcome?	Increasing the employee's monthly rate between 4923 to 5725 increases the likelihood that the employee stays by 3.86%.	Improvement 1 field (top improvement) Improvement 1 impact (percent impact of making the suggested change)

Save and Share Your Work

Note: Starting in version 2020.4, you can also create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted.

At any point in your flow you can manually save your work, or let Tableau automatically do it for you when creating or editing flows on the web. When working with flows on the web, there are a few differences.

For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

Tableau Prep Builder	Tableau Prep on the web
• View a preview of the data in your flow in Tableau Desktop. • Include direct file connections in your flow input, or package your files and publish the packaged flow to your server. • Output your flow to a file, published data source or to a database (version 2020.3.1 and later).	• Create and edit flows on the web. • Upload files for your flow inputs and connect to various data sources. • Output your flow to a published data source or to a database.

To keep data fresh, you can manually run your flows from Tableau Prep Builder or from the command line. You can also run flows published on Tableau Server or Tableau Cloud manually or on a schedule. For more information about running flows, see **Publish a Flow to Tableau Server or Tableau Cloud** on page 463.

Save a flow

In Tableau Prep Builder, you can manually 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.

In web authoring, local files are automatically packaged with our flow. Direct file connections aren't yet supported.

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

- To manually save your flow, from the top menu, select **File** > **Save**.
- In Tableau Prep Builder, 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.

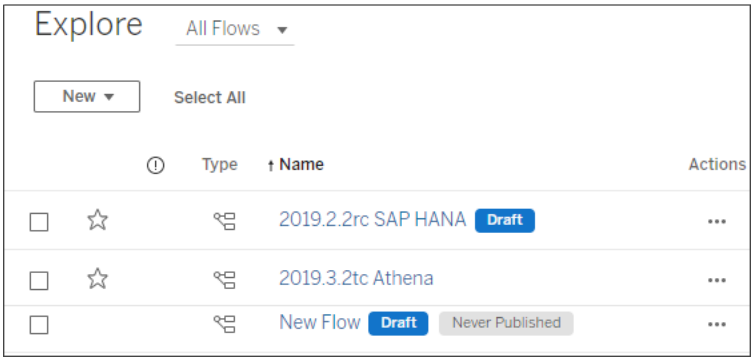
Automatically save your flows on the web

Supported in Tableau Server version 2020.4 and later.

If you create or edit flows on the web, when you make a change to the flow (connect to a data source, add a step and so on) your work is automatically saved every few seconds as a draft so you won't lose your work.

You can only save flows to the server you're currently signed into. You can't create a draft flow on one server and try to save or publish it to another server. If you want to publish the flow to a different project on the server, use the **File** > **Publish As** menu option, then select your project from the dialog.

Draft flows can only be seen by you until you publish them and make them available to anyone who has permissions to access the project on your server. Flows in a draft status are tagged with a **Draft** badge, so you can easily find your flows that are in progress. If the flow has never been published, a **Never Published** badge is shown next to the Draft badge.



After a flow is published and you edit and republish the flow, a new version is created. You can see a list of flow versions in the **Revision History** dialog. From the **Explore** page, click the ··· **Actions** menu and select **Revision History**.

For more information about managing revision history, see Work with Content Revisions in the Tableau Desktop help.

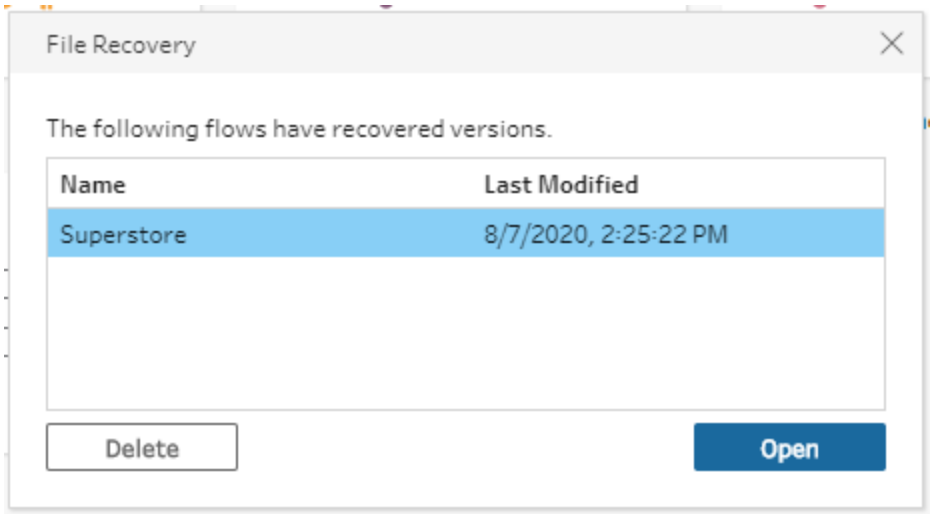
> **Note**: Autosave is enabled by default. It's possible, but not recommended, for administrators to disable autosave on a site. To turn off autosave, use the Tableau Server REST API method "Update Site" and set the `flowAutoSaveEnabled` attribute to false. For more information, see Tableau Server REST API Site Methods: Update Site.

Automatic file recovery

Supported in Tableau Prep Builder version 2020.3.3 and later.

By default, Tableau Prep Builder automatically saves a draft of any open flows if the application freezes or crashes. Draft flows are saved in your **Recovered Flows** folder in your **My Tableau Prep repository**. The next time you open the application, a dialog is shown with a list of the recovered flows to select from. You can open a recovered flow and continue where you left off, or delete the recovered flow file if you don't need it.

> **Note**: If you have recovered flows in your **Recovered Flows** folder, this dialog shows every time you open the application until that folder is empty.



If you don't want this feature enabled, as an Administrator, you can turn it off during install or after install. For more information about how to turn off this feature, see Turn off file recovery in the Tableau Desktop and Tableau Prep Deployment Guide.

View flow output in Tableau Desktop

> **Note**: This option isn't available on the web.

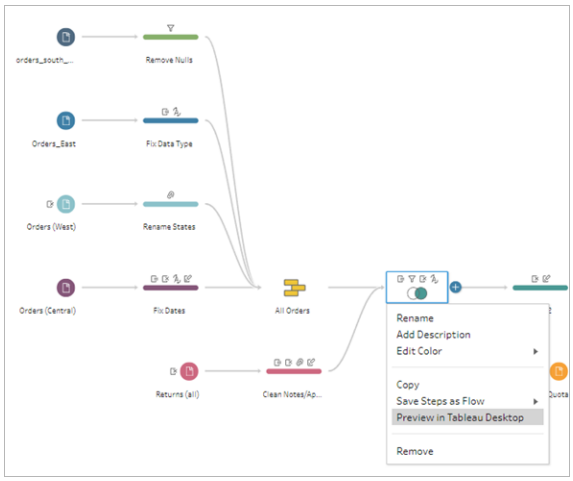
Sometimes when you're cleaning your data you might want to check your progress by looking at it in Tableau Desktop. When your flow opens in Tableau Desktop, Tableau Prep Builder creates a permanent Tableau .hyper file and a Tableau data source (.tds) file. These files are saved in your Tableau repository in the **Data sources** 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're working with in your flow with the operations applied to it, up to the step that you selected.

> **Note**: While you can experiment with your data, Tableau only shows you a sample of your data and you won't be able to save the workbook as a packaged workbook (.twbx). When you're ready to work with your data in Tableau, create an output step in your flow and save the output to a file or as a published data source, then connect to the full data source in Tableau.

To view your data sample in Tableau Desktop do the following:

1. Right-click on 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 data extract files and published data sources

Important: Starting in Tableau Prep Builder version 2020.3.1, Tableau Data Extract (.tde) files are no longer supported as a flow output type. To avoid flow run failures, convert flow outputs from (.tde) files to Hyper Extract (.hyper) files. Flows published to Tableau Server or Tableau Cloud must be downloaded to Tableau Prep Builder to change the file output type.

To create your flow output, 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 (.hyper) file.

> **Note:** You can publish data extracts or published data sources to Tableau Server version 10.0 and later as well as to Tableau Cloud.

Tableau Prep Builder

You can create an extract file from your flow output to use in Tableau Desktop or to share your data with third parties. Create an extract file in the following formats:

- **Hyper Extract (.hyper)**: This is the latest Tableau extract file type and 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. The encoding of the exported CSV file will be UTF-8 with BOM.
- **Microsoft Excel (.xlsx)**: Starting in version 2021.1.2, you can output your flow data to a Microsoft Excel spreadsheet. Legacy Microsoft Excel .xls file types aren't supported.

Tableau Prep Builder and on the web

Publish your flow output as a published data source or output to a database.

- Save your flow output as a data source to Tableau Server or Tableau Cloud to share your data and provide centralised access to the data you've cleaned, shaped and combined.
- Save your flow output to a database to create, replace or append the table data with your clean, prepared flow data. For more information, see **Save flow output data to external databases** on page 394.

Use incremental refresh when running your flow to save time and resources by refreshing only new data instead of your full data set. For information about how to configure and run your flow using incremental refresh, see **Refresh Flow Data Using Incremental Refresh** on page 416.

> **Note**: To publish Tableau Prep Builder 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 Builder. For more information, see the Before you Install in the Tableau Desktop and Tableau Prep Builder Deployment Guide.

Include parameters in your flow output

Supported in Tableau Prep Builder and on the web starting in version 2021.4

Include parameter values in your flow output filenames, paths, table names or custom SQL scripts (version 2022.1.1 and later) to easily run your flows for different data sets. For more information, see **Create and Use Parameters in Flows** on page 205.

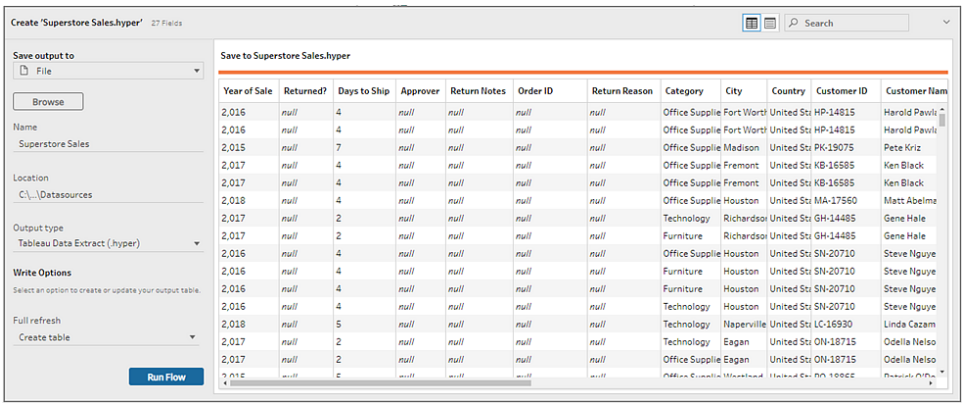
Create an extract to a file

> **Note**: This output option is not available when creating or editing flows on the web.

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 runs the flow and updates your output.

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



2. In the left pane select **File** from the **Save output to** drop-down list. In prior versions, 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 from the following output types:
   - Tableau Data Extract (.hyper)
   - Comma Separated Values (.csv)
5. (Tableau Prep Builder version 2020.2.1 and later) In the **Write Options** section, view the default write option to write the new data to your files and make any changes as needed. For more information, see **Configure write options** on page 421.
   - **Create table:** This option creates a new table or replaces the existing table with the new output.
   - **Append to table**: This option adds the new data to your existing table. If the table doesn't already exist, a new table is created and subsequent runs add new rows to this table.

> **Note**: **Append to table** isn't supported for .csv output types. For more information about supported refresh combinations, see **Flow refresh options** on page 417.

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

Create an extract to a Microsoft Excel Worksheet

Supported in Tableau Prep Builder version 2021.1.2 and later. This output option is not available when creating or editing flows on the web.

When you output flow data to a Microsoft Excel worksheet you can create a new worksheet or append or replace data in an existing worksheet. The following conditions apply:

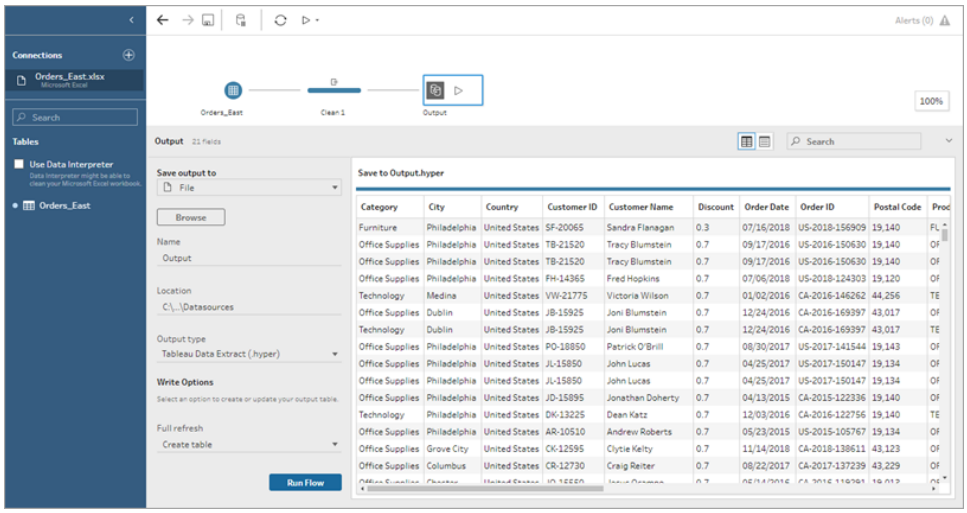
- Only Microsoft Excel .xlsx file formats are supported.
- The worksheet rows begin at cell A1.
- When appending or replacing data, the first row is assumed to be headers.
- Header names are added when creating a new worksheet, but not when adding data to an existing worksheet.
- Any formatting or formulas in existing worksheets aren't applied to the flow output.
- Writing to named tables or ranges is not currently supported.
- Incremental refresh is not currently supported.

Output flow data to a Microsoft Excel worksheet file

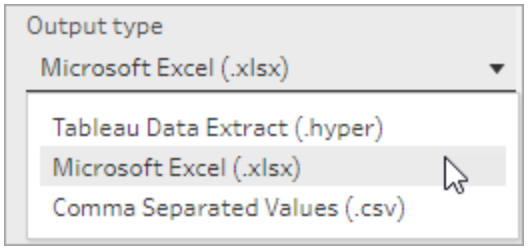
1. Click the plus icon ⊕ on a step and select **Add Output**.

   If you have run the flow before, click the run flow ▷ button on the Output step. This runs the flow and updates your output.

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

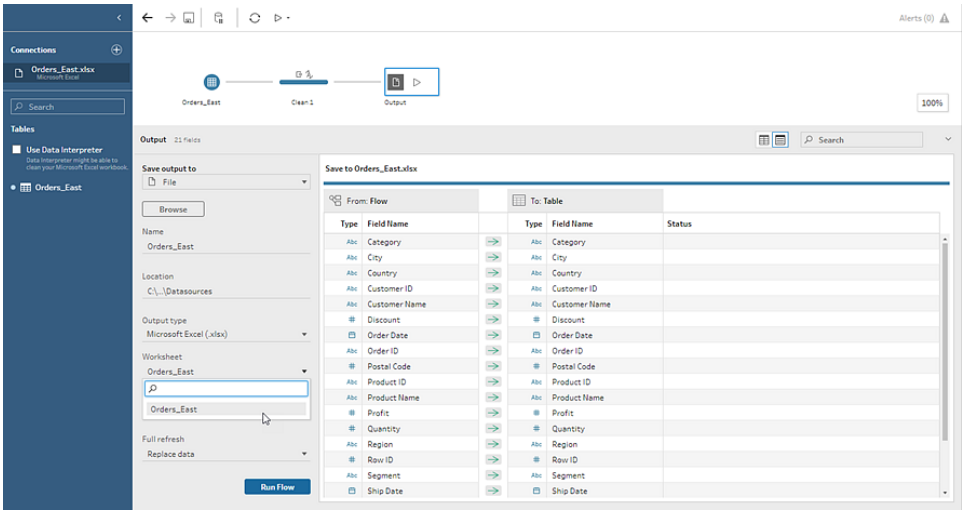


2. In the left pane select **File** from the **Save output to** drop-down list.
3. Click the **Browse** button, then in the **Save Extract As** dialog, enter or select the file name and click **Accept**.
4. In the **Output type** field, select Microsoft Excel (.xlsx).



5. In the **Worksheet** field, select the worksheet you want to write your results to, or enter a new name in the field instead, then click on **Create new table**.
6. In the **Write Options** section, select one of the following write options:
   - **Create table:** Creates or re-creates (if the file already exists) the worksheet with your flow data.
   - **Append to table**: Adds new rows to an existing worksheet. If the worksheet doesn't exist, one is created and subsequent flow runs add rows to that worksheet.
   - **Replace data**: Replaces all of the existing data except the first row in an existing worksheet with the flow data.

A field comparison shows you the fields in your flow that match the fields in your worksheet, if it already exists. If the worksheet is new, then a one-to-one field match is shown. Any fields that don't match are ignored.



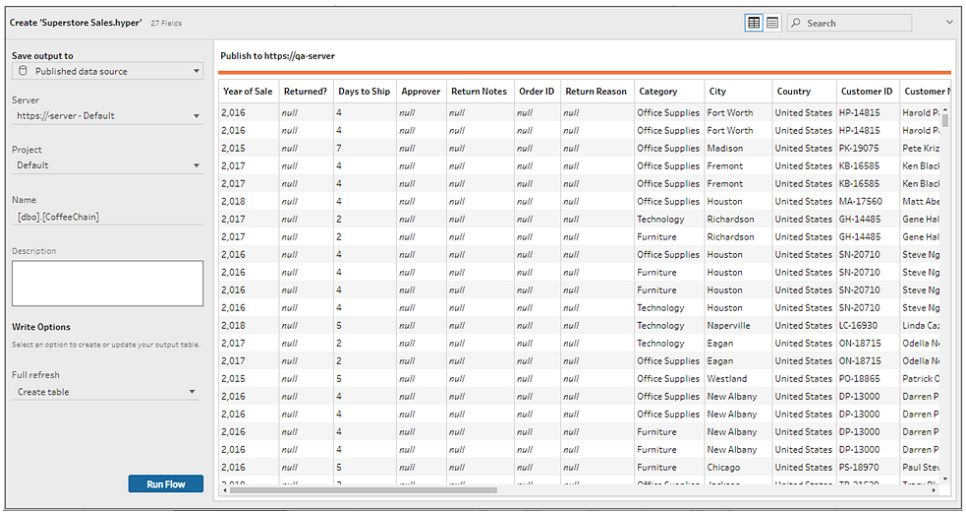
7. Click **Run Flow** to run the flow and generate the Microsoft Excel extract file.

Create a published data source

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

> **Note**: Tableau Prep Builder refreshes previously published data sources and maintains any data modelling (for example calculated fields, number formatting, and so on) that might be included in the data source. If the data source can't be refreshed, the data source, including data modelling, will be replaced instead.

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



3. From the **Save output to** drop-down list, select **Published data source** (**Publish as data source** in previous versions) . Complete the following fields:

   - **Server** (Tableau Prep Builder only): Select the 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.

     > **Note**: Starting in Tableau Prep Builder version 2020.1.4, after you sign in to your server, Tableau Prep Builder remembers your server name and credentials when you close the application. The next time you open the application, you are already signed into your server.

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

     If you are outputting to Tableau Cloud, include the pod your site is hosted on in the "serverUrl". For example, "https://eu-west-1a.online.tableau.com" not "https://online.tableau.com".

   - **Project**: Select the project where you want to load the data source and extract.
   - **Name**: Enter a file name.
   - **Description**: Enter a description for the data source.

4. (Tableau Prep Builder version 2020.2.1 and later) In the **Write Options** section, view the default write option to write the new data to your files and make any changes as needed.

For more information, see **Configure write options** on page 421

- **Create table:** This option creates a new table or replaces the existing table with the new output.
- **Append to table**: This option adds the new data to your existing table. If the table doesn't already exist, a new table is created and subsequent runs will add new rows to this table.

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

Save flow output data to external databases

Supported in Tableau Prep Builder version 2020.3.1 and later and on Tableau Server and Tableau Cloud starting in version 2020.4.

Important: This feature enables you to permanently delete and replace data in an external database. Be sure that you have permissions to write to the database.
To prevent data loss, you can use the **Custom SQL** option to make a copy of your table data and run it before writing the flow data to the table.

You can connect to data from any of the connectors that Tableau Prep Builder or the web supports and output data to an external database. This enables you to add or update data in your database with clean, prepped data from your flow each time the flow is run. This feature is available for both incremental and full refresh options. For more information about how to configure incremental refresh, see **Refresh Flow Data Using Incremental Refresh** on page 416.

When you save your flow output to an external database, Tableau Prep does the following:

1. Generates the rows and runs any SQL commands against the database.
2. Writes the data to a temporary table (or staging area if outputting to Snowflake) in the output database.
3. If the operation is successful, the data is moved from the temporary table (or your staging area for Snowflake) into the destination table.
4. Runs any SQL commands that you want to run after writing the data to the database.

If the SQL script fails, the flow will fail. However your data will still be loaded to your database tables. You can try running the flow again or manually run your SQL script on your database to apply it.

Output options

You can select the following options when writing data to a database. If the table doesn't already exist, it's created when the flow is first run.

- **Append to table**: This option adds data to an existing table. If the table doesn't exist, the table is created when the flow is first run and data is added to that table with each subsequent flow run.
- **Create table**: This option creates a new table with the data from your flow. If the table already exists, the table and any existing data structure or properties defined for the table is deleted and replaced with a new table that uses the flow data structure. Any fields that exist in the flow are added to the new database table.
- **Replace data**: This option deletes the data in your existing table and replaces it with the data in your flow, but preserves the structure and properties of the database table. If the table doesn't exist, the table is created when the flow is first run and the table data is replaced with each subsequent flow run.

Additional options

In addition to the write options, you can also include custom SQL scripts or add a new tables to your database.

- **Custom SQL scripts**: Enter your custom SQL and select whether to run your script before, after or both before and after data is written to the database tables. You can use these scripts to create a copy of your database table before the flow data is written to the table, add an index, add other table properties and so on.

  > **Note**: Starting in version 2022.1.1, you can also insert parameters in your SQL scripts. For more information, see **Apply user parameters to output steps** on page 215.

- **Add a new table**: Add a new table with a unique name to the database instead of selecting one from the existing table list. If you want to apply a schema other than the default schema (Microsoft SQL Server and PostgreSQL), you can specify it using the syntax `[schema name].[table name]`.

Supported databases and database requirements

Tableau Prep supports writing flow data to tables in a select number of databases. Flows that run on a schedule in Tableau Cloud can only write to these databases if they are cloud-hosted.

Private network connections and on-premise connections through Tableau Bridge are not supported. You can create a published data source in Tableau Cloud that uses Bridge to refresh the data, then Prep can connect to the published data source.

Some databases have data restrictions or requirements. Tableau Prep may also impose some limits to maintain peak performance when writing data to the supported databases. The following table lists the databases where you can save your flow data and any database restrictions or requirements. Data that doesn't meet these requirements can result in errors when running the flow.

Note Setting character limits for your fields isn't yet supported. However, you can create the tables in your database that include character limit constraints, then use the **Replace data** option to replace your data but maintain the table's structure in your database.

Database	Requirements or restrictions
Amazon Redshift	• Collation sequences aren't supported. See the Amazon Redshift documentation for more information. • Field names are converted to all lowercase. • Up to 8192 characters can be written for text field values. Longer values are truncated.
Google BigQuery	• Tableau can write up to 2 GB as output to the table.
Microsoft SQL Server	• Up to 3072 characters can be written for text field values. Longer values are truncated. • (Version: 2022.3.1) Flow outputs published to Tableau Server are allowed write access to a Microsoft SQL Server database using Run As credentials. See `maestro.output.write_to_mssql_using_runas` in tsm configuration set Options.
MySQL	• Up to 8192 characters can be written for text field values. Longer

	values are truncated.
Oracle	• Field and table names can't exceed 30 characters. • Up to 1000 characters can be written for text field values. Longer values are truncated. • Special characters in field names may cause errors.
Pivotal Greenplum Database	• Up to 8192 characters can be written for text field values. Longer values are truncated.
PostgreSQL	• Up to 8192 characters can be written for text field values. Longer values are truncated.
SAP HANA	• Up to 8192 characters can be written for text field values. Longer values are truncated.
Snowflake	• Up to 8192 characters can be written for text field values. Longer values are truncated. • Warehouse options must be set to auto-resume to enable Tableau Prep to write data to the database warehouse. For more information, see Auto-suspension and Auto-resumption in the Snowflake documentation.
Teradata	• Up to 1000 characters can be written for text field values. Longer values are truncated.
Vertica	• Up to 8192 characters can be written for text field values. Longer values are truncated.

Save flow data to a database

Note: Writing flow output to a database using Windows Authentication isn't supported. If you use this method of authentication, you'll need to change the connection authentication to use the username and password.

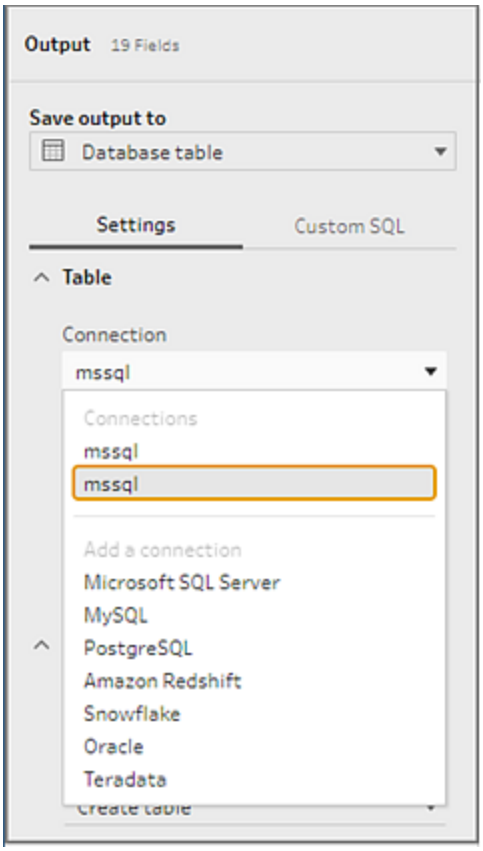
You can embed your credentials for the database when publishing the flow. For more

information about embedding credentials, see the **Databases** section in **Publish a flow from Tableau Prep Builder** on page 467

1. Click the plus icon ⊕ on a step and select **Add Output**.
2. From the **Save output to** drop-down list, select **Database table**.
3. In the **Settings** tab, enter the following information:

   - In the **Connection** drop down list , select the database connector where you want to write your flow output. Only supported connectors are shown. This can be the same connector that you used for your flow input or a different connector. If you select a different connector, you'll be prompted to sign in.

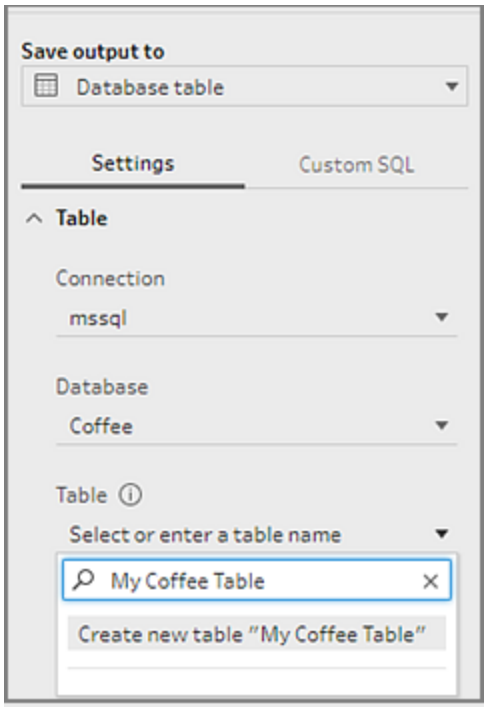
     Important: Make sure you have write permission to the database you select. Otherwise the flow might only partially process the data.



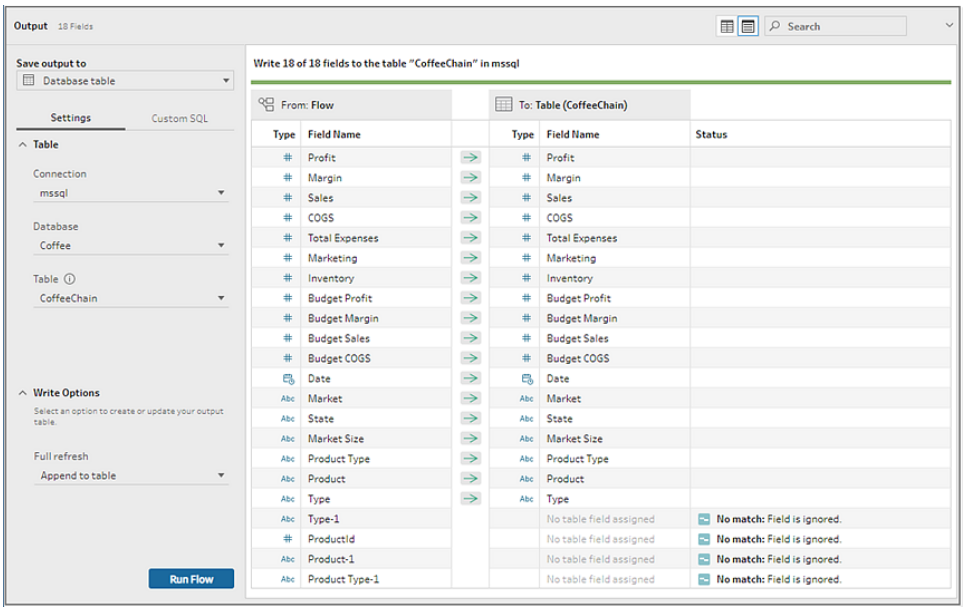
   - In the **Database** drop-down list, select the database where you want to save your flow output data.

- In the **Table** drop-down list, select the table where you want to save your flow output data. Depending on the **Write Option** you select, a new table will be created, the flow data will replace any existing data in the table or flow data will be added to the existing table.

  To create a new table in the database, enter a unique table name in the field instead, then click on **Create new table**. When you run the flow for the first time, no matter which write option you select, the table is created in the database using the same schema as the flow.



4. The output pane shows you a snapshot of your data. A field comparison shows you the fields in your flow that match the fields in your table, if the table already exists. If the table is new, then a one-to-one field match is shown.



If there are any field mismatches, a status note shows you any errors.

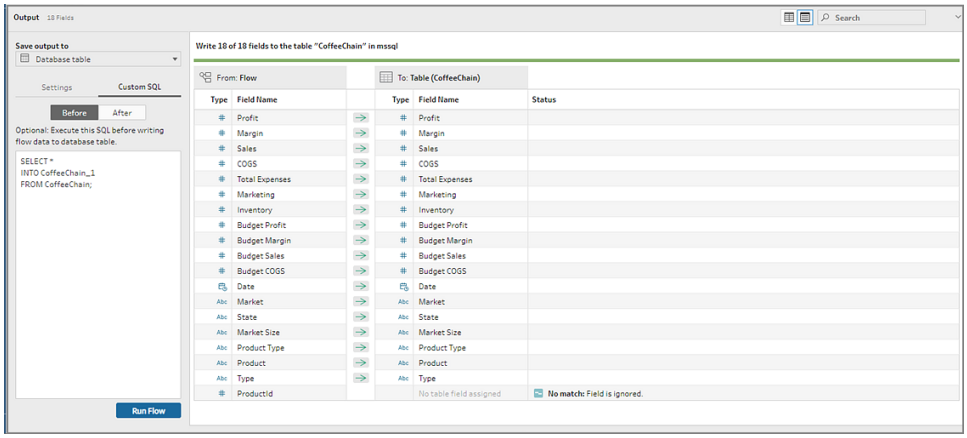
- **No match: Field is ignored**: Fields exist in the flow but not in the database. The field won't be added to the database table unless you select the **Create table** write option and perform a full refresh. Then the flow fields are added to the database table and use the flow output schema.
- **No match: Field will contain Null values**: Fields exist in the database but not in the flow. The flow passes a Null value to the database table for the field. If the field does exist in the flow, but is mismatched because the field name is different, you can navigate to a cleaning step and edit the field name to match the database field name. For information about how to edit field name, see **Apply cleaning operations** on page 236.
- **Error: Field data types do not match**: The data type assigned to a field in both the flow and the database table you are writing your output to must match, otherwise the flow will fail. You can navigate to a cleaning step and edit the field data type to fix this. For more information about changing data types, see **Review the data types assigned to your data** on page 163.

5. Select a write option. You can select a different option for full and incremental refresh and the option is applied when you select your flow run method. For more information about running our flow using incremental refresh, see **Refresh Flow Data Using Incremental Refresh** on page 416.
    - **Append to table**: This option adds data to an existing table. If the table doesn't exist, the table is created when the flow is first run and data is added to that table

with each subsequent flow run.

- **Create table**: This option creates a new table. If the table with the same name already exists, the existing table is deleted and replaced with the new table. Any existing data structure or properties defined for the table are also deleted and replaced with the flow data structure. Any fields that exist in the flow are added to the new database table.
- **Replace data**: This option deletes the data in your existing table and replaces it with the data in your flow, but preserves the structure and properties of the database table.

6. (optional) Click on the **Custom SQL** tab and enter your SQL script. You can enter a script to run **Before** and **After** the data is written to the table.



7. Click **Run Flow** to run the flow and write your data to your selected database.

Save flow output data to Datasets in CRM Analytics

Supported in Tableau Prep Builder and on the web starting in version 2022.3.

> **Note**: CRM Analytics has several requirements and some limitations when integrating data from external sources. To make sure that you can successfully write your flow output to CRM Analytics, see Considerations before integrating data into datasets in the Salesforce help.

Clean your data using Tableau Prep and get better prediction results in CRM Analytics. Simply connect to data from any of the connectors that Tableau Prep Builder or Tableau Prep on the web supports. Then, apply transformations to clean your data and output your flow data directly to Datasets in CRM Analytics that you have access to.

Flows that output data to CRM Analytics can't be run using the command line interface. You can run flows manually using Tableau Prep Builder or using a schedule on the web with Tableau Prep Conductor.

Prerequisites

To output flow data to CRM Analytics, check that you have the following licences, access and permissions in Salesforce and Tableau.

Salesforce Requirements

requirement	description
Salesforce Permissions	You must be assigned to either the **CRM Analytics Plus** or **CRM Analytics Growth** licence. The **CRM Analytics Plus** licence includes the permission sets: • **CRM Analytics Plus Admin**: Enables all permissions required to administer the CRM Analytics platform and Einstein Discovery, including permissions to create and manage CRM Analytics templated apps and Apps. • **CRM Analytics Plus User**: Enables all permissions required to use the CRM Analytics platform, Einstein Discovery and CRM Analytics templated apps and Apps. The **CRM Analytics Growth** licence includes the permission sets: • **CRM Analytics Growth Admin:** Enables all permissions required to administer the CRM Analytics platform, including permissions to create and manage CRM Analytics templated apps and Apps. • **CRM Analytics Growth User:** Enables all permissions required to use the CRM Analytics platform and CRM

requirement	description
	Analytics templated apps and Apps. For more information, see Learn About CRM Analytics Licences and Permissions Sets and Select and assign user permissions sets in the Salesforce help.
Administrator settings	Salesforce administrators will need to configure: • **Tableau Prep Extensions**: Configure Salesforce to create a connected app for Tableau Server (basic). Required for Tableau Server only.

Tableau Prep Requirements

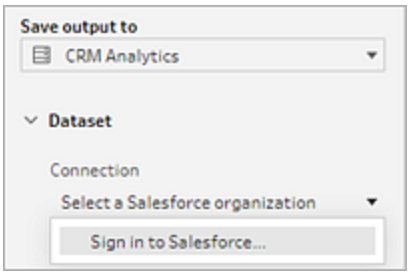
requirement	description
Tableau Prep licence and permissions	Creator licence. As a creator you need to sign in to your Salesforce.org account and authenticate before you can select Apps and Datasets to output your flow data.
OAuth Data Connections	As a Server Administrator, configure Tableau Server with an OAuth client ID and secret on the connector. This is required to run flows on Tableau Server. For more information, see Configure Tableau Server for Salesforce.com Oauth in the Tableau Server help.

Save flow data to CRM Analytics

The following CRM Analytics input limits apply when saving from Tableau Prep Builder to CRM Analytics.

- Maximum file size for external data uploads: 40 GB
- Maximum file size for all external data uploads in a rolling 24-hour period: 50 GB

1. Click the plus icon ⊕ on a step and select **Add Output**.
2. From the **Save output to** drop-down list, select **CRM Analytics**.

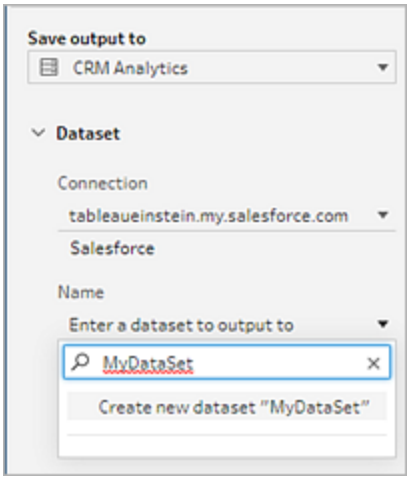


3. In the **Dataset** section, connect to Salesforce.

   Sign in to Salesforce and click **Allow** to give Tableau access to CRM Analytics Apps and datasets or select an existing Salesforce connection

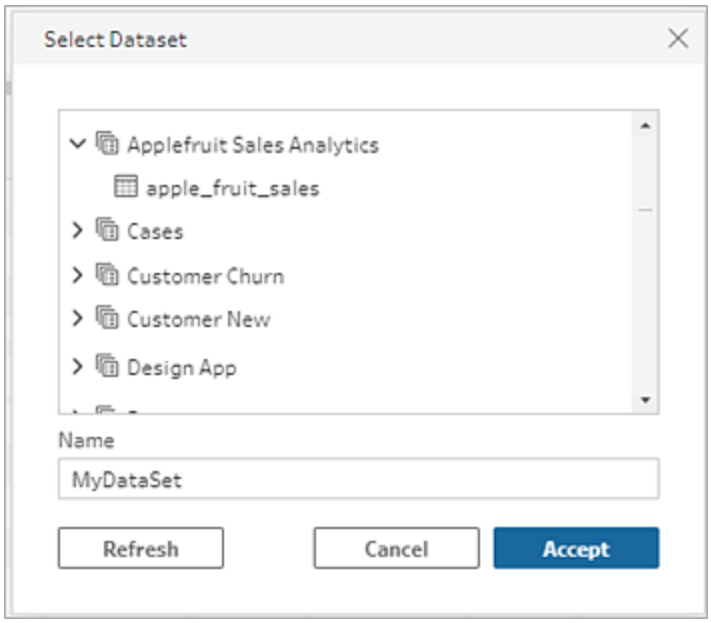
4. In the **Name** field, select an existing data set name. This will overwrite and replace the data set with your flow output. Otherwise, type a new name and click **Create new dataset** to create a new dataset in the selected CRM Analytics App.

   > **Note**: Dataset names can't exceed 80 characters.



5. Below the **Name** field, verify that the App shown is the App you have permissions to write to.

   To change the App, click **Browse Datasets**. Then select the App from the list, enter the dataset name in the **Name** field and click **Accept**.



6. In the **Write Options** section, **Full refresh** and **Create table** are the only supported options.
7. Click **Run Flow** to run the flow and write your data to the CRM Analytics dataset.

   If your flow run is successful, you can verify the output results in CRM Analytics in the Monitor tab of the data manager. For more information about this feature, see Monitor an External Data Load in the Salesforce help.

Save flow output data to Data Cloud

Supported in Tableau Prep Builder and on the web starting in version 2023.3.

Prepare your data with Tableau Prep and then associate the data with existing data sets in Data Cloud. Use any of the connectors that Tableau Prep Builder or Tableau Prep on the web supports to import your data, clean and prep your data, and then output your flow data directly to Data Cloud using Ingestion API.

Permission prerequisites

Salesforce licence	For information about Data Cloud editions and add-on licenses, see Data Cloud Standard Editions and Licences in the Salesforce help.

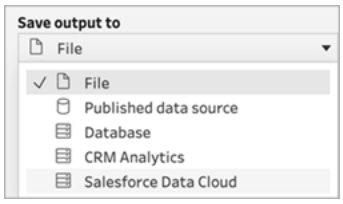
	Also see Data Cloud Limits and Guidelines.
Data Space permissions	You must be assigned to a Data Space and be assigned to one of the following permission sets in Data Cloud: • Customer Data Cloud Admin • Data Cloud for Marketing Admin • Data Cloud for Marketing Data Aware Specialist For more information, see Data Spaces and Data Space Permission Sets.
Ingestion to Data Cloud permission	You must be assigned to the following for field access for ingestion to Data Cloud: • Data Cloud Salesforce Connector For more information, see Enable Object and Field Permissions.
Salesforce Profiles	Enable the profile access for: • Administrative Permissions: API Enabled • Connected App Access: *NameOfConnectedApp*
Tableau Prep licence and permissions	Creator licence. As a creator you must sign into your Salesforce org account and authenticate before you can select Apps and Datasets to output your flow data.

Save flow data to Data Cloud

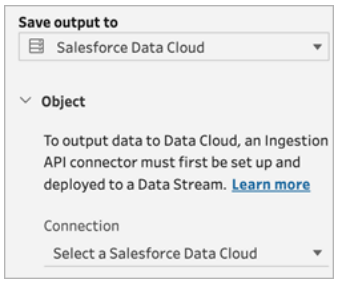
If you're already using Ingestion API and manually calling the APIs to save data sets to Data Cloud, you can simplify that workflow using Tableau Prep. The prerequisite configurations are the same for Tableau Prep.

If this is the first time you're saving data to Data Cloud, follow the setup requirements in **Data Cloud setup prerequisites** on page 409.

1. Click the plus icon ⊕ on a step and select **Add Output**.
2. From the Save output to drop-down list, select **Salesforce Data Cloud**.



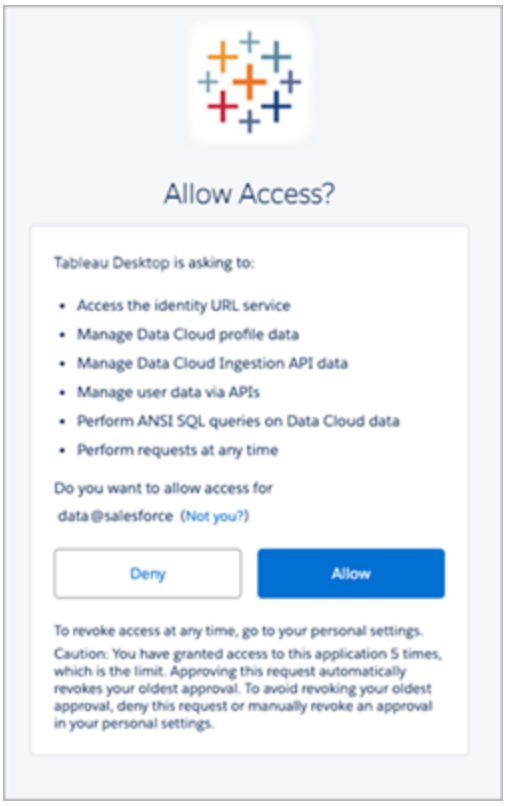
3. From the Object section, select the Salesforce Data Cloud org to sign into.



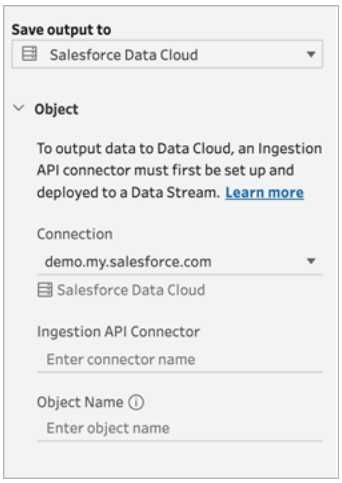
4. From the Salesforce Data Cloud menu, click **Sign In**.

   A browser window will open to *https://login.salesforce.com/*.

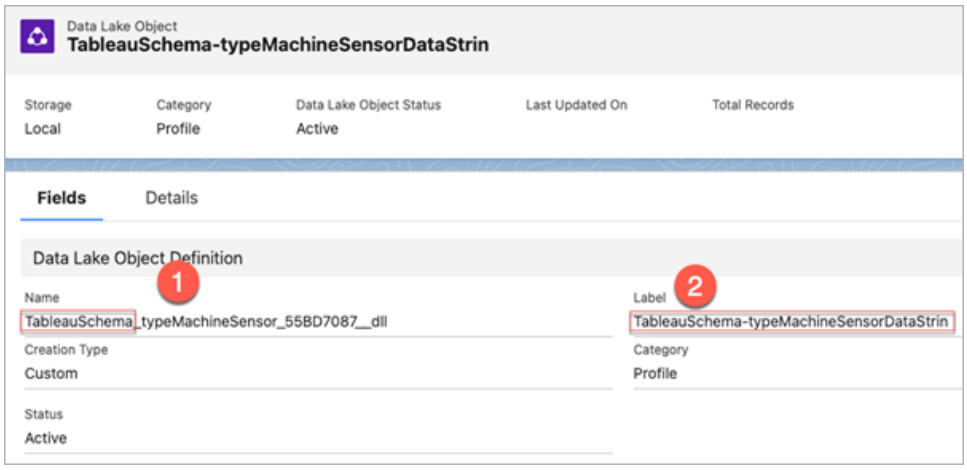
5. Sign into the Data Cloud org using your user name and password.
6. In the Allow Access form, click **Allow**.



7. In the Save output to section enter the Ingestion API connector and Object name.



The Data Lake Object name is a combination of the connector and object name: *IngestionAPIConnectorName-ObjectName*. In the example below, the connector name is *TableauSchema* (1). The Object Name is *typeMachineSensorDataStrin* (2).



8. The Write Options section indicates that existing rows will be updated if the specified value already exists in a table, or a new row will be inserted if the specified value doesn't already exist.
9. Click **Run Flow** to run the flow and write your data to Data Cloud.
10. Validate the data in Data Cloud by viewing the run status in the Data Stream and the objects in the Data Explorer.

Considerations

- You can run one flow at a time. The run must be completed in Data Cloud before another Save output can be run.
- Wait times for saving a flow to Data Cloud can take some time to complete. Check the status in Data Cloud.
- The data is saved to Data Cloud using the Upsert function. If a record in a file matches an existing record, the existing record is updated with the values in your data. If no match is found, the record is created as a new entity.
- For Prep Conductor, if you schedule the same flow to run automatically, the data won't be updated. This is because only Upsert is supported.
- You can't abort the job during the save to Data Cloud process.
- There's no validation of fields that are saved to Data Cloud. Validate the data in Data Cloud.

Data Cloud setup prerequisites

These steps are a prerequisite for saving Tableau Prep flows to Data Cloud. For detailed information about Data Cloud concepts and mapping data between Tableau data sources and Data Cloud, see About Salesforce Data Cloud.

Set up an Ingestion API Connector

Create an Ingestion API data stream from your source objects by uploading a schema file in OpenAPI (OAS) format with a .yaml file extension. The schema file describes how data from your website is structured. For more information, see the **YMAL file example** on page 413 and Ingestion API.

1. Click the **Setup** gear icon and then **Data Cloud Setup**.
2. Click **Ingestion API**.
3. Click **New** and provide a Connector name.
4. On the details page for the new connector, upload a schema file in OpenAPI (OAS) format with a `.yaml` file extension. The schema file describes how data transferred via the API is structured.

   > **Note**: Ingestion API schemas have set requirements. See **Schema requirements** on page 412 before ingestion.

5. Click **Save** on the Preview Schema form.

Create a data stream

Data streams are a data source that is brought into Data Cloud. It consists of the connections and associated data ingested into Data Cloud.

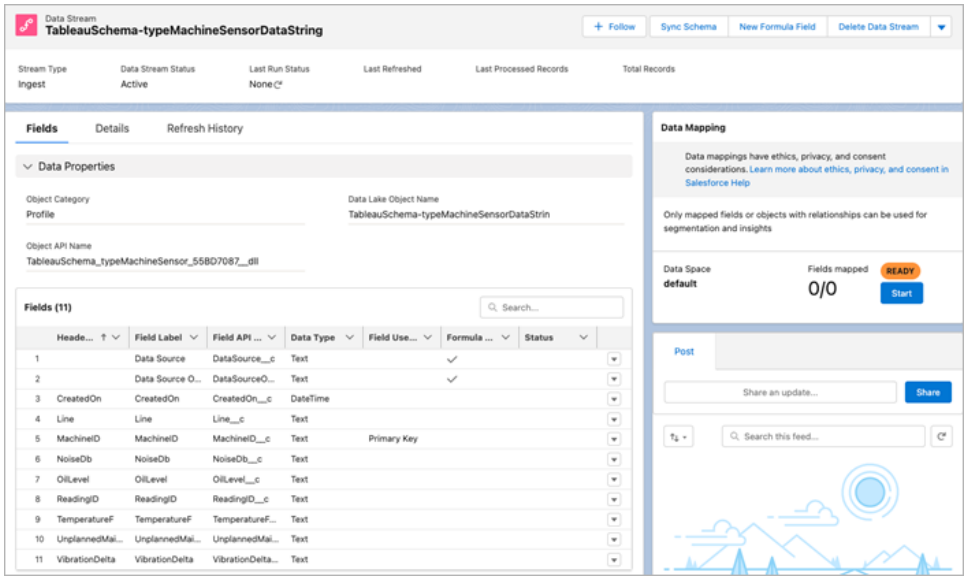
1. Go to App Launcher and select **Data Cloud**.
2. Click the **Data Streams** tab.
3. Click New and select **Ingestion API**, then click **Next**.
4. Select the ingestion API and objects.
5. Select the Data Space, Category and Primary Key, then click **Next**.

   A true Primary Key must be used for Data Cloud. If one doesn't exist, you need to create a Formula Field for the Primary Key.

   For Category, choose between Profile, Engagement or Other. A datetime field must be present for objects that are intended for the engagement category. Objects of type profile or other don't impose this same requirement. For more information, see Category and Primary Key.
6. Click **Deploy**.

   You now have a data stream and a data lake object. Your data stream can now be added

to a data space.



Add your data stream to a data space

When you bring in data from any source to the Data Cloud, you associate the Data Lake Objects (DLOs) to the relevant data space with or without filters.

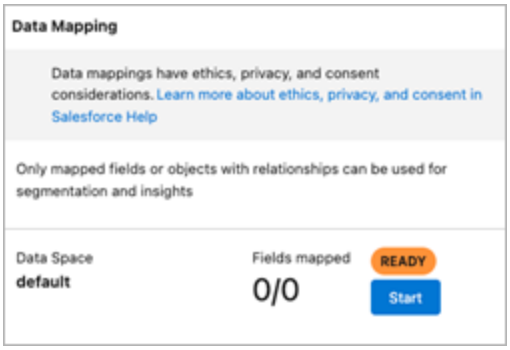
1. Click the **Data Spaces** tab.

2. Choose the default Data Space or the name of the Data Space that you're assigned to.
3. Click **Add Data**.
4. Select the Data Lake Object you created and click **Next**.
5. (Optional) select filters for the object.
6. Click **Save**.

Map the Data Lake Object to Salesforce objects

Data mapping relates Data Lake Object fields to Data Model Object (DMO) fields.

1. Go to the Data Stream tab and select the Data Stream you created.



2. From the Data Mapping section, click **Start**.

   The field-mapping canvas shows your source DLOs on the left and target DMOs on the right. For more information, see Map Data Model Objects.

Create a Connected App for Data Cloud Ingestion API

Before you can send data into Data Cloud using Ingestion API, you must configure a Connected App. For more details, see Enable OAuth Settings for API Integration and Create a connected App for Data Cloud Ingestion API.

As part of your Connected App set up for Ingestion API, you must select the following OAuth scopes:

- Access and manage your Data Cloud Ingestion API data (cdp_ingest_api)
- Manage Data Cloud profile data (cdp_profile_api)
- Perform ANSI SQL queries on Data Cloud data (cdp_query_api)
- Manage user data via APIs (api)
- Perform requests on your behalf at any time (refresh_token, offline_access)

Schema requirements

To create an ingestion API source in Data Cloud, the schema file you upload must meet specific requirements. See Requirements for Ingestion API Schema.

- Uploaded schemas have to be in valid OpenAPI format with a .yml or .yaml extension. OpenAPI version 3.0.x is supported.
- Objects can't have nested objects.
- Each schema must have at least one object. Each object must have at least one field.
- Objects can't have more than 1000 fields.

- Objects can’t be longer than 80 characters.
- Object names must contain only a–z, A–Z, 0–9, _, –. No unicode characters.
- Field names must contain only a–z, A–Z, 0–9, _, –. No unicode characters.
- Field names can’t be any of these reserved words: date_id, location_id, dat_account_currency, dat_exchange_rate, pacing_period, pacing_end_date, row_count, version. Field names can’t contain the string __.
- Field names can’t exceed 80 characters.
- Fields meet the following type and format:
    - For text or boolean type: string
    - For number type: number
    - For date type: string; format: date-time
- Object names can’t be duplicated; case-insensitive.
- Objects can’t have duplicate field names; case-insensitive.
- DateTime data type fields in your payloads must be in ISO 8601 UTC Zulu with format yyyy-MM-dd’T’HH:mm:ss.SSS’Z'.

When updating your schema, be aware that:

- Existing field data types can’t be changed.
- Upon updating an object, all the existing fields for that object must be present.
- Your updated schema file only includes changed objects, so you don’t have to provide a comprehensive list of objects each time.
- A datetime field must be present for objects that are intended for the engagement category. Objects of type `profile` or `other` don’t impose this same requirement.

YMAL file example

```
openapi: 3.0.3
  components:
    schemas:
      owner:
        type: object
        required:
          - id
          - name
          - region
          - createddate
        properties:
```

```
      id:
        type: integer
        format: int64
      name:
        type: string
        maxLength: 50
      region:
        type: string
        maxLength: 50
      createddate:
        type: string
        format: date-time
  car:
    type: object
    required:
      - car_id
      - color
      - createddate
    properties:
      car_id:
        type: integer
        format: int64
    color:
      type: string
      maxLength: 50
    createddate:
      type: string
      format: date-time
```

Save flow output data to Amazon S3

Available in Tableau Prep Builder 2024.2 and later and Web Authoring and Tableau Cloud. This feature is not available in Tableau Server yet.

You can connect to data from any of the connectors that Tableau Prep Builder or the web supports and save your flow output as a .parquet or .csv file to Amazon S3. The output can be saved as new data or you can overwrite existing S3 data. To prevent data loss, you can use the

Custom SQL option to make a copy of your table data and run it before saving the flow data to S3.

Saving your flow output and connecting to the S3 connector are independent of each other. You can't reuse an existing S3 connection that you used as a Tableau Prep input connection.

The total volume of data and number of objects you can store in Amazon S3 are unlimited. Individual Amazon S3 objects can range in size from a minimum of 0 bytes to a maximum of 5 TB. The largest object that can be uploaded in a single PUT is 5 GB. For objects larger than 100 MB, customers should consider using the multipart upload capability. See Uploading and copying objects using multipart upload.

Permissions

To write to your Amazon S3 bucket, you need your bucket region, bucket name, access key ID and secret access key. To get these keys, you will need to create an Identity and Access Management (IAM) user within AWS. See Managing access keys for IAM users.

Save flow data to Amazon S3

1. Click the plus icon ⊕ on a step and select **Add Output**.
2. From the Save output to drop-down list, select **Database and Cloud Storage**.
3. From the Table > Connection section, select **Amazon S3 (Output Only)**.
4. In the Amazon S3 (Output Only) form, add the following information:
   - **Access Key ID**: The Key ID that you used to sign the requests you send to Amazon S3.
   - **Secret Access Key**: Security credentials (passwords, access keys) used to verify that you have permission to access the AWS resource.
   - **Bucket Region**: The Amazon S3 bucket location (AWS Region endpoint). For example: us-east-2.
   - **Bucket name**: The name of the S3 bucket where you want to write the flow output. The bucket names of any two AWS accounts in the same region can't be the same.

   Note: To find your S3 region and bucket name, log in to your AWS S3 account and navigate to the AWS S3 console.
5. Click **Sign In**.

6. In the S3 URI field, enter the name of the `.csv` or `.parquet` file. By default, the field is populated with `s3://<your_bucket_name>`. The file name must include the extension `.csv` or `.parquet`.

   You can save the flow output as a new S3 object or overwrite an existing S3 object.

   - For a new S3 object, type the name of the `.parquet` or `.csv` file. The URI is shown in the Preview text. For example: `s3://<bucket_name><name_file.csv>`.
   - To overwrite an existing S3 object, type the name of the `.parquet` or `.csv` file or click **Browse** to find existing S3 `.parquet` or `.csv` files.

     > **Note**: The Browse Object window will only show files that have been saved from previous sign-ins to Amazon S3.

7. For Write Options, a new S3 object is created with the data from your flow. If the data already exists, any existing data structure or properties defined for the object are deleted and replaced with new flow data. Any fields that exist in the flow are added to the new S3 object.
8. Click **Run** Flow to run the flow and write your data S3.

   You can verify the data was saved to S3 by logging in to your AWS S3 account and navigating to the AWS S3 console.

Refresh Flow Data Using Incremental Refresh

> **Note**: Starting in version 2020.4.1, you can now create and edit flows in Tableau Server and Tableau Cloud. The content in this topic applies to all platforms, unless specifically noted. For more information about authoring flows on the web, see Tableau Prep on the Web in the Tableau Server help.

Starting in Tableau Prep Builder version 2020.2.1 and on the web, you can configure your flow inputs and outputs to refresh incrementally so that only the new rows are retrieved and processed when the flow runs, saving you time and resources.

For example, if your flow includes transaction data that updates daily, you can set up incremental refresh to retrieve and process only the new transactions every day, then run a full refresh weekly or monthly to refresh all of your flow data.

> **Note**: To run incremental refresh on flow inputs that use the Salesforce connector, you must be using Tableau Prep Builder version 2021.1.2 or later. Incremental refresh is not currently supported when writing flow outputs to Microsoft Excel or CRM Analytics.

To run your flow using incremental refresh, Tableau Prep needs the following information:

- The field that detects new rows in the input table.
- The field to use to compare the last processed values in the flow output with the values in the input to determine which rows are new. For more information, see **Incremental refresh with Append** on page 421.
- How you want to write the new data to your tables. You can either add new data to your existing tables, overwrite your table data with the new data, or starting in Tableau Prep Builder version 2020.3.1 and on the web, replace data in an existing table.

Flow refresh options

Tableau Prep enables you to select how your data is refreshed and how your tables are updated with the flow output. The following table describes the different options and their benefits.

Refresh Combination	Data Processed	Table Update	Benefits
Full Refresh + **Create Table**	All	Create or overwrite the existing table with the full data set.	Refresh all the data on every flow run.
Full Refresh + **Append to Table**	All	Add new rows to the existing table.	Keep track of both new and existing data on every flow run. **Append to table** isn't available for .csv output types.
Full Refresh + **Replace data**	All	Replace rows in the existing table.	Maintain your existing table schema structure but replace all the data with every flow run.
Incremental Refresh + **Create Table**	New rows only	Create or overwrite the existing table	Create a new table with only the new rows as the complete data set.

		with only the new rows.	If the output data source doesn't exist or can't be connected to when the flow runs, the flow will fail. A full refresh is necessary to create the output before it can be used incrementally.
Incremental Refresh + Append to Table	New rows only	Add the new rows to the existing table.	Add only the new rows to the existing table. **Append to table** isn't available for .csv output types. See **Incremental refresh with Append** on page 421.
Incremental Refresh + Replace data	New rows only	Replace all rows in the existing table with only the new rows.	Maintain your existing table schema structure, but replace all the data with only the new rows, making this your complete data set.

Configure incremental refresh

To configure your flow to use incremental refresh, you need to specify settings on both the **Input** steps and the **Output** steps where you want to use this option. In the **Input** step, specify how Tableau Prep will find your new rows. In the **Output** step, specify how the new rows are written to your table. When you run the flow, you can select either a full or incremental refresh type.

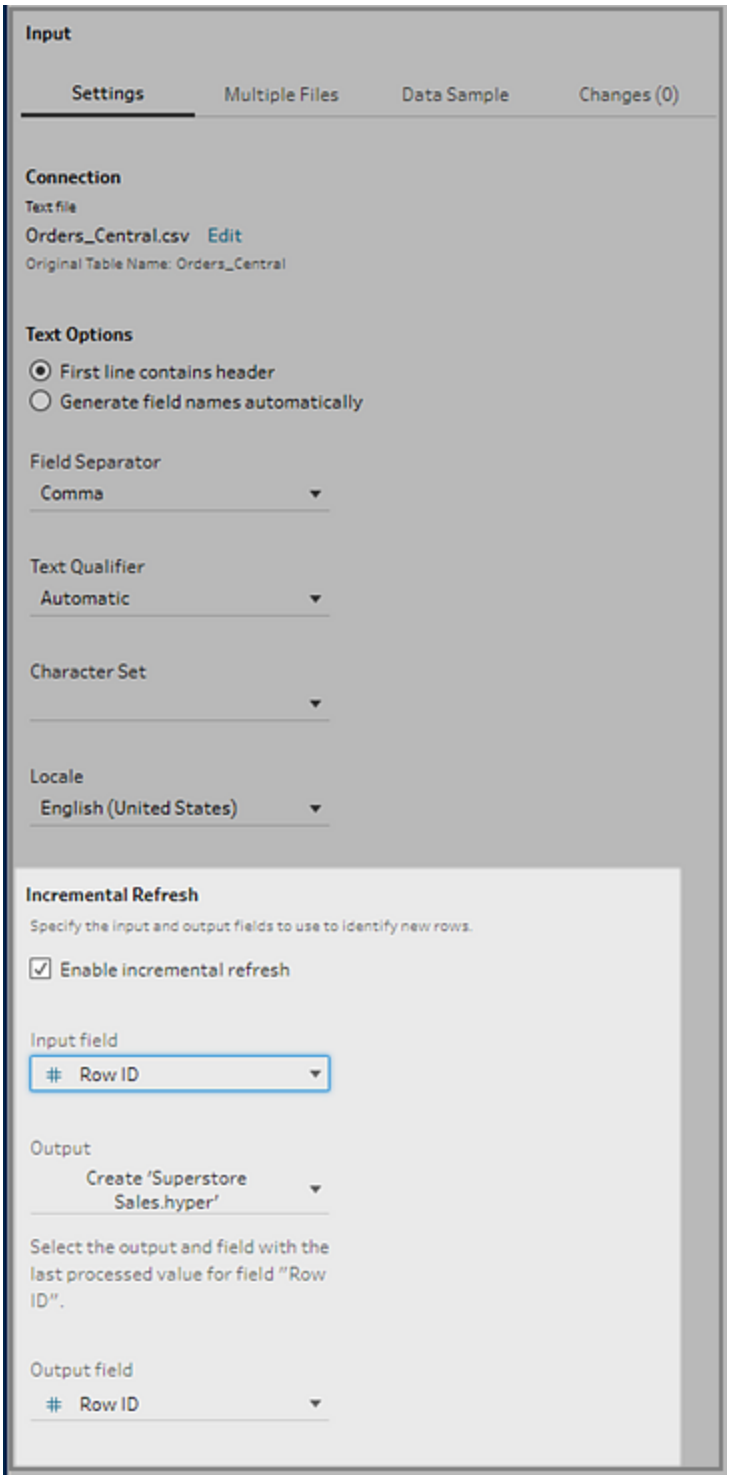
Tip: After you configure your input and output steps for incremental refresh, you can preserve your configurations and reuse them. Copy and paste the steps to use them elsewhere in your current flow or in Tableau Prep Builder, use **Save Steps as Flow** to save the selected steps to a local file or to your server to reuse the steps in other flows. For more information about copying, pasting or reusing steps, see **Copy steps, actions and fields** on page 262.

1. In the flow pane, select the input step that you want to configure for incremental refresh.
2. In the **Input** pane on the **Settings** tab, under the **Incremental Refresh** (**Set up Incremental Refresh** section in prior versions), set the following options:
    - Select **Enable incremental refresh** (**Enable** in prior versions).
    - **Input field** (**Identify new rows using field** in prior versions): Select the field that you want to refresh in your input data. This field must be assigned a data type of **Number (whole)**, **Date** or **Date & Time**. Currently, you can only select a single

field.

> **Note**: You can remove or rename this field later in the flow, as long as the field you specify in the **Output field** (**Field name in output** in prior versions) can be used to compare this field with the latest output to find new rows.

- **Output**: Select the output that is related to your input and that includes the field that will be used to compare rows.
- **Output field** (**Field name in output** in prior versions): Select the field to use to compare the last processed values in the flow output with the values in the input to find new rows. This field must have the same data type as the field you specified in the **Input field** (**Identify new rows using field** in prior versions).


Input
Settings
Multiple Files
Data Sample
Changes (0)
Connection
Text file
Orders_Central.csv
Edit
Original Table Name: Orders_Central
Text Options
First line contains header
Generate field names automatically
Field Separator
Comma
Text Qualifier
Automatic
Character Set
Locale
English (United States)
Incremental Refresh
Specify the input and output fields to use to identify new rows.
Enable incremental refresh
Input field
Row ID
Output
Create 'Superstore Sales.hyper'
Select the output and field with the last processed value for field "Row ID".
Output field
Row ID

Incremental refresh with Append

Incremental refresh first searches for the existing maximum value of the incremental field in the output. It then filters the rows from the input to add only rows with a value larger in the incremental field. For example:

Existing table:

Col1	Col2
ID 5	Row 5

Append new rows to the table based on col1:

Col1	Col2
ID 1	NewRow1
ID 6	NewRow6

- NewRow1 is not added.
- NewRow6 is added.

Configure write options

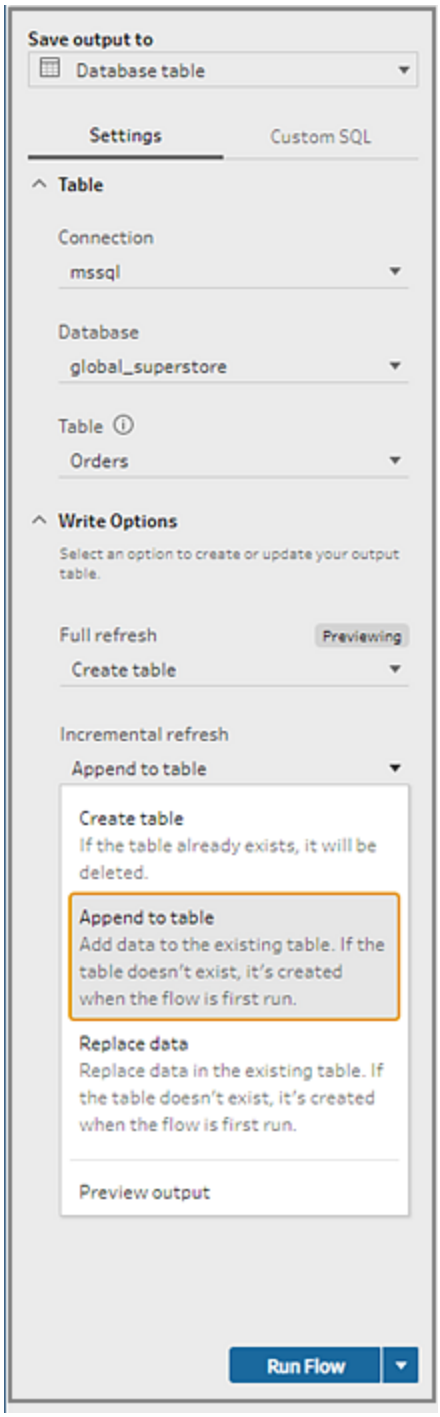
To finish setting up incremental refresh, set your output **Write Options** to specify how the new rows are written to your tables. All outputs that are related to the configured input step have a default write option selected, but you can change it to a supported option.

You can output your rows to a file (Tableau Prep Builder only), a published data source or a database. By default, outputs to local or published .hyper extracts are set to **Append to table**. Outputs to .csv file types are set to **Create table**.

1. In the flow pane, select the output step that you want to configure for incremental refresh.
2. In the Output pane, in the **Write Options** section, view the default write option and make any changes as needed.
   - **Create table:** This option creates a new table or replaces the existing table with the new output.
   - **Append to table**: This option adds the new data to your existing table. If the table doesn't already exist, a new table is created when the flow is first run and subsequent runs will add new rows to this table. Not available for .csv output types. For

more information about supported refresh combinations, see **Flow refresh options** on page 417

- **Replace data** (Tableau Prep Builder version 2020.3.1 and later and on the web): This option is available when you want to write your output back to an existing table in a database. It replaces the data in the database table with the flow data, but maintains the table schema structure.



Run your flow

You can run individual flows using incremental refresh in Tableau Prep Builder, on the web, or from the command line. For information about running your flow from the command line, see **Run**

the flow with incremental refresh enabled on page 438.

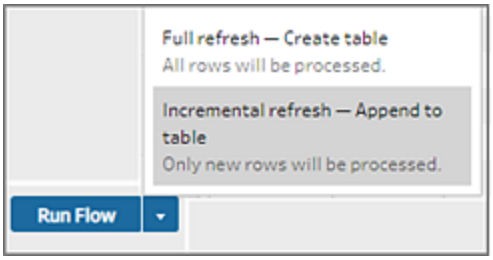
If you have Data Management with Tableau Prep Conductor enabled, you can run your flow using incremental refresh using a schedule on Tableau Server or Tableau Cloud.

> **Note**: In prior versions, write options are set in Tableau Prep Builder and can't be changed when running your flow in Tableau Server or Tableau Cloud. Starting in Tableau Server and Tableau Cloud version 2020.4, you can edit the flow directly in the web. For more information about using Tableau Prep On the web, see Tableau Prep on the Web in the Tableau Server help.

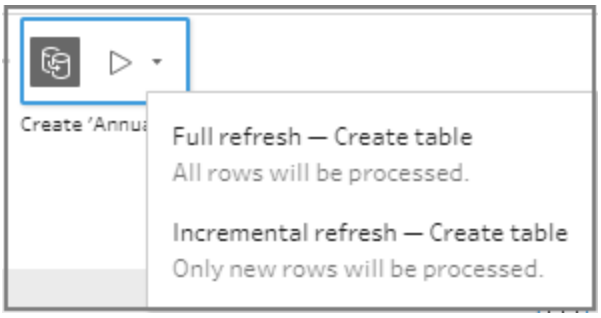
Tableau Prep runs a full refresh for all outputs regardless of the run option you select if no existing output is found. Subsequent flow runs use the incremental refresh process and retrieve and process only the new rows unless incremental refresh configuration data is missing or the existing output is removed.
To run the flow in Tableau Prep using incremental refresh, select **Incremental refresh** from one of the following locations:

- From the top menu, click the drop-down option on the **Run** button.

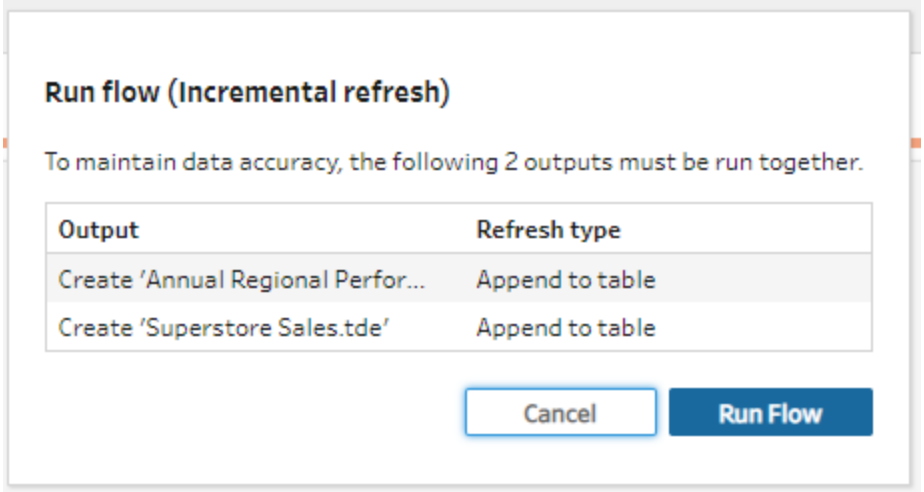
- From the Output pane, click the drop-down option on the **Run Flow** button.



- From the Flow pane, click the drop-down on the Run button next to the Output step.



If one input with incremental refresh enabled is associated with multiple outputs, those outputs must be run together and must use the same refresh type. When you run your refresh in Tableau Prep, a dialog shows letting you know that you must run both outputs together.



Refresh flow output files from the command line

Supported in Tableau Prep Builder only.

You can run your flow from the command line to refresh your flow output instead of running the flow from Tableau Prep Builder. You can run one flow at a time using this method. This option is available on both Windows and Mac machines where Tableau Prep Builder is installed.

> **Note**: If you're using Login-based Licence Management (LBLM), make sure you periodically open Tableau Prep. Otherwise the lease can expire, causing flows run via the command line to fail. You can also contact your administrator to change your lease

duration to the maximum length. See Login -based Licence Management for more information.

Connector limitations:

- **JDBC or ODBC connectors**: Flows that include these connectors can be run from the command line starting in version 2019.2.3.
- **Cloud connectors**: Flows that include cloud connectors, such as Google BigQuery, can't be run from the command line. Instead run the flow manually or run the flow on a schedule in Tableau Server or Tableau Cloud using Tableau Prep Conductor. For more information, see **Keep Flow Data Fresh** on page 457.
- **Single Sign-on authentication**: Running flows from the command line isn't supported if you use single-sign-on authentication. You can run flows from Tableau Prep Builder instead.
- **Multi-factor authentication**: The Tableau Prep Command Line Interface (CLI) does not support Tableau with Multifactor authentication (MFA). For more information, see this article in the Tableau Knowledge Base.

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

When you run flows from the command line, Tableau Prep Builder refreshes all outputs for the flow using the settings for the output steps specified in Tableau Prep Builder. For information about how to specify your output locations, see **Create data extract files and published data sources** on page 387. For information about setting your write options (version 2020.2.1 and later), see **Configure write options** on page 421.

Before running the flow

To run the flow from the command line, you'll need administrator privileges on the machine where you are running the flow and you'll need the following information:

- The path where Tableau Prep Builder is installed.
- If connecting to databases and publishing output files to a server or a database (version 2020.3.1 and later) – create a credentials .json file that includes all required credentials.
- The path where the Tableau Flow (.tfl) file is located.

Credentials .json file requirements

> **Note:** Credentials .json files are not required 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.

Tableau Prep Builder 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 sign in credentials come from the credentials .json file.

- If you plan to reuse the file, place it in a folder where it won't be overwritten by the Tableau Prep Builder install process.
- If you are running a flow that includes any of the following, you must include a .json file that includes the credentials that are required to connect.
    - Connects to database files or published data sources.
    - The output is published to a server or to a database (version 2020.3.1 and later).
    - The flow includes script steps for Rserve or TabPy. The .json file must include the credentials that are required to connect to these services. For more information, refer to the array requirements for your version below.
- The credentials specified in your flow and the credentials included in your .json file must match, otherwise the flow will fail to run.
- When you run the process, the hostname, port and username are used to find the matching connection in the Tableau flow file (.tfl) and updated before running the process. Port ID and Site ID are optional if your connections don't require this information.
- If connecting to a published data source, include hostname, contentUrl and port (80 for http and 443 for https) in the input connections. The hostname is required to find the matching connection in the Tableau flow file (.tfl), and the contentUrl and port are used to establish the connection to the server.
- If you connect to Tableau Cloud, include the port (80 or 443) in the input connections for the pod that you are connecting to, and in the Server connections URI make sure to include the corresponding pod prefix along with online.tableau.com. For more information about Tableau Cloud, see Tableau Bridge connections to Tableau Cloud in the Tableau Cloud help.
- (version 2021.4.1 and later) If you include parameters in your flow, you can create and include a parameters override .json file in the command line to change parameter values

from the current default values. For more information, see **Run flows that include parameter values** on the facing page.

Depending on your Tableau Prep Builder version, your credential information may be formatted differently. Click on the tab below to view the credential format for your Tableau Prep Builder version.

Version 2020.3.1 and later

Depending on your connections, include your server or database credentials or both. When your flow connects to and outputs to the same server or database, you only need to include a single block in the .json file. If you connect to a server or database that uses different credentials, use a comma delimited array.

Server connections	Database connections	Rserver or Tableau Python connections
Connection block name: "tableauServerConnections" Include the following data in the array: • serverUrl (Server name) For Tableau Cloud, include the corresponding pod prefix along with online.tableau.com. For example "https://10az.online.tableau.com" • contentUrl (Site ID. This appears after /site/ in the URL for Tableau Server or Tableau Cloud. For example "https://my.server-/#/site/mysite" set `"contentUrl": "mysite"`.)) • port (Port ID) • username • password	Connection block name: "databaseConnections" Include the following data in the array: • hostname (Server name) • port (Port ID) • username • password	Only include this array if your flow includes script steps for R or TabPy. Connection block name: "extensions" Include the following data in the array: • extensionName: Specify `"rSupport"` or `"pythonSupport"` • regular: Include "host"and "port". You can also include "username" or "sslCertificate" (content of your public .pem file encoded as base64 string) if applicable.

		• sensitive: Include "password" if you use one. Otherwise include a blank array.

Note: **ContentUrl** is always required in the .json file for sever connections. If connecting to a default site, for example "https://my.server/#/site/", set ContentUrl to blank. For example `"contentUrl": ""`

Run flows that include parameter values

Supported in Tableau Prep Builder version 2021.4.1 and later.

To run flows from the command line that include parameter values, you can create a parameters override .json file that includes the parameter values that you want to use. These values override the current (default) values defined for the parameters.

This is a separate file from your credentials.json file and includes your parameter names and values.

Note: Starting in version 2022.1.1, parameter values no longer need to be enclosed in quotes. In previous versions, all parameter names and values must include quotes.

Example:

```
{
    "Parameter 1": Value 1,
    "Number Parameter": 40,
    "Boolean Parameter": True
}
```

When you run the flow, include -p --parameters and the name of your file in the command line.

Examples:

Windows

```
 "\[Tableau Prep Builder install location]\Tableau Prep Builder <ver-
sion>\scripts"\tableau-prep-cli.bat -t "path\to\[your flow file
name].tfl" -p|--parameters parameters.override.json
```

Mac

```
/Applications/Tableau\ Prep\ Builder\ [Tableau Prep Builder ver-
sion].app/Contents/scripts/./tableau-prep-cli -t path/to/[your flow
file name].tfl -p|--parameters parameters.override.json
```

Examples

This section shows different examples of credentials files that you can create using the credentials .json requirements.

Connecting to a server connection

This example shows a .json credentials file that connects to and outputs to a server connection that uses the same credentials:

```
{
 "tableauServerConnections":[
   {
    "serverUrl":"https://my.server",
    "contentUrl": "mysite",
    "port":443,
    "username": "jsmith",
    "password": "passw0rd$"
   }
 ]
}
```

Connecting to a server connection and output to a database connection

This example shows a .json credentials file that connects to a server connection and outputs to a database connection:

```
{
 "tableauServerConnections":[
   {
    "serverUrl":"https://my.server",
```

```
      "contentUrl": "mysite",
      "port":443,
      "username": "jsmith",
      "password": "passw0rd$"
     }
  ],
   "databaseConnections":[
     {
      "hostname":"example123.redshift.amazonaws.com",
      "port":"5439",
      "username":"jsmith",
      "password":"p@s$w0rd!"
     }
   ]
  }
```

Flow includes Rserve and TabPy script connections and outputs to a database connection

This example shows a .json credentials file that includes Rserve and Tabpy credentials and outputs to a database connection:

```
{
 "extensions": [
    {
         "extensionName": "rSupport",
         "regular": {
               "host": "localhost",
               "port": "9000",
               "username": "jsmith"
       },
       "sensitive": {
         "password": "pwd"
       }
    },
    {
       "extensionName": "pythonSupport",
       "regular": {
```

```
          "host": "localhost",
          "port": "9000"
        },
        "sensitive": {
        }
    }
  ],
  "databaseConnections":[
    {
     "hostname":"example123.redshift.amazonaws.com",
     "port": "5439",
     "username": "jsmith",
     "password": "p@s$w0rd!"
    },
    {
     "hostname":"mysql.mydb.tsi.lan",
     "port": "3306",
     "username": "jsmith",
     "password": "mspa$$w0rd"
    }
  ]
}
```

Connecting to and outputting to different database connections

This example shows a .json credentials file that connects to and outputs to different database connections:

```
{
 "databaseConnections":[
{
  "hostname":"example123.redshift.amazonaws.com",
  "port": "5439",
  "username": "jsmith",
  "password": "p@s$w0rd!"
 },
 {
   "hostname":"mysql.mydb.tsi.lan",
```

```
    "port": "3306",
    "username": "jsmith",
    "password": "mspa$$w0rd"
   }
  ]
}
```

Version 2020.2.3 and earlier

Enter an array for your input and output connections.

> **Note**: If using Tableau Prep Builder version 2018.2.2 to 2018.3.1, 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. If you are using Tableau Prep Builder version 2018.3.2 and later you don't need to include the blank arrays.

Input connections	Output connections	Rserver or Tableau Python connections
• hostname (Server name) • contentUrl (Always required for published data sources. See **Output connections** for description.) • port (Port ID) • username • password	• serverUrl • contentUrl (Site ID. This appears after /site/ in the URL for Tableau Server or Tableau Cloud. For example "https://my.server/#/site/mysite" set `"contentUrl": "mysite"`.) • username • password	Only include this array if your flow includes script steps for R or TabPy that require a password. • extensionName: Specify `"rSupport"` or `"pythonSupport"` • credentials: Include "password".

Examples

This section shows two different examples of credentials files that you can create using the credentials .json requirements.

Connecting to a published data source

This example shows a .json credentials file that connects to a published data source and outputs data to a server that includes a Site ID

> **Note**: If the inputConnection or outputConnection uses the Default site, for example "https://my.server/#/site/", set ContentUrl to blank. For example `"contentUrl": ""`

```
{
  "inputConnections":[
    {
     "hostname":"https://my.server",
     "contentUrl": "mysite",
     "port":443,
     "username": "jsmith",
     "password": "passw0rd$"
    }
 ],
  "outputConnections":[
    {
     "serverUrl":"https://my.server",
     "contentUrl":"mysite",
     "username":"jsmith",
     "password":"passw0rd$"
    }
 ]
}
```

Connecting to two databases

This example shows a .json credentials file that connects to MySQL and Oracle and outputs data to a server that includes a Site ID.

```
{
  "inputConnections":[
    {
     "hostname":"mysql.example.lan",
     "port":1234,
     "username": "jsmith",
     "password": "passw0rd"
  },
    {
     "hostname":"Oracle.example.lan",
     "port":5678,
     "username": "jsmith",
     "password": "passw0rd"
  }
],
  "outputConnections":[
    {
     "serverUrl":"http://my.server",
     "contentUrl":"mysite",
     "username":"jsmith",
     "password":"passw0rd$"
    }
  ]
}
```

Flow includes script steps for Rserve and TabPy and connects to a database

This example shows a .json credentials file that includes the password for Rserve and TabPy services and connects to MySQL.

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

```
],
  "extensions":[
    {
     "extensionName":"rSupport",
     "credentials":{
     "password":"pwd",
    }
  },
   {
       "extensionName" : "pythonSupport",
       "credentials": {
       "password": "pwd"
       }
   }
  ]
}
```

Tips for creating your credentials file

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

- If using Tableau Prep Builder version 2018.2.2 through 2018.3.1, 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.

  If you are using Tableau Prep Builder version 2018.3.2 and later you don't need to include the blank array.

  - 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 or the port is specified as part of the server name.

If there is no port ID for your connection, don't include the `"port":xxxx,` reference in the .json file, not even `"port": ""`. If the port ID is included in the server name, include the port ID in the host name. For example `"hostname": "mssql.example.lan,1234"`

- 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.
- If connecting to published data sources, make sure to include the hostname and contentUrl in the input connections.

Run the flow

Important: The examples below include the name change for "Tableau Prep" version 2019.1.2 to "Tableau Prep Builder". If you are using an earlier version of the product use "Tableau Prep" instead.

1. Open the command prompt or terminal command prompt (MacOS) as an Administrator.
2. Run one of the following commands using the syntax shown below.
   - 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 Builder install location]\Tableau Prep Builder <version>\scripts"\tableau-prep-cli.bat -t "path\to\[your flow file name].tfl"
     ```

     Mac

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

For common errors and resolutions see **Common errors when using the command line to run flows** on page 539.

Run the flow with incremental refresh enabled

Supported in Tableau Prep Builder version 2020.2.1 and later, and on the web starting in version 2020.4. Incremental refresh is not currently supported when writing flow outputs to Microsoft Excel.

If you don't have Tableau Prep Conductor enabled on your server to schedule your flow runs, you can run your flow using incremental refresh from the command line. Simply include the parameter `--incrementalRefresh` in your command line as shown in the example below.

Windows

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

Mac

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

If the input steps in your flow have incremental refresh enabled and the incremental refresh parameters are properly configured, Tableau Prep Builder will do the following:

- All inputs in the flow that have incremental refresh enabled will run all corresponding outputs using incremental refresh.
- If no input in the flow has incremental refresh enabled, all outputs will be run using full refresh. A message will show the refresh method details.
- If some inputs in the flow have incremental refresh enabled, the corresponding outputs will run using incremental refresh. The other outputs will be run using full refresh and a message will show the refresh method details.

For more information about configuring flows to use incremental refresh, see **Refresh Flow Data Using Incremental Refresh** on page 416

Command options

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

Command options	Description	Notes
`-c,--connections <arg>`	The connection	Requires the path to where the credentials file is located.

	path to the credentials file.	
`-d`, `--debug`	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 Builder Repository\Command Line Repository\Logs`
`-dsv`, `--disableSslValidation`	Disable SSL validation (MacOS)	When running flows using the command line on the MacOS, a dialog may show asking for the keychain user and password. Starting with Tableau Prep Builder version 2019.3.2, you can pass in this additional parameter to disable this keychain dialog. For example: `/Applications/Tableau\ Prep\ Builder\ [Tableau Prep Builder version].app/Contents/scripts/./tableau-prep-cli -dsv -c path/to/[your credential file name].json -t path/to/[your flow file name].tfl`

`-h`, `--help`	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`
		`-dsv,--disableSslValidation  Disable SSL validation`
		`-h,--help  Print usage message`
		`-inc,--incrementalRefresh  Run incremental`

		`refresh for all outputs that are configured to support it`
		
		`-t,--tflFile <arg>  The Tableau Prep Builder flow file`

`-inc, --incrementalRefresh`	Run incremental refresh for all outputs that are configured to use it.	Include this option to run incremental refresh for all inputs that are configured to use it. Incremental refresh enables Tableau Prep Builder to retrieve and process only new rows instead of all rows in a flow. The incremental refresh configuration settings on the input steps determine which flow outputs can be run incrementally. All other outputs will be run using a full refresh and a message will show the refresh method details. For more information about running flows using incremental refresh, see **Refresh Flow Data Using Incremental Refresh** on page 416.
`-t, --tflFile <arg>`	The .tfl flow file	Requires the path to where the .tfl flow file is located.
`-p, --parameters`	The parameters override .json file	Include this file if you want to override the current (default) parameter values applied to your flow. For more information about using flow parameters, see **Create and Use Parameters in Flows** on page 205

Syntax examples

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

- **Tableau Prep Builder version**: 2022.1.1

  Important: The examples below include the name change for Tableau Prep version 2019.1.2 to Tableau Prep Builder. If you are using an earlier version of the product use "Tableau Prep" instead.

- **Flow name**: Flow1.tfl
- **Flow location**: C:\Users\jsmith\Documents\My Tableau Prep Builder 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 Builder 2022.1.1\scripts"\t-
ableau-prep-cli.bat -t "\C:\Users\jsmith\Documents\My Tableau Prep
Builder Repository\Flows\Flow1.tfl"
```

Mac

```
/Applications/Tableau\ Prep\ Builder\ 2022.1.1.ap-
p/Contents/scripts/./tableau-prep-cli -t /Users/jsmith/Documents/My\
Tableau\ Prep\ Builder\ Repository/Flows.Flow1.tfl
```

The flow connects to and publishes to local files and uses the short form for incremental refresh

Windows

```
"\Program Files\Tableau\Tableau Prep Builder 2022.1.1\scripts"\t-
ableau-prep-cli.bat -inc -t "\C:\Users\jsmith\Documents\My Tableau
Prep Builder Repository\Flows\Flow1.tfl"
```

Mac

```
/Applications/Tableau\ Prep\ Builder\ 2022.1.1.ap-
p/Contents/scripts/./tableau-prep-cli -inc -t /User-
s/jsmith/Documents/My\ Tableau\ Prep\ Builder\
Repository/Flows.Flow1.tfl
```

The flow connects to databases and publishes to a server

Windows

```
"\Program Files\Tableau\Tableau Prep Builder 2022.1.1\scripts"\t-
ableau-prep-cli.bat -c "\C:\Users\jsmith\Desktop\Flow cre-
dentials\Flow1.json" -t "\C:\Users\jsmith\Documents\My Tableau Prep
Builder Repository\Flows\Flow1.tfl"
```

Mac

```
/Applications/Tableau\ Prep\ Builder\ 2022.1.1.ap-
p/Contents/scripts/./tableau-prep-cli -c /Users/jsmith/Desktop/Flow\
credentials/Flow1.json -t /Users/jsmith/Documents/My\ Tableau\ Prep\
Builder\ 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 Builder 2022.1.1\scripts"\t-
ableau-prep-cli.bat -c "\\tsi.lan\files\Flow credentials\Flow1.json" -
t "\C:\Users\jsmith\Documents\My Tableau Prep Builder Repos-
itory\Flows\Flow1.tfl"
```

Mac

```
/Applications/Tableau\ Prep\ Builder\ 2022.1.1.ap-
p/Contents/scripts/./tableau-prep-cli -c /Volumes/files/Flow\ cre-
dentials/Flow1.json -t /Users/jsmith/Documents/My\ Tableau\ Prep\
Builder\ Repository/Flows.Flow1.tfl
```

Version Compatibility with Tableau Prep

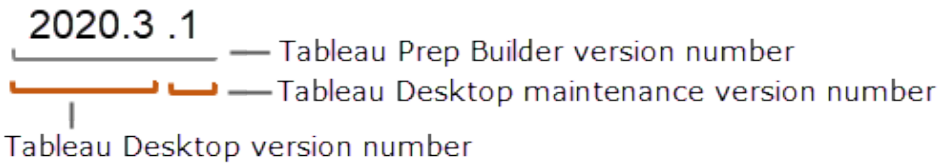
If new features or connectors are introduced in a new version of Tableau Prep Builder and you are working in an older version, compatibility may be an issue if you try to open a flow.

> **Note**: Starting in version 2020.4, you can create and edit flows directly on Tableau Server and Tableau Cloud. Flows created on the web will always be compatible with the server version you are using. For more information about authoring flows on the web, see **Tableau Prep on the Web** in the Tableau Server and Tableau Cloud help.

Similarly, if you publish flows to Tableau Server or Tableau Cloud to schedule them to run using Tableau Prep Conductor and your flows include new features or connectors that aren't supported in your version of Tableau Server or Tableau Cloud, you can run into compatibility errors that might prevent you from scheduling and running your flows.

Version number format

Starting in Tableau Prep Builder version 2022.3, the release version numbering scheme is now aligned with Tableau Desktop and Tableau Server. In prior versions, the version numbers for Tableau Desktop and Tableau Prep Builder had different formats. For example:



The maintenance releases for Tableau Desktop and Tableau Prep Builder didn't follow the same sequence.

	Release Upgrade Example	**First Maintenance Release Example**
Prep Builder	2022.1.1	2022.1.2

	Release Upgrade Example	First Maintenance Release Example
Desktop	2022.1	2022.1.1

Finding your version

> **Note**: To download a specific version of Tableau Prep Builder, open the Downloads page and select **Tableau Prep Builder** from the list on the left side of the page.

Tableau Prep Builder

To find the release version for your product, open Tableau Prep Builder, then in the top menu do one of the following:

- **Windows**: In the top menu, click **Help** > **About Tableau Prep Builder** or **About Tableau Prep**, depending on your version.
- Mac: In the top menu, click **Tableau Prep Builder** > **About Tableau Prep Builder** or **Tableau Prep** > **About Tableau Prep**, depending on your version.

The release number displays in the lower left corner of the dialog.

Tableau Server

Tableau Prep Conductor was introduced as part of Data Management in Tableau Server version 2019.1. To schedule flows to run on Tableau Server, you must be using Tableau Server version 2019.1 or later and Tableau Prep Conductor must be enabled.

To find your version of Tableau Server, open Tableau Server in your web browser. In the top menu bar, click the information icon in the top right corner and select **About Tableau Server**.

A dialog opens that tells you which version of Tableau Server you are using. For information about how to enable Tableau Prep Conductor, see Step 2: Configure Flow Settings for your Server in the Tableau Server help.

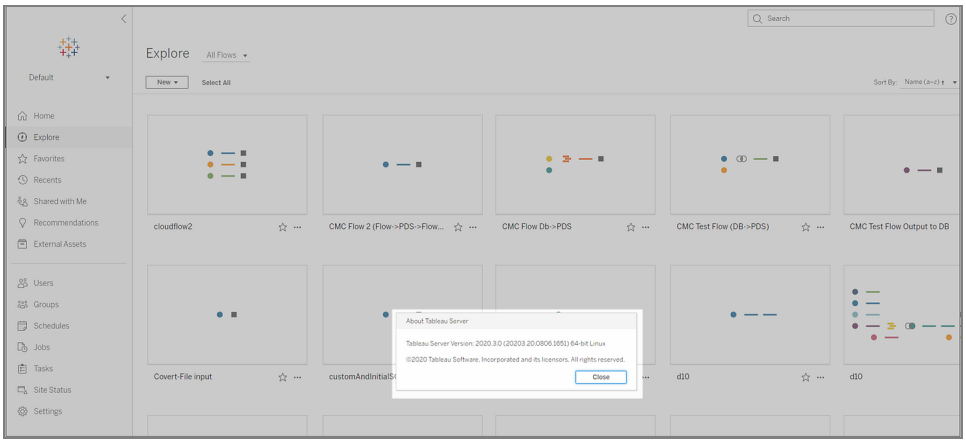
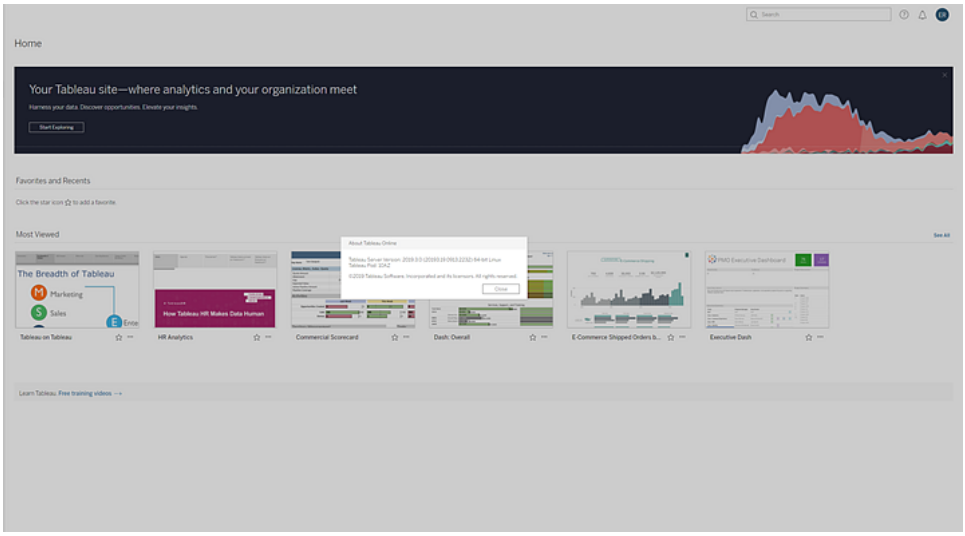


Tableau Cloud

Tableau Prep Conductor was introduced as part of Data Management in Tableau Cloud version 2019.3. To schedule flows to run on Tableau Cloud, you must be using Tableau Cloud version 2019.3 or later and Tableau Prep Conductor must be enabled.

To find your version, open Tableau Cloud in your web browser. In the top menu bar, click the information icon in the top-right corner and select **About Tableau Cloud**. A dialog opens that tells you which version of Tableau Cloud you are using. For information about enabling Tableau Prep Conductor, see Tableau Prep Conductor in the Tableau Cloud help.



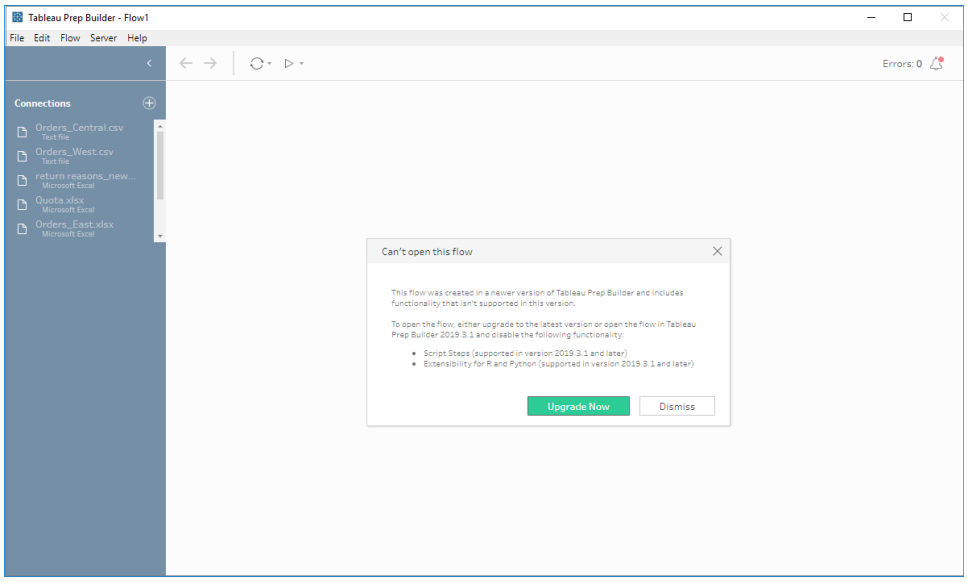
Compatibility between different versions of Tableau Prep Builder

Generally, a new version of Tableau Prep Builder can open flows created in an older version. However, compatibility issues can occur when you try to open a flow between newer and older versions of Tableau Prep Builder or even when opening flows in the same version of Tableau Prep Builder using different computers.

For example:

- The flow includes input connectors or features that aren't supported in the version where the flow is opened.
- The machine that you use to open the flow doesn't have the required input connectors installed or has a driver version for the connector that isn't compatible. Tableau Prep Builder requires 64-bit drivers to be installed to work with flow input connectors.

If compatibility is an issue, when you try to open the flow, the flow may open but contains errors or the flow won't open at all and you receive an error message. In the example below, the flow won't open and an error message displays and lists the incompatible features and options for resolving the issue.



Fix compatibility issues with Tableau Prep Builder

To fix compatibility issues, try one of the following:

- Upgrade to the latest version of Tableau Prep Builder.

  Click the update button on the bottom of the Discover pane to download the latest version of the product and follow the instructions to Install Tableau Prep Builder in the Tableau Desktop and Tableau Prep Builder Deployment Guide. If you don't have access to the update button on the Discover pane, instructions about how to download the latest version of the product are included in the **Install Tableau Prep Builder** topic.
- Make sure your computer is compatible with Tableau Prep Builder. For example, make sure that you have the 64-bit drivers installed for the connectors used by the flow. To install drivers, see the Driver Download page.
- Open a copy of the flow that has the incompatible features removed.

Compatibility between different versions of Tableau Prep Builder and Tableau Server

Publishing from a newer version of Tableau Prep Builder to an older version of Tableau Server can result in compatibility issues. For example, new features added in Tableau Prep Builder version 2021.3.1 may not be compatible with Tableau Server version 2021.2 but would be compatible with Tableau Server version 2021.4 and any later major versions of Tableau Server, such as version 2022.3.

In Tableau Server, Tableau Prep Conductor detects the features that are included in a flow when it has been published. If it finds features that it doesn't support, the flow can still be published to Tableau Server, but the flow can't be run, scheduled or added to a task. Tableau Cloud is updated automatically on a regular basis, so is generally compatible with all versions of Tableau Prep Builder.

If you have an older version of Tableau Server, you can still run incompatible flows manually in Tableau Prep Builder or using the command line. For more information about using this process, see Refresh flow output files from the command line.

Detect incompatible features

Depending on the version of Tableau Prep Builder you are using, you can spot incompatible features in different ways.

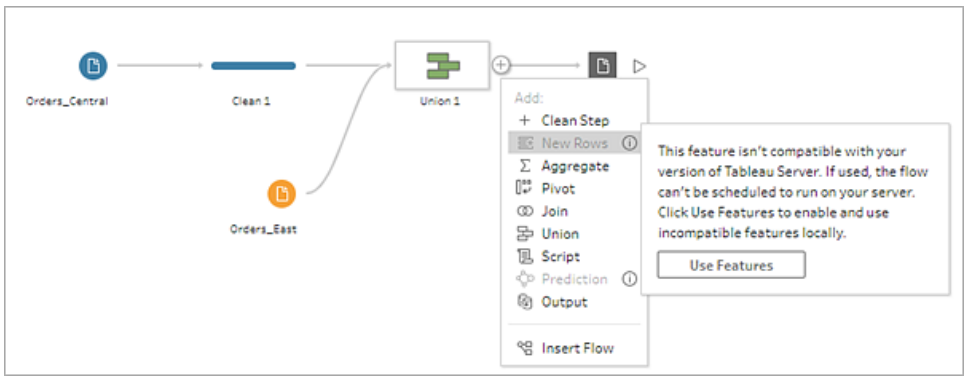
Tableau Prep Builder (version 2020.1.1 and later)

Sign into Tableau Server, and Tableau Prep Builder will detect and disable incompatible features for you. Any features that aren't compatible will show as greyed out. If you want to still use the feature and run the flow manually or from the command line, you can enable it from the menu.

> **Note**: Starting in Tableau Prep Builder version 2020.1.4, once you sign into your server, Tableau Prep Builder remembers your server name and credentials when you close the application so that the next time you open the application you are already logged into your server.

1. Hover over the disabled feature to see if it's disabled because it isn't compatible with your server version, then click the **Use Features** button. This option is available in the Flow pane and from the menus in the Profile pane, Results pane and data grid.

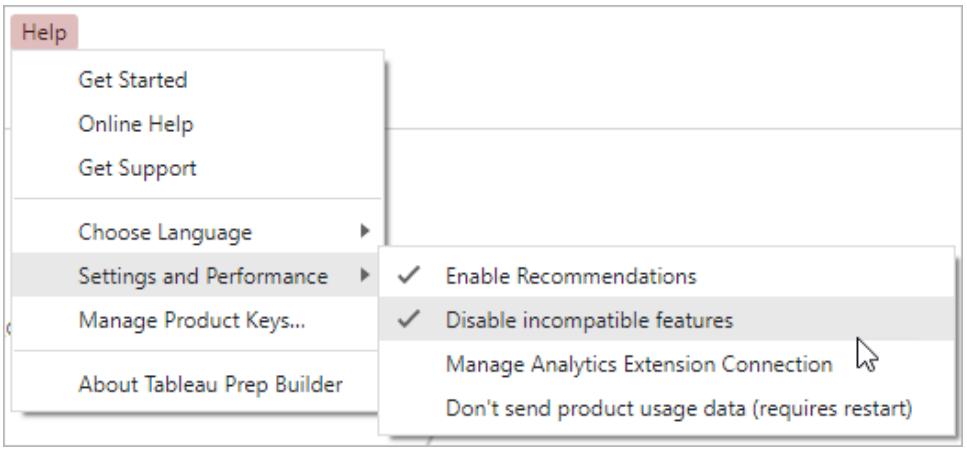
   > **Note**: Features can be disabled for other reasons, such as data updates being paused or if the option isn't available for a particular step or data type.



2. The selected feature is applied and all incompatible features are enabled and available to use. Incompatible features are flagged with a warning so that you can easily find and remove them if you want to run the flow using a schedule in your version of Tableau Server.

To disable this feature entirely and enable all incompatible features, do the following:

1. From the top menu, select **Help** > **Settings and Performance** > **Disable Incompatible Features**.



2. Select **Disable incompatible features** to clear the tick next to this option. To enable the feature again, select **Disable incompatible features**. This option should be enabled by default.

Tableau Prep Builder (version 2019.3.1 and later)

As you build your flow, Tableau Prep Builder can detect incompatible features as you add them, and it flags these features with an alert icon. You must be signed into your server to see these alerts. This alert system helps you quickly identify incompatible features in your flow so you can decide whether to keep the feature in your flow or remove it.

Hover over alerts in the Flow pane to view information about the incompatible feature, or use the alert centre to see more details. In the alert centre, click the **View in Flow** link to navigate directly to the step, annotation, field or change that triggered the warning.

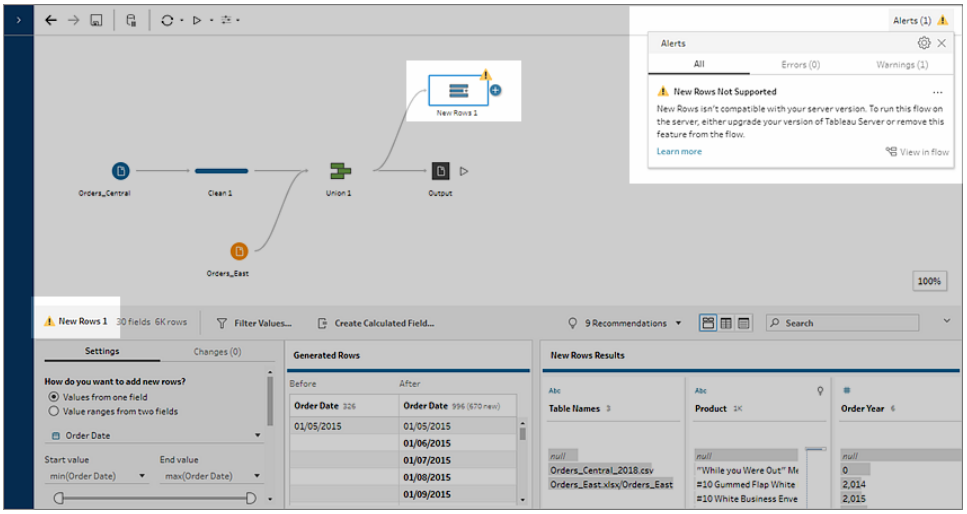
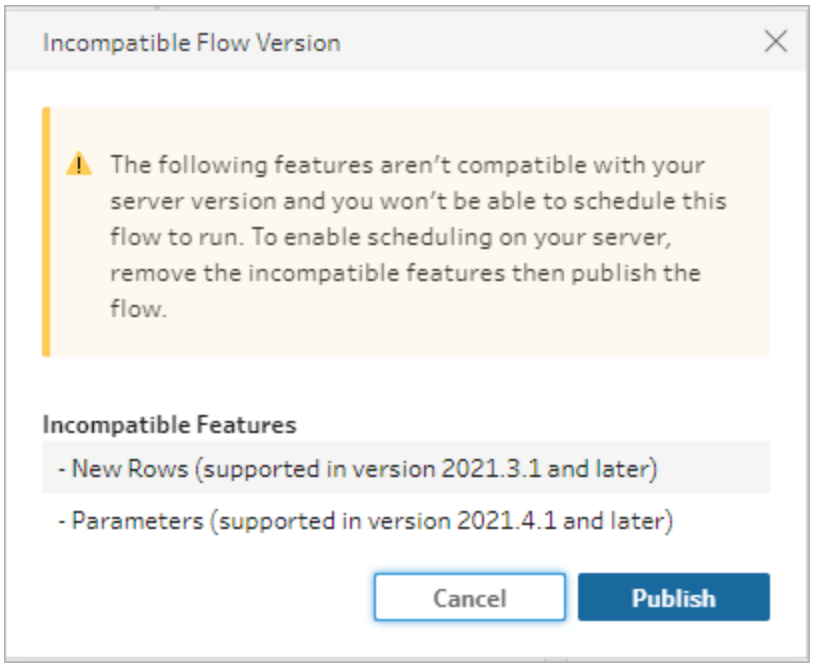


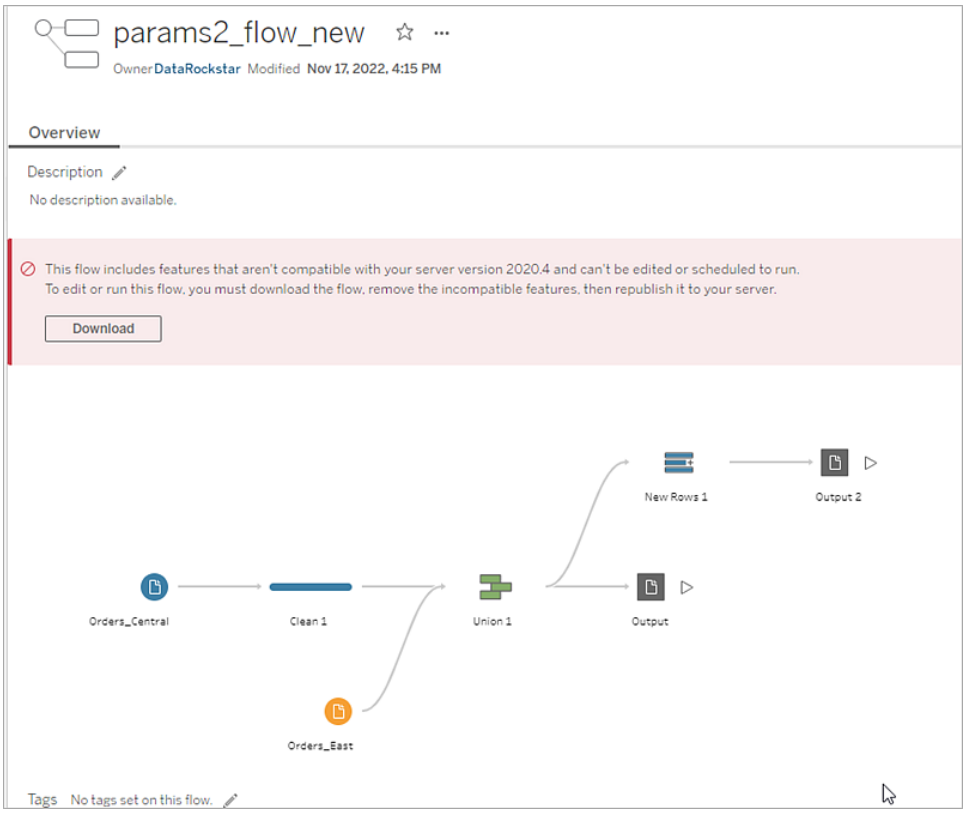
Tableau Prep Builder (all versions)

If you publish a flow with incompatible features, the following message is displayed and lists the features that aren't supported in the version of Tableau Server that you are signed into. In Tableau Prep Builder version 2019.2.3 and earlier, this is the only way to see which features are incompatible in your flow.

> **Note**: The error message lists the Tableau Prep Builder version when the feature was introduced. Tableau Prep Builder doesn't release features in maintenance versions, so for the feature to be compatible, Tableau Server must be running the next major release version. In the example below, the Duplicate Fields feature was introduced in Tableau Prep Builder version 2019.2.3 so it won't be compatible with the 2019.2.3 Tableau Server maintenance release version. Instead it would be compatible with the next major release for Tableau Server, version 2019.3.



If you continue to publish the flow, publishing will complete successfully. However, when you open the flow in Tableau Server or Tableau Cloud, you will see the following message:



To schedule and run the flow in Tableau Server, you can do one of the following:

- Look for the latest major release of Tableau Server that is compatible with the version of Tableau Prep Builder that you are using. For example, if you are using features introduced in Tableau Prep Builder version 2019.2.3, to run the flow in Tableau Server, you would need the server version to be 2020.3 or later.

  Tableau Cloud is updated automatically on a regular basis, usually every quarter. Test your flow first to make sure it is compatible with your current version of Tableau Cloud before publishing.

- Before publishing the flow, remove the incompatible features from the flow then publish the flow.

- If you already published your flow to Tableau Server, try editing the flow directly on the server (version 2019.4 and later), download the flow and remove the features, or create the flow in an older version of Tableau Prep Builder using only the features available in that version.

> **Note**: To download a specific version of Tableau Prep Builder, open the Downloads page and select **Tableau Prep Builder** from the list on the left side of the page.

Fixing compatibility issues

If the flow is already published to Tableau Server, try the steps below to remove the incompatible features using your current version of Tableau Prep Builder. After you remove the features and no longer see the version incompatibility message or warnings, republish your flow to Tableau Server or Tableau Cloud and schedule it using the Tableau Prep Conductor.

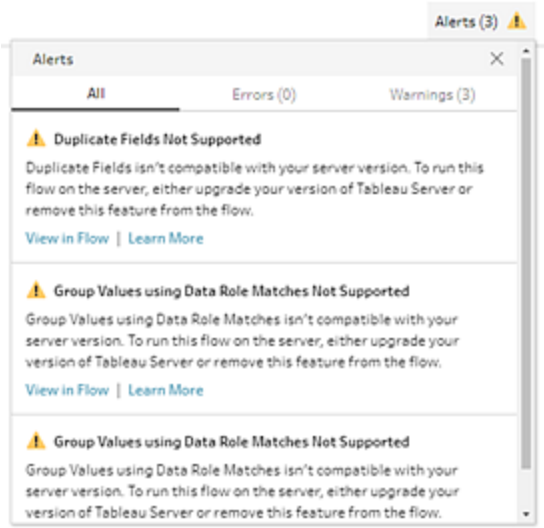
> **Note**: Tableau Prep Conductor is part of Data Management. It must be enabled in Tableau Server or Tableau Cloud to run flows using the scheduling functionality. For more information about Data Management, see Tableau Data Management. For more information about enabling Tableau Prep Conductor in Tableau Server or Tableau Cloud, see Step 2: Configure Flow Settings for your Server in the Tableau Server help or Tableau Prep Conductor in the Tableau Cloud help.

Identify incompatible features

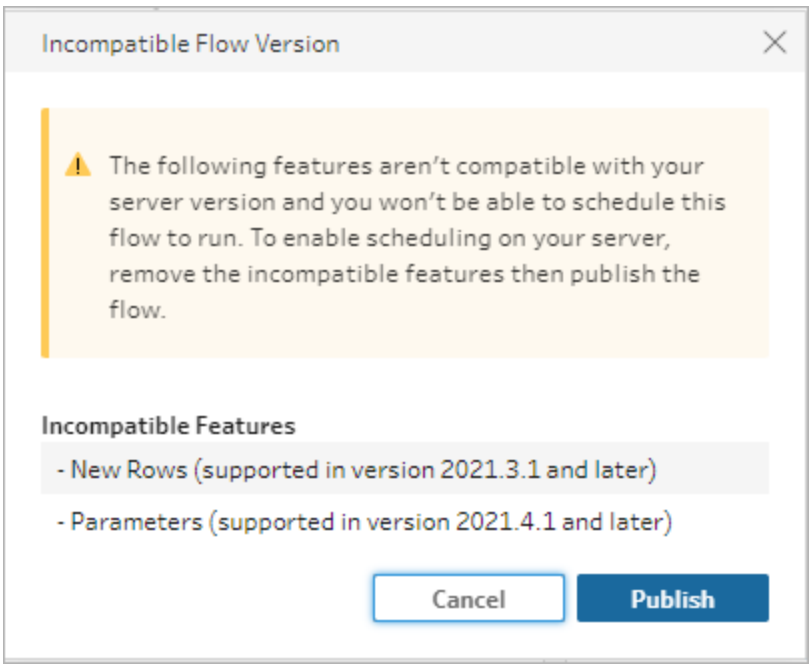
If you are working in Tableau Server, it doesn't currently list the incompatible features in your flow. To identify the list of features to remove from the flow, you need to open the flow in Tableau Prep Builder then find and remove them in your flow.

1. Open your flow. If you are in Tableau Prep Conductor, from the **More actions** ••• menu, click **Download** to download and open the flow in Tableau Prep Builder or simply open the flow in Tableau Prep Builder.
2. If you downloaded the flow, click on the downloaded flow to open it.
3. Depending on your version, do one of the following:
   - **Version 2019.3.1 and later**: From the top menu select **Server** >**Sign In**. Make sure you select the same server that is incompatible with the flow. Any incompatible steps, annotations, fields or changes should be marked with an alert icon.

     In the top right corner of the flow pane, click **Alert** to view the details for each incompatible feature. Click **View in Flow** to navigate to the incompatible feature to take action.



- **Version 2019.2.3 and earlier**: From the top menu select **Server** >**Publish Flow**. If you need to sign in to the server again, make sure you select the same server that is incompatible with the flow. A warning dialog opens which lists the features that are not compatible with your server version. Note the features so that you can identify and remove them from the flow. Then click **Cancel** to close the dialog.



4. From the top menu, click **File** > **Save As** to save a copy of your flow. Use the options in the following sections to remove incompatible features from your flow.

Remove incompatible features from the flow

You can use various methods to find and remove features from your flow. This section shows some options to help you resolve incompatibility errors.

Incompatible data sources

If the data source isn't compatible, for example a new connector was added that isn't yet supported in Tableau Prep Conductor, you'll need to connect to a data source that is supported.

To change your data connection, see **Replace your data source** on page 127.

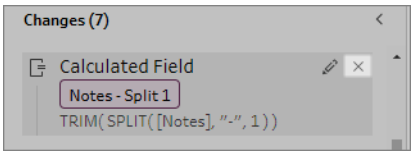
Incompatible features

To remove incompatible features you'll need to find the steps where the features were used and remove them. You can follow the instructions in **Identify incompatible features** on page 453 to locate the incompatible features.

1. If the feature is a step type, in the Flow pane click on the step where the feature is used. Right-click or Ctrl-click (MacOS) on the step and select **Remove**.
2. If the feature is a cleaning operation, in the Flow pane click on the step where the feature is used. You can hover over the annotations in the **Flow** pane or in the **Profile** or **Results** panes to see a list of changes.

   > **Note**: In Tableau Prep Builder version 2019.1.3 and later, you can hover on the icon that represents the change you are looking for over a step in the Flow pane or in the profile card then select the annotation from the list of changes. The change is highlighted in the **Changes** pane, **Profile** or **Results**pane and data grid.

3. Open the **Changes** pane if needed, and select the change that matches the feature you need to remove. Click on the change to select it and click **Remove** to delete it from the flow.



4. Repeat these steps to replace any other features. Then save your flow and republish it.

Keep Flow Data Fresh

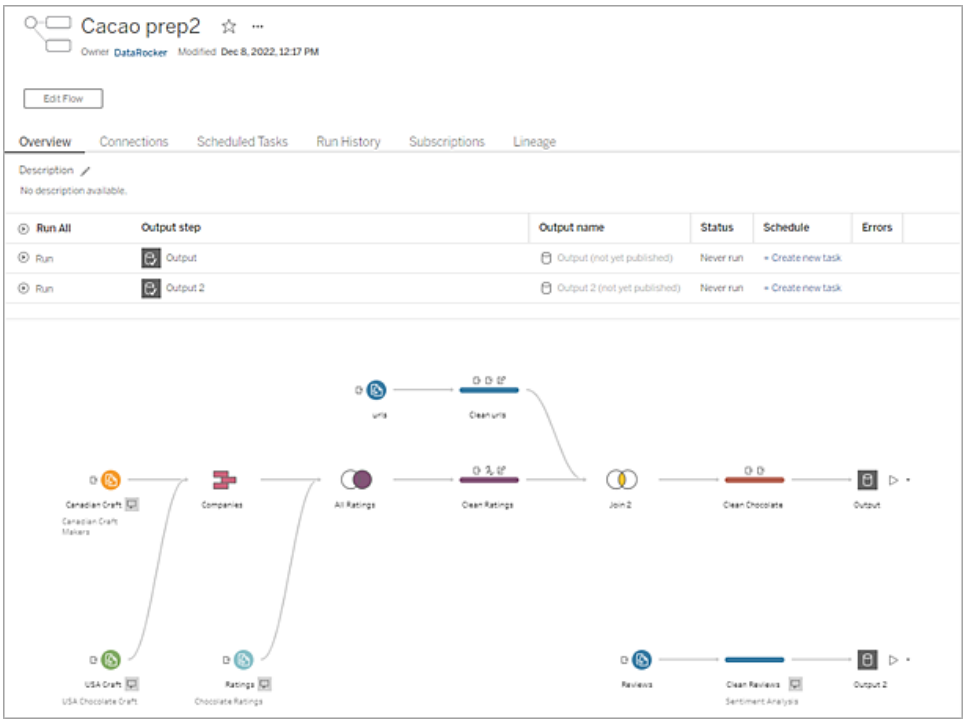
> **Note**: The content in this topic is focused on running flows on a schedule, which requires the Data Management with Tableau Prep Conductor enabled. Starting in version 2020.4.1, the Data Management isn't required to create and edit flows in Tableau Server and Tableau Cloud and to run your flows manually.

You've built your flow and cleaned your data, but now you want to share your data set with others and you want to keep that data fresh. You can manually run your flows in Tableau Prep Builder and on the web and publish an extract to Tableau Server, but now there's a better way.

Meet Tableau Prep Conductor, part of Data Management and available in Tableau Server starting in version 2019.1 and in Tableau Cloud. If you add this option to your Tableau Server or Tableau Cloud installation, you can use Tableau Prep Conductor to run your flows on a schedule to keep your flow data fresh.

For information about how to configure Tableau Prep Conductor, see Tableau Prep Conductor content in the Tableau Server and Tableau Cloud help.

And starting in version 2021.3, you can run up to 20 flows on a schedule, one after the other using the new Linked Tasks option. For more information about running flows using linked tasks, see **Schedule linked tasks** in the Tableau Server or Tableau Cloud help.



Note:If Tableau Catalog is installed, you can also see data quality warnings about your flow input data and view the upstream and downstream impact of fields in your flow on the new **Lineage** tab. For more information about Tableau Catalog, see About Tableau Catalog in the Tableau Server help.

With Tableau Prep Conductor you can do the following:

- Configure your Server or Site to use Tableau Prep Conductor
    - Enable or disable Tableau Prep Conductor for individual sites
    - Set up email notifications for flow failures for flows that are run either on-demand or using a schedule
    - Configure flow timeout settings
- Publish a flow from Tableau Prep Builder to Tableau Server or Tableau Cloud. Starting in version 2020.4.1, the Data Management is not required to publish flows to the web.
    - Upload data files or connect directly to your files (Tableau Prep Builder only) or databases. If connecting to databases, you can either embed the database credentials or require a user prompt.

> **Note**: If you connect to data files through a direct connection or publish your flow output to a file share, the files need to be in a location that Tableau Server can access. This option is not available for flows created on the web. For more information, see Step 4: Safe list Input and Output locations in the Tableau Server help.

    - Select from a project hierarchy when publishing your flows
    - Enter tags and a description to help others find your flow
- Manage the flow
    - Set permissions
    - Move the flow to a different project
    - Change the flow owner
    - Add or edit tags
    - View the version history and select from the list to restore the flow to a previous version
    - Mark a flow as a favourite and add it to your favourites list
    - Edit an input connection and update credentials
    - View data sources created from a flow and link back to the flow that created it
- Create schedules to run your flows or run your schedules on demand
    - Add scheduled tasks to run the flow and select which flow outputs to update
    - Add scheduled linked tasks to run multiple flows one after the other
- Run the flow on demand without a schedule
- Monitor the flow
    - Set up email alert notifications
    - View errors
    - Monitor and restart flows that have been suspended
    - View run history
    - Use Admin views

Run your Flow

Important: Starting in version 2020.4.1, Data Management is no longer required to run flows manually on the web. It is only required (with Tableau Prep Conductor enabled) if you plan to run your flows on a schedule.

To generate your flow output, you need to run your flow. When you run the flow, all of your data (not just the data sample you might be working with) is run through your flow steps. All of your cleaning operations are applied to your full data set, resulting in a tidy, clean data set that you can now use to analyse your data.

> **Note**: Starting in version 2021.4.1, when you run flows that include parameters, you'll be prompted to enter parameter values. You must enter required parameter values. You can also enter any optional parameter values or accept the current (default) value for the parameter. For more information about using parameters in flows, see **Run flows with parameters** on page 223.

Flow run options

Run your flows manually, from the command line, using Tableau Server REST API flow methods or using a schedule.

- **Manual**: Run your flows manually anytime in Tableau Prep Builder and on the web. The Data Management isn't required. Flows on the web must be published before they can be run. For more information, see **Publishing flows** in the Tableau Server or Tableau Cloud help.
- **Command Line interface**: If you don't have the Data Management, you can run flows one at a time using the command line interface. For more information, see **Refresh flow output files from the command line** on page 425.
- **REST API**: Use the **Flow** and **Flow Task** REST API methods in Tableau Server to run flows. The Data Management is required. For more information, see Flow Methods in the Tableau REST API help.
- **Using a schedule**: In Tableau Server and Tableau Cloud, you can schedule single flows to run or run multiple flows one after the other using linked tasks. Your server must include Data Management with Tableau Prep Conductor enabled.

For more information, see **Tableau Prep Conductor** in the Tableau Server or Tableau Cloud help. For information about scheduling your flow to run automatically, see Schedule a Flow Task in the Tableau Server help.

Run flows manually

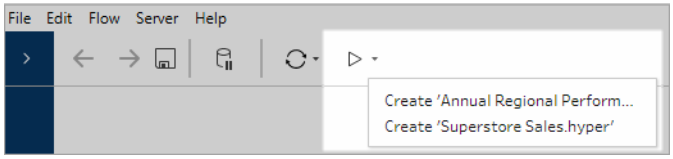
When you run flows manually, you can run one flow at a time. You can run the whole flow or run the flow for a selected output.

If running flows in web authoring (version 2020.4 and later), the flow must be published to the server to run it, and you can't run another flow until the first flow is finished, even from a separate tab. For more information, see **Publish a Flow to Tableau Server or Tableau Cloud** on page 463.

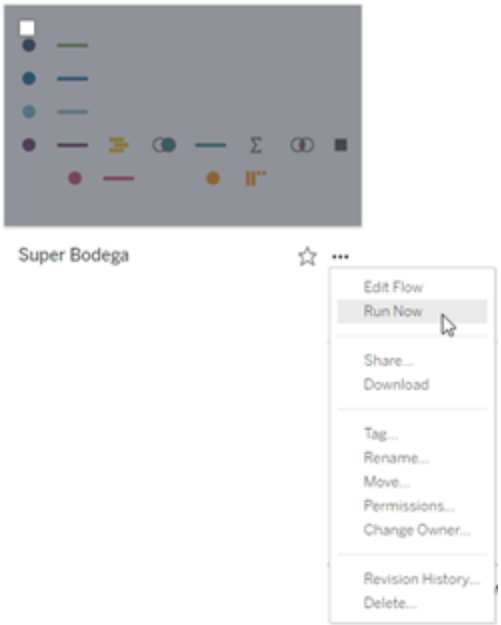
In Tableau Cloud, the number of flow runs you can perform in a day is also limited by the site administrator. For more information, see Tableau Cloud Site Capacity in the Tableau Cloud help.

1. In Tableau Prep Builder or on your server, open your flow.
2. Do one of the following:

   - From the top menu, click Run ▷ to run the entire flow, or click on the drop-down arrow to select a flow output in the list.

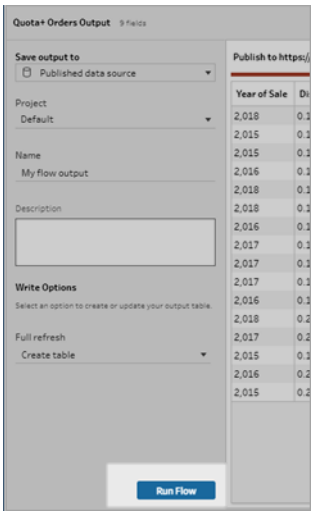


   - On the server, from the **Explore** page, right-click or Cmd-click (MacOS) on ··· **More actions** and select **Run Now** from the menu. This will run your entire flow.



- Click on an Output step in your flow, then in the Output pane, click **Run Flow**.

  If the flow isn't open on the web, you will need to click **Edit Flow** to open your flow in editing mode, then either click **Publish** to publish the flow or accept the prompt to publish the flow, then click **Run Flow**.



Publish a Flow to Tableau Server or Tableau Cloud

Important: Starting in version 2020.4.1, Data Management is no longer required to publish your flows to Tableau Server or Tableau Cloud, or run flows manually on the web. It is only required (with Tableau Prep Conductor enabled) if you plan to run your flows on a schedule.

Publish your flows to Tableau Server or Tableau Cloud to share them with others or automatically run them on a schedule and refresh the flow output using Tableau Prep Conductor. You can also manually run individual flows on the server. Flows created or edited on the web (version 2020.4 and later) must first be published before they can be run.

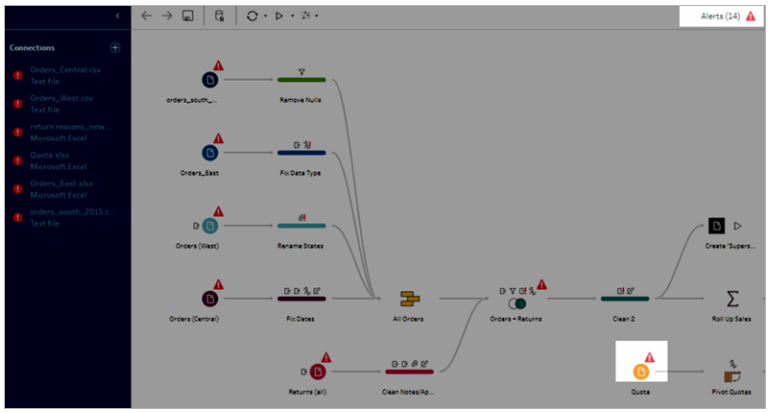
For information about publishing flows on the web, see **Publishing flows** in the Tableau Server or Tableau Cloud help. For information about running flows, see **Run your Flow** on page 460.

Before you publish

To make sure that you can run your flow, check the following:

1. Verify that there are no errors in the flow.

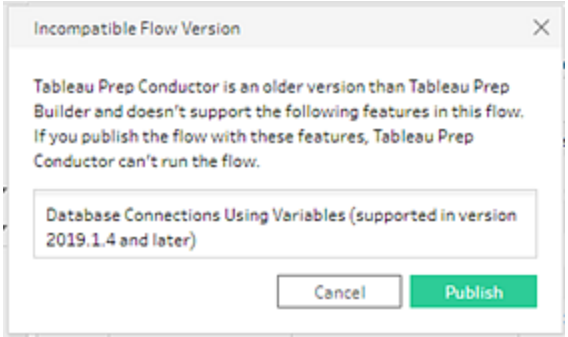
   Flows that contain errors will fail when you try to run them in Tableau Server or Tableau Cloud. Errors in the flow are identified by a red exclamation mark and a red dot with an **Errors** indicator in the upper right corner of the canvas.



2. Verify that your flow doesn't include input connectors or features that aren't compatible with your version of Tableau Server. Flows created on the web are always compatible with the server version they are created on.

You can still publish flows from Tableau Prep Builder that include connectors or features that aren't yet supported in your version of Tableau Server, but you can't schedule them to run.

For example, the SAP HANA connector was introduced in Tableau Prep Builder version 2019.1.4 but this connector isn't supported until Tableau Server version 2019.2 for Tableau Prep Conductor. When you publish the flow, you will see a message similar to the following:



> **Note**: To schedule flows to run on Tableau Server, you must be using Tableau Server version 2019.1 or later and Tableau Prep Conductor must be enabled.

To run your flow in Tableau Server, you need to take the appropriate actions to make the flow compatible. For more information about working with incompatible flows, see **Version Compatibility with Tableau Prep** on page 444.

3. Flows that include input or output steps with connections to a network share require safe listing. Tableau Cloud doesn't support this option and files must be packaged with the flow on publish.

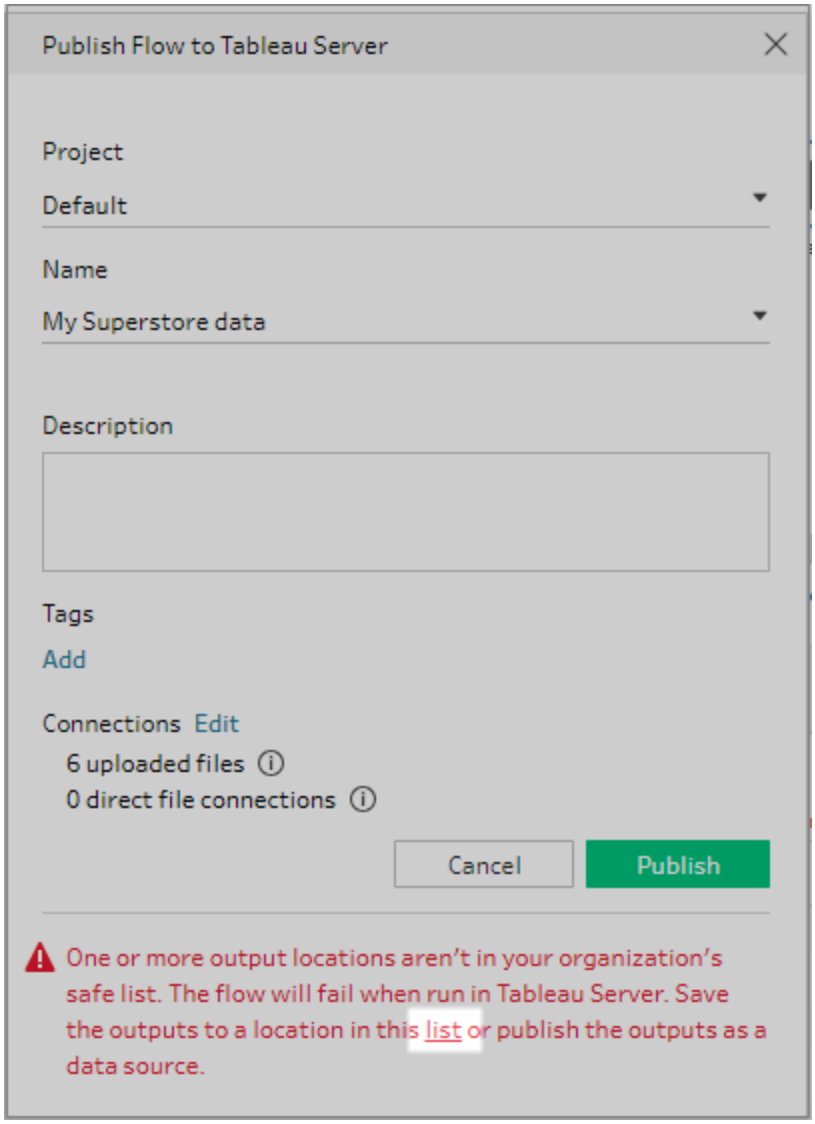
> **Note**: Currently, flows that are created on the web can only output to a published data source or a database.

Tableau Prep Builder

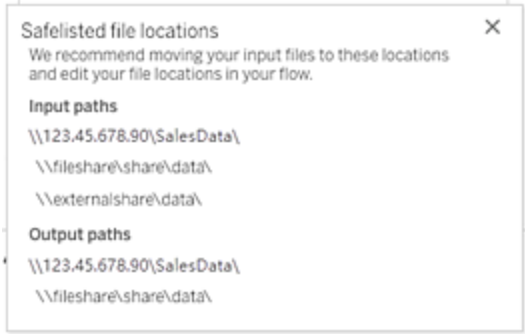
Flow input and output steps that point to files stored in a network share (UNC path) aren’t permitted unless the file and path is accessible by the server and are included in your organisation's safe list. If you publish the flow without adding the file location to your safe

list, the flow will publish, but you will get an error when you try and run the flow manually or using a schedule in Tableau Server.

If the files aren't stored in a safe listed location, you will see a warning message when you publish the flow.



Click the "list" link in the message to see a list of allowed locations. Move your files to one of the locations in the list, and make sure that your flow points to these new locations.



In Tableau Server, to configure the allowed network paths, use the tsm command options described in Step 4: Safe list Input and Output locations in the Tableau Server help.

If you don't want to move your files to a safe listed location, you will need to package the input files with the flow and publish the flow output to Tableau Server as a published data source. For more information about setting these options, see **Publish a flow from Tableau Prep Builder** on the facing page in this topic.

4. (**Tableau Prep Builder only**) If your flow output steps are set to **Publish as a data source**, all flow output steps must point to the same server or site where the flow is published. They can point to different projects on that server or site, but only one server or site can be selected.

   To set the publishing location for your output steps, do the following:

   a. In the flow pane, select the output step.

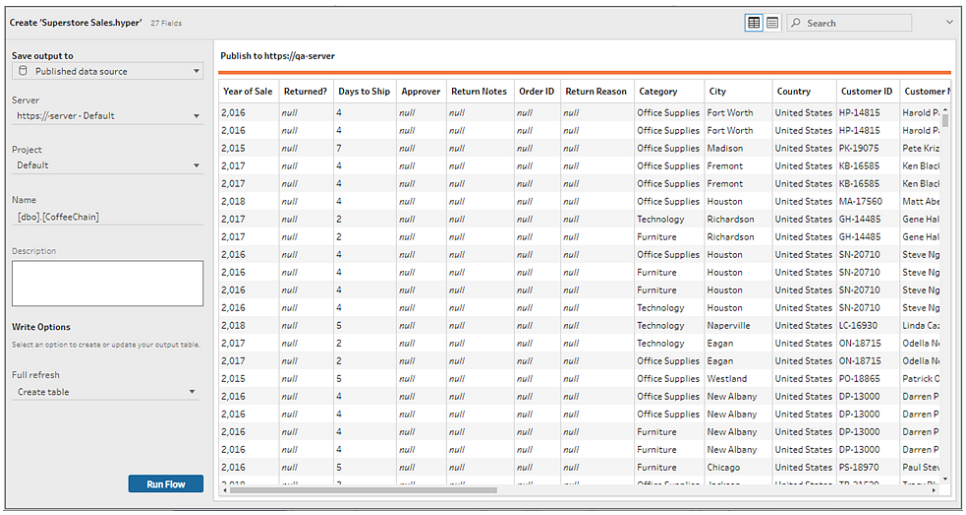
   b. In the publishing pane, select **Publish as a data source**.

   c. Select the server or site and the project where you want to publish the flow. Sign in to the server or site if needed.

   d. Enter a name and description for each output.

      The output file name should be distinctive enough so that the person running the flow can easily identify which output files to refresh. The file name shows on the **Overview** and **Connections** page for the flow in Tableau Server or Tableau Cloud.

   e. Save your flow.

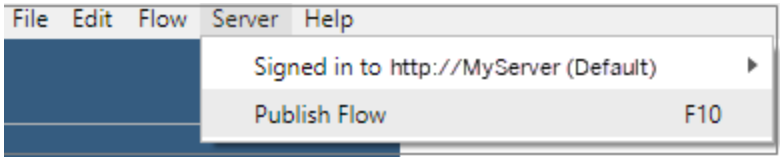


For more information about how to configure output steps for publishing, see **Create data extract files and published data sources** on page 387.

Publish a flow from Tableau Prep Builder

Note: When you publish a flow, you are automatically assigned as the default flow owner. If the flow connects to a published data source, the server uses the flow owner to connect to the published data source. Only the Site or Server Administrator can change the flow owner, and only to themselves.

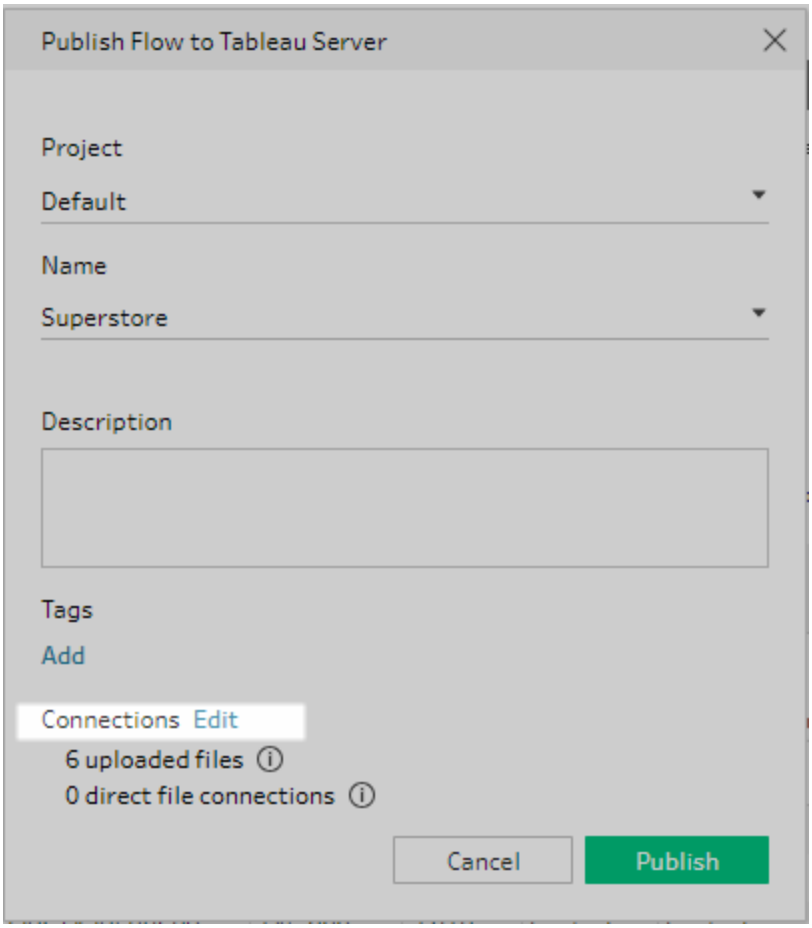
1. Open your flow in Tableau Prep Builder.
2. From the top menu select **Server** > **Publish Flow**.



3. Complete the fields for your platform. Then click **Publish**. Tableau Server or Tableau Cloud opens automatically in your default browser on the flow **Overview** page.

Tableau Server

1. In the **Publish to Tableau Server** dialog, complete the following fields:
   - **Project:** Click on the drop-down option to select your project from the project hierarchy. This should be the same project that the output files are published to.
   - **Name:** Enter a name for your flow. This name shows on the server on the Flow pages. If you want to overwrite an existing flow, click the drop-down option to select a name from the list.
   - **Description (optional):** Enter a description for the flow.
   - **Tags (optional):** Click **Add** to type in one or more tags to identify your flow so that users can easily find it. Tags can also be added after publishing in the Flow pages in Tableau Server.
2. Click **Edit** in the Connections section to edit connections settings or change authentication.

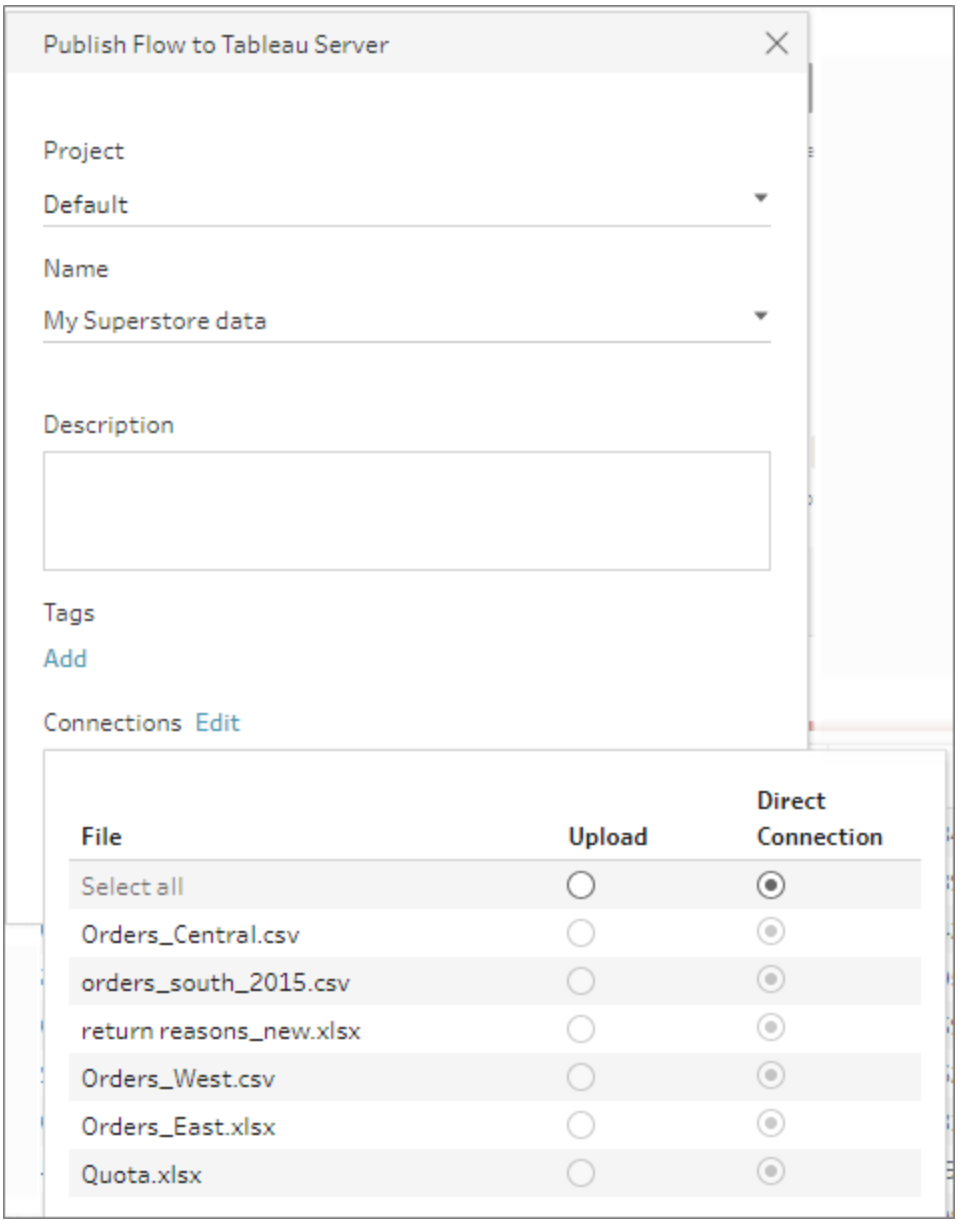


Files

By default, file input connections are packaged with the flow. Packaged files aren't refreshed when the flow is run in Tableau Server. All files must have the same setting, either **Upload** or **Direct Connection**.

Direct Connection

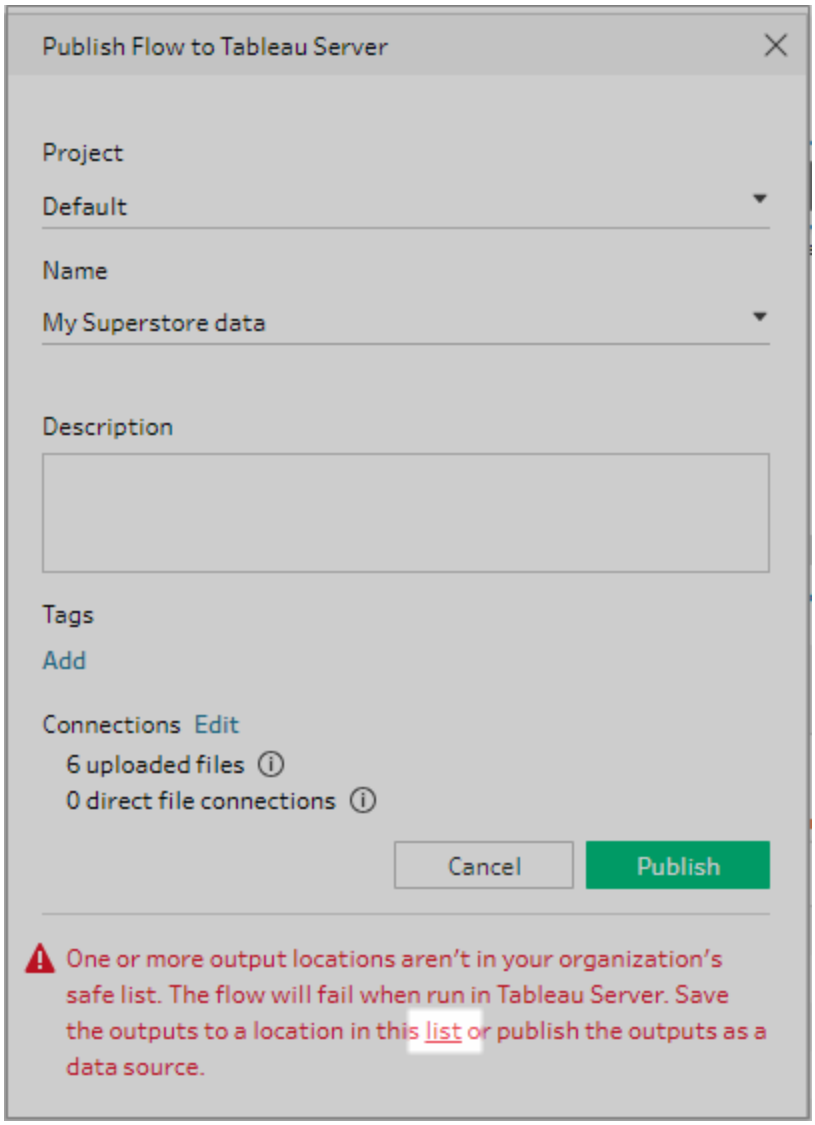
To retrieve the most current data when refreshing the output files, select **Direct Connection** if Tableau Server can connect to the file location and the location is included in your organisation's safe list.



Files stored in a network share

If your input or output steps point to files stored in a network share (UNC path) and the location isn't included in your organisation's safe list, you will see a warning message. Click the link in the message to see a list of safe listed locations, move your files and point your input and output steps to the new file location. For more information, see Step 3 in **Before you publish** on page 463.

For information about how to add locations to your organisation's safe list, see Step 4: Safe list Input and Output locations in the Tableau Server help.



Parameters in the input file path

Starting in version 2022.1.1, you can schedule and run flows on the web that include parameters in the input file path. This requires a direct file connection.

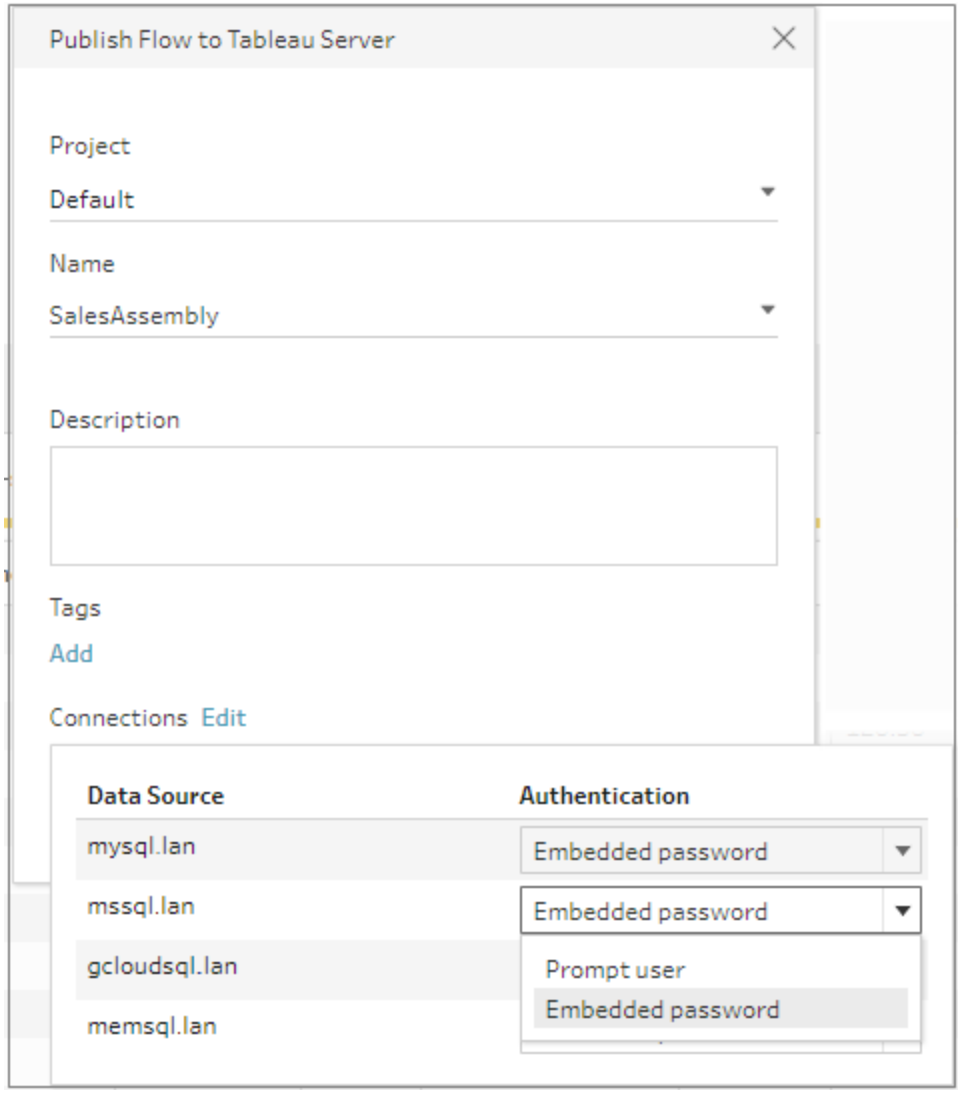
If your files are packaged with your flow or you are using an earlier version of Tableau Prep, any parameters included in the file paths are changed to the current (default) value

and the file path is made static. For more information about using parameters in flows, see **Apply parameters to input steps** on page 213.

Databases

If your flow connects to one or more databases, select one of the following authentication types to use to connect to the flow input data sources.

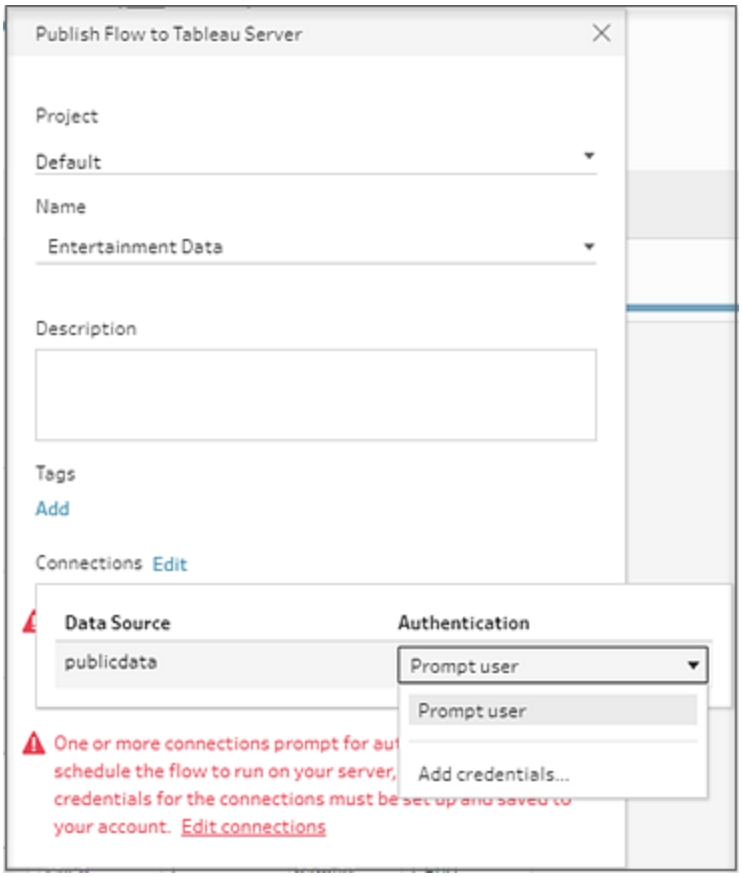
- **Server Run As Account:** The server's **Run As User** account will authenticate all users.
- **Prompt User:** You must edit the connection in Tableau Server and enter the database credentials before running the flow.
- **Embedded Password:** The credentials you used to connect to the data will be saved with the connection and used when the flow is run on a schedule. If you open the flow to edit it, you'll need to re-enter your credentials.



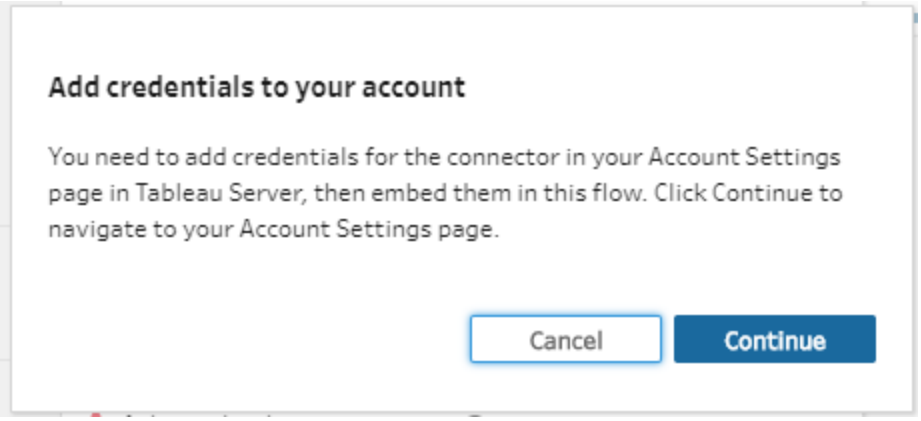
Add Credentials (version 2020.1.1 and later)

If you connect to cloud connectors, you can add your credentials directly from the **Publish Flow** dialog to embed them in the flow.

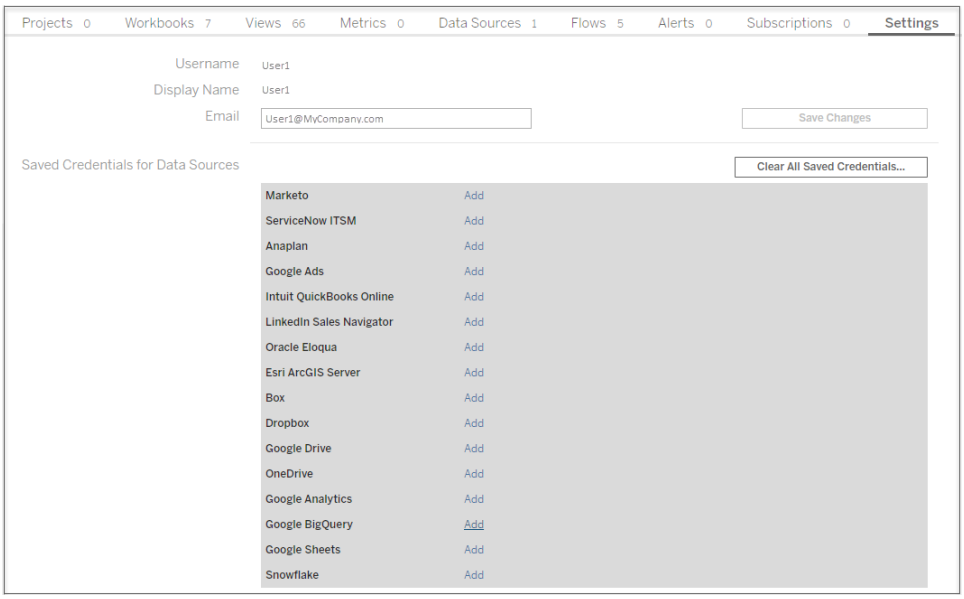
1. Click **Edit** in the Connections section, or click **Edit credentials** from the warning message. Then click **Add credentials** from the **Authentication** drop-down menu.



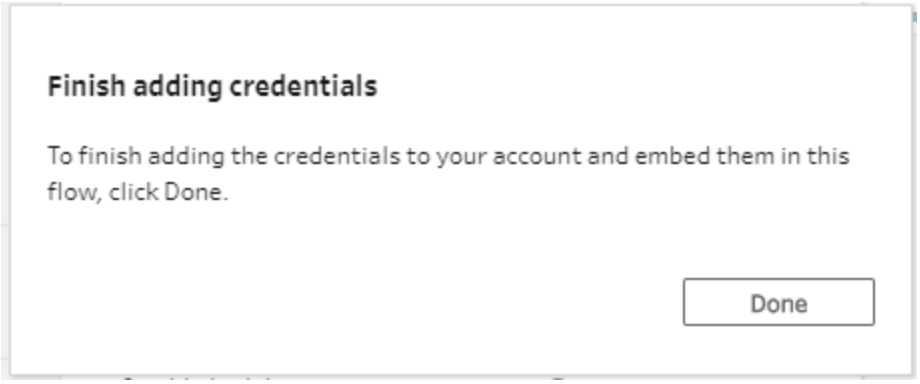
2. In the confirmation dialog, click **Continue**. Tableau Prep Builder automatically opens the Account Settings page for the server you are signed into.



3. Add your credentials, then navigate back to Tableau Prep Builder.



4. In the **Finish adding credentials** dialog, click **Done**.



5. Click **Edit** in the **Connections** section and verify that your credentials were added and embedded in your flow.

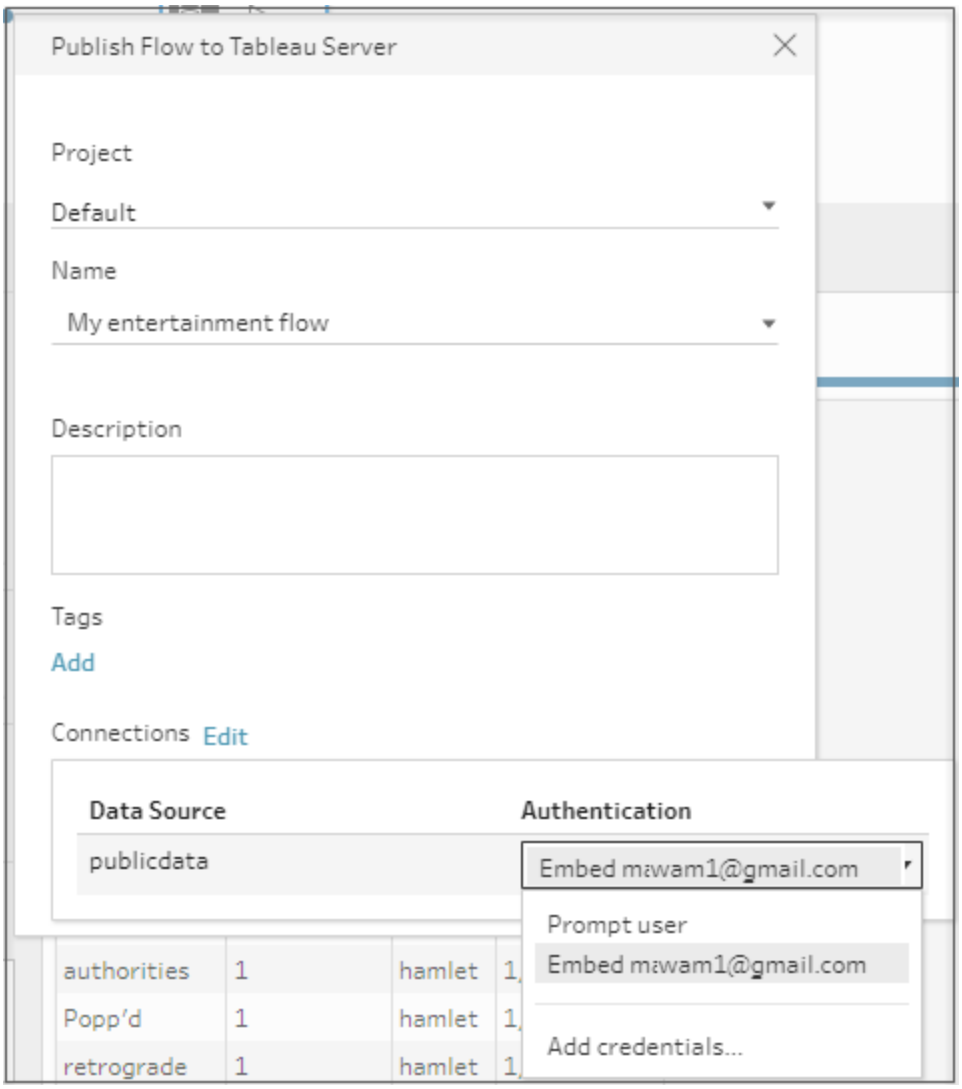
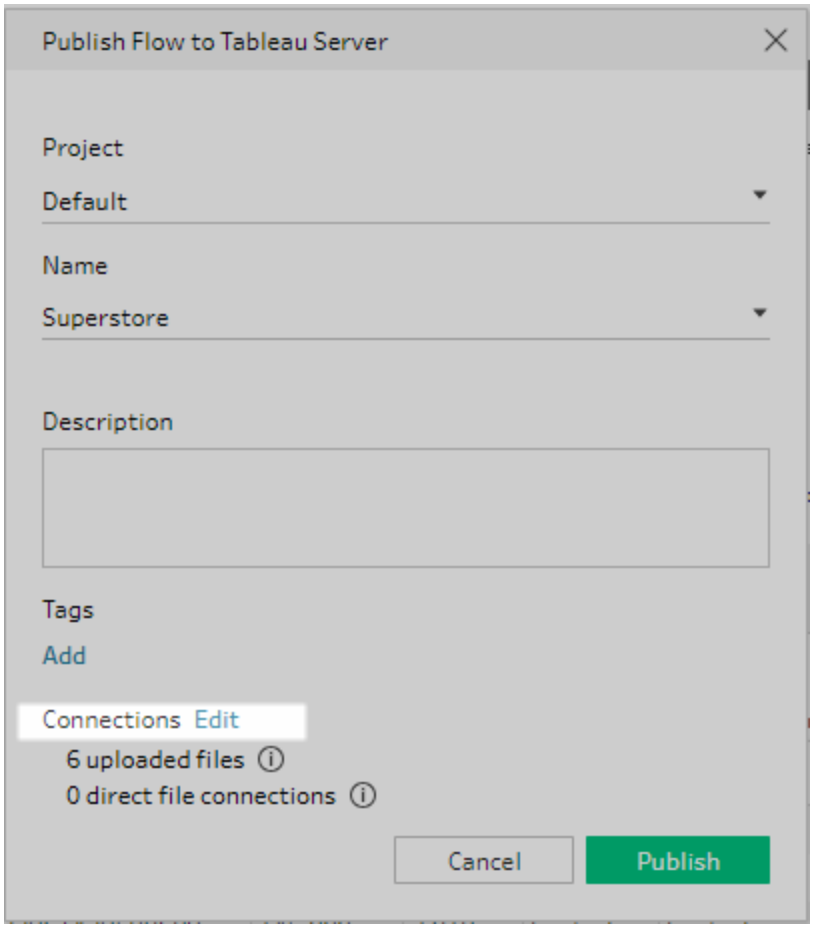


Tableau Cloud

1. In the **Publish to Tableau Cloud** dialog, complete the following fields:
   - **Project:** Click on the drop-down option to select your project from the project hierarchy. This should be the same project that the output files are published to.
   - **Name:** Enter a name for your flow. This name shows on the server on the Flow pages. If you want to overwrite an existing flow, click the drop-down option to select a name from the list.
   - **Description (optional):** Enter a description for the flow.

- **Tags (optional):** Click **Add** to type in one or more tags to identify your flow so that users can easily find it. Tags can also be added after publishing in the Flow pages in Tableau Server.

2. Click **Edit** in the Connections section to edit connections settings or change authentication.



Files

Tableau Cloud doesn't support direct file connections for input step data and you must package your files with the flow. Packaged files aren't refreshed when the flow is run in Tableau Cloud.

> **Note**: Scheduling and running flows that include parameters in the input file path isn't currently supported in Tableau Cloud because this requires a direct file

connection. When you publish the flow, any parameters included in the file paths are changed to the current (default) value and the file path is made static.

As an alternative, you can run flows with parameters in the file path in Tableau Prep Builder or using the command line. For more information about using parameters in flows, see **Apply parameters to input steps** on page 213.

Databases

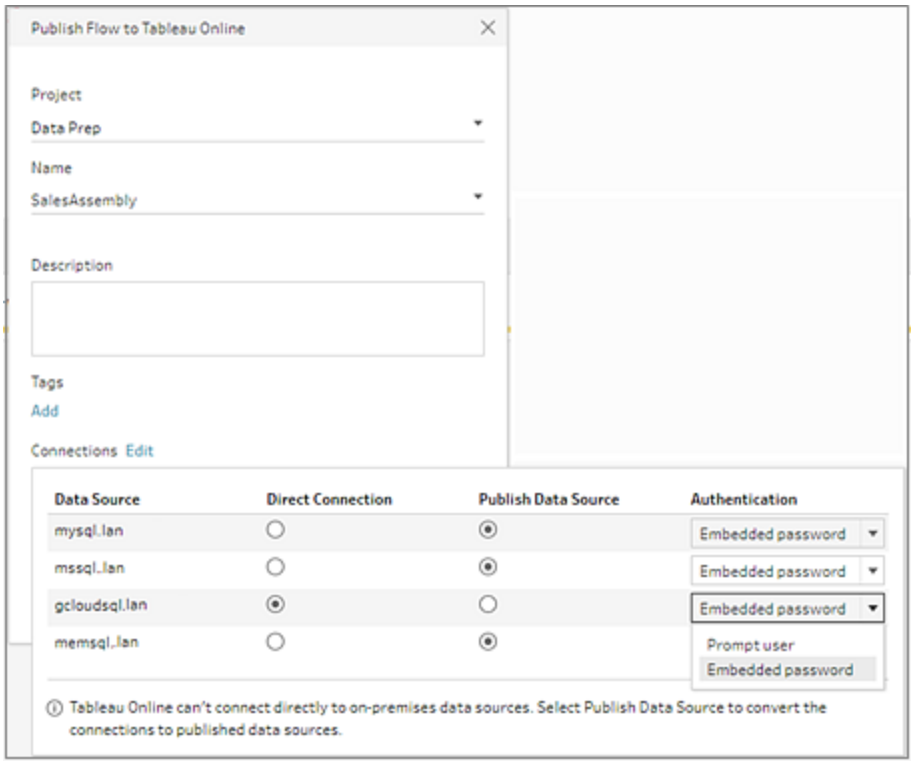
To keep data fresh when publishing flows to Tableau Cloud, you can only connect directly to cloud-hosted data sources. When connecting to on-premises data sources, you must convert the data sources to a published data source and Tableau Cloud can use a Tableau Bridge client to connect to your data if Tableau Bridge is configured for the data source.

For more information about direct connections supported by Tableau Cloud, see Allow Direct Connections to Data Hosted on a Cloud Platform.

For more information about using a Tableau Bridge, see Allow your Publishers to Maintain Live Connections to On Premises Data.

If your flow connects to a cloud-based data source that supports a direct connection, select one of the following authentication types to use to connect to the flow input data sources.

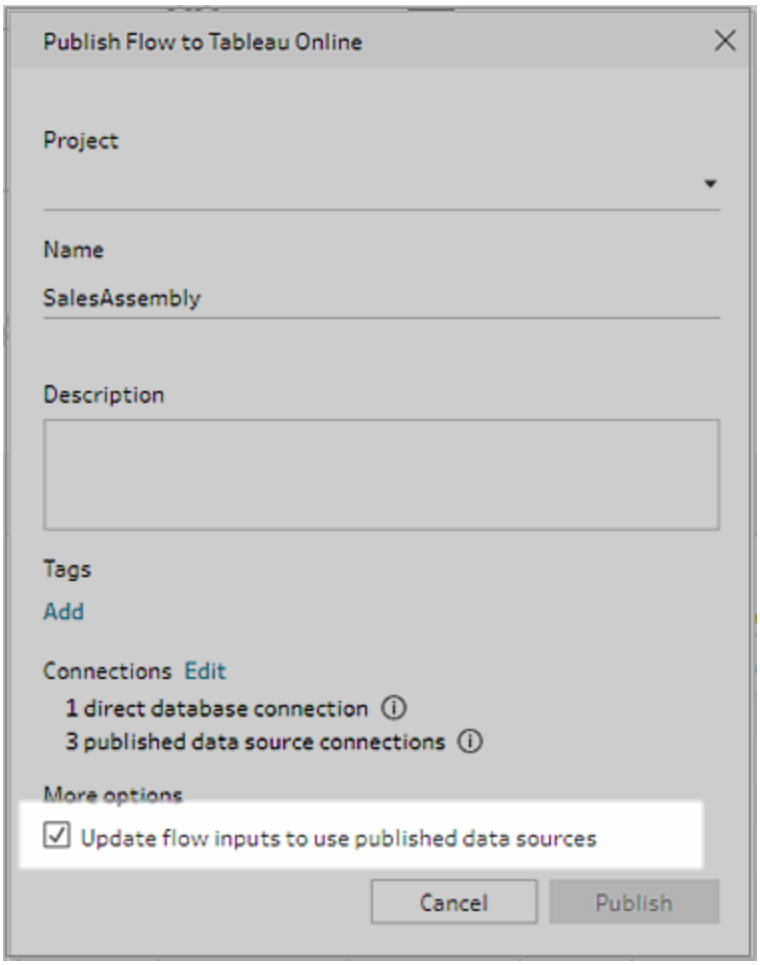
- **Prompt User:** You must edit the connection in Tableau Cloud and enter the database credentials before running the flow.
- **Embedded Password:** The credentials you used to connect to the data will be saved with the connection and used when the flow is run on a schedule. If you open the flow to edit it, you'll need to re-enter your credentials.



- Select the **Publish Data Source** radio button for on-premises data sources. Tableau Cloud can't connect directly to these data sources to refresh your data. Selecting this option converts the data source input connection to a published data source when you publish the flow to Tableau Cloud.

  If Tableau Bridge is configured for the data source and the data source is supported by Tableau Cloud, the data can be refreshed when the flow is run. See Allow Direct Connections to Data Hosted on a Cloud Platform for more information.
- To replace the on-premises data source connections for the flow in Tableau Prep Builder with the published data source, select **Update flow inputs to use published data sources** in the **More options** section before publishing your flow.

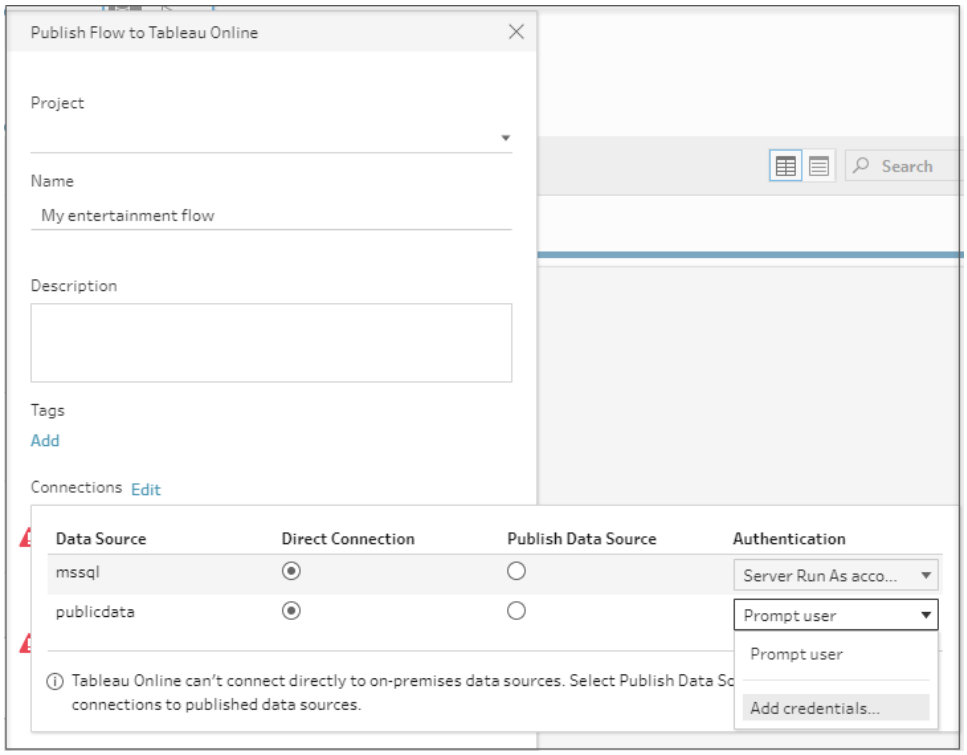
  If you don't tick the box, the flow in Tableau Prep Builder remains connected to the local on-premises data source and the flow in Tableau Prep Builder can become out of sync with the published version of the flow. To continue working with your flow, you would need to download the flow from Tableau Cloud to edit it, then republish it.



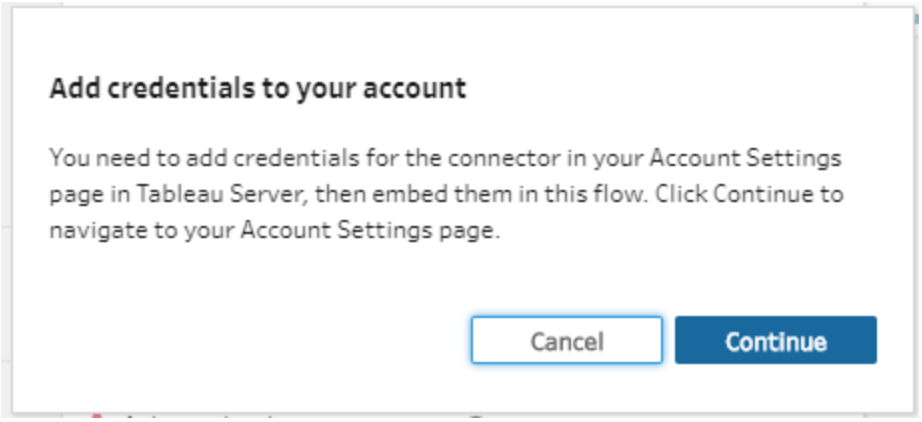
Add Credentials (version 2020.1.1 and later)

If you connect to cloud connectors, you can add your credentials directly from the **Publish Flow** dialog to embed them in the flow.

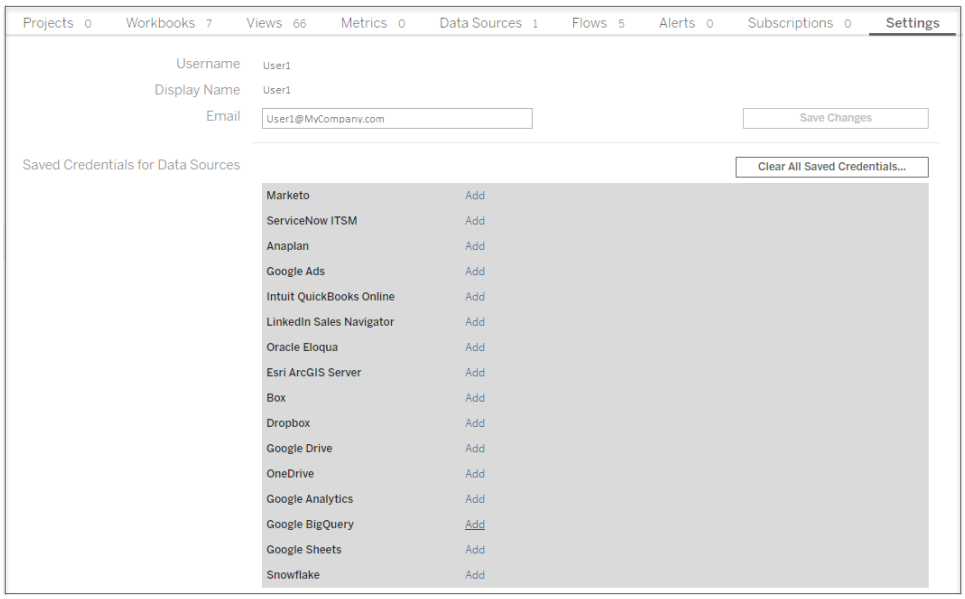
1. Click **Edit** in the Connections section, or click **Edit credentials** from the warning message. Then click **Add credentials** from the **Authentication** drop-down menu.



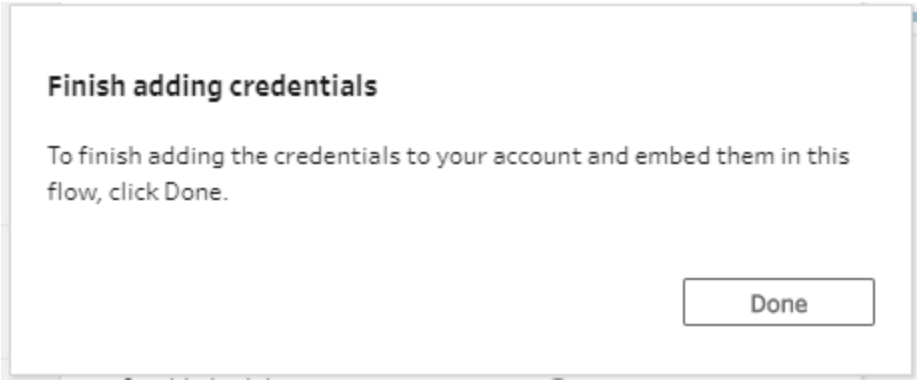
2. In the confirmation dialog, click **Continue**. Tableau Prep Builder automatically opens the Account Settings page for the server you are signed into.



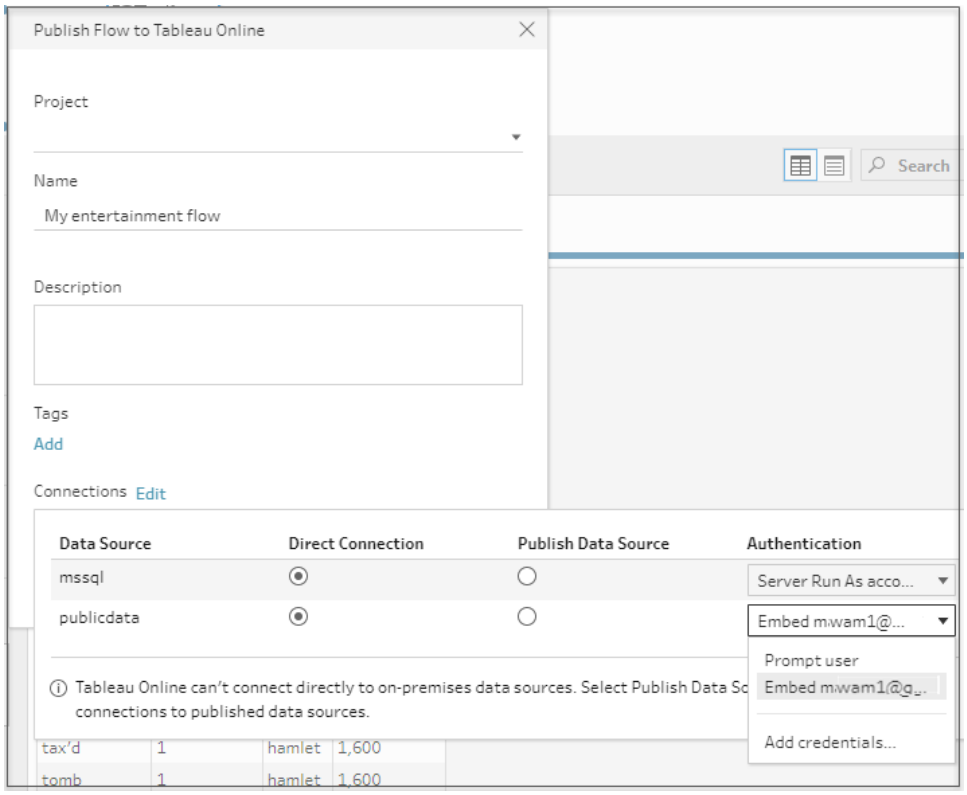
3. Add your credentials, then navigate back to Tableau Prep Builder.



4. In the **Finish adding credentials** dialog, click **Done**.



5. Click **Edit** in the **Connections** section and verify that your credentials were added and embedded in your flow.



Who can do this

- Server Administrator, Site Administrator Creator and Creator allow full connecting and publishing access.
- Creator can perform web authoring tasks.
- Explorer (can publish)

Day in the Life Scenarios

What does it mean to shape data? How does that impact what visualisations can be built and what analysis can be performed? In the tutorials below, we explore scenarios for analysis and visualisation, 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 Desktop or Tableau Prep Builder from the User Interface 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 over-abundance 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 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

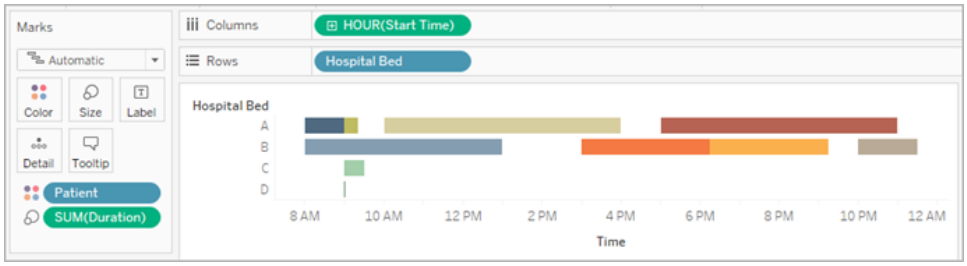
The Data

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 visualisations.

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

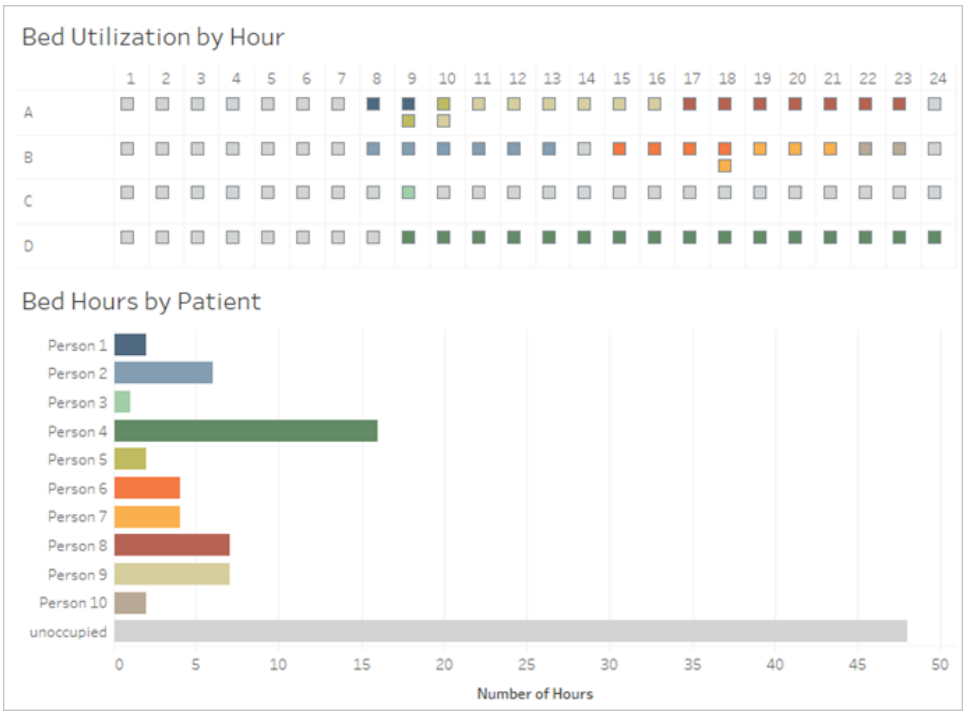
	A	B
1	Bed	Hour
2	A	1
3	B	1
4	C	1
5	D	1
6	A	2
7	B	2

Next, we'll bring in the **Patient Beds** information, labelling 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 75.

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** and click the suggested **Clean 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.

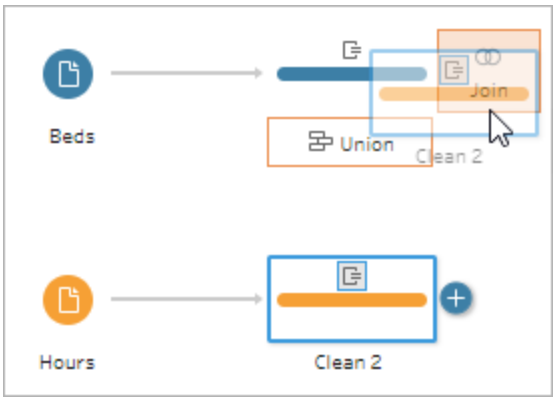
Click for directions

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** and click the suggested **Clean Step** to add it to the flow.
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

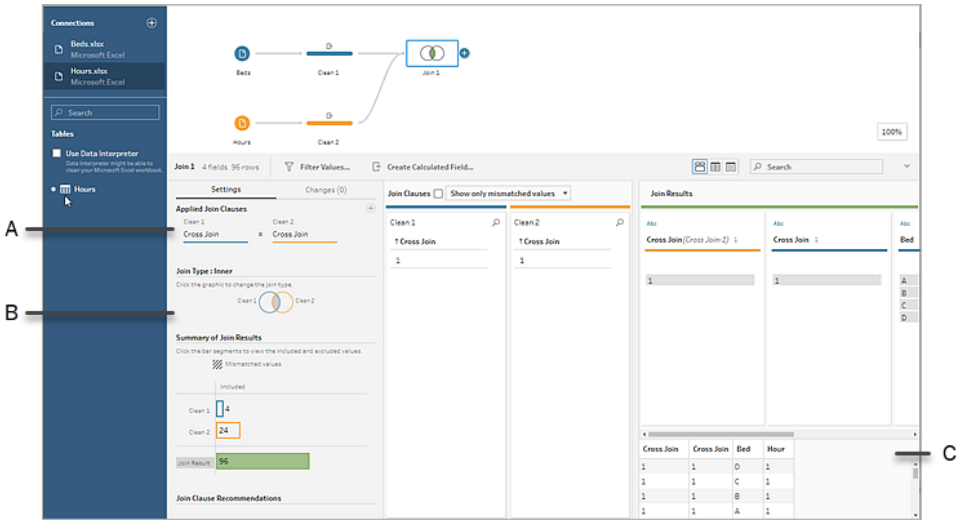
Both data sets now have a shared field, **Cross Join**, and can be joined.

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 360.

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 **Clean 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**, then click the suggested **Clean Step** to add it to the flow.

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).`

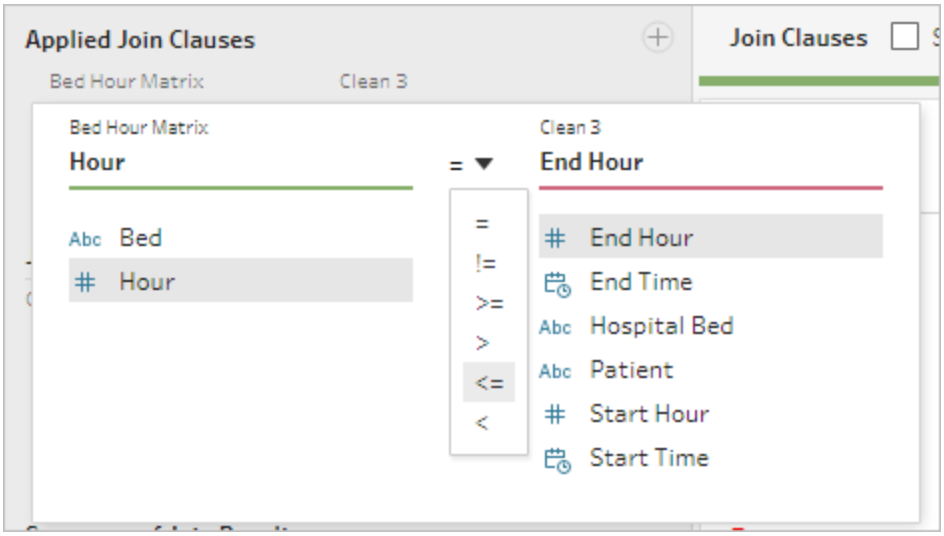
The `DATEPART` portion takes the hour of the end time. The `IFNULL` portion will assign an end time of 24 (midnight) to any missing end time.

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). So in addition to matching the bed the patient is in we also 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.

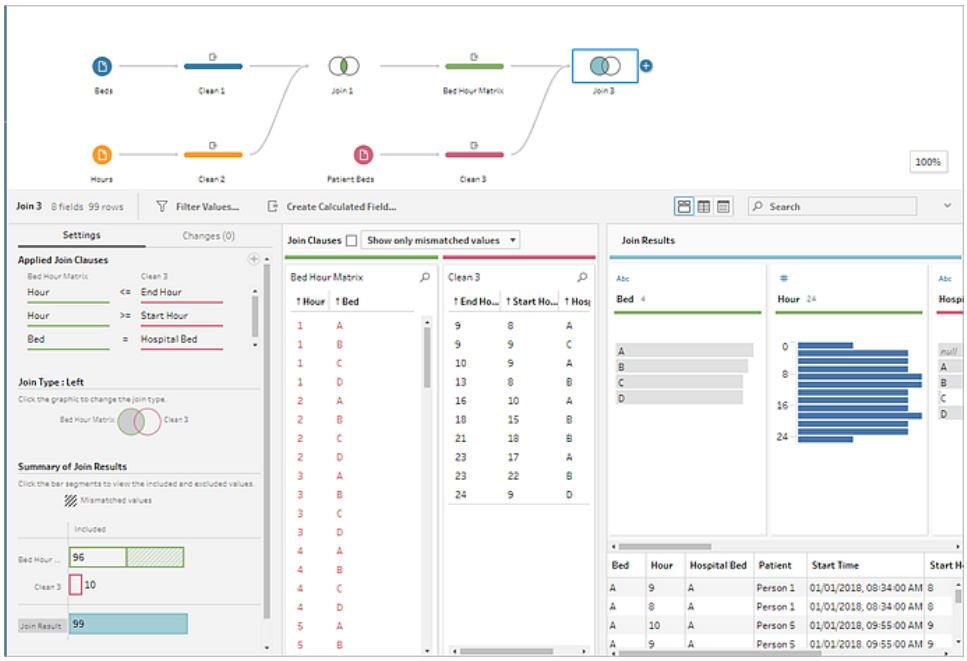
Therefore, three join clauses are needed to appropriately match these two data sets together.

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 383.

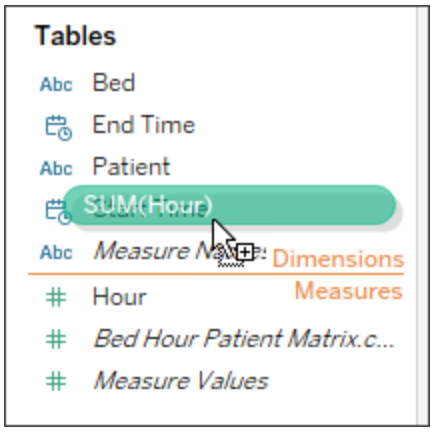
The final flow should look like this:

Analysis in Tableau Desktop

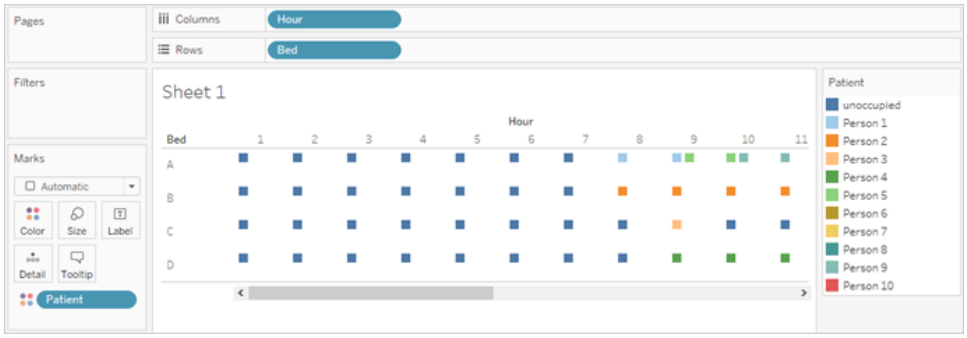
To install Tableau Desktop before continuing with this tutorial, you can download the free trial.

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** above the line separating Measures and Dimensions to make it a discrete dimension.

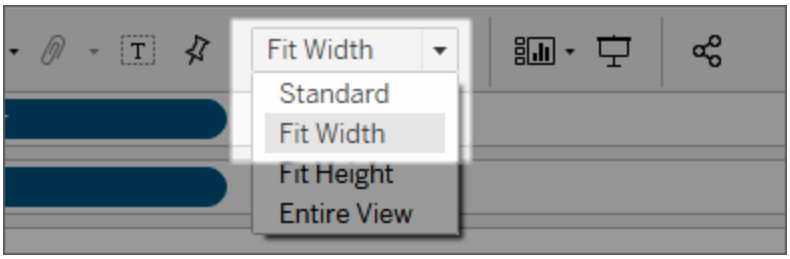


4. Drag **Bed** to the **Rows** shelf and **Hour** to the **Columns** shelf.
5. Drag **Patient** to the **Colour** shelf.

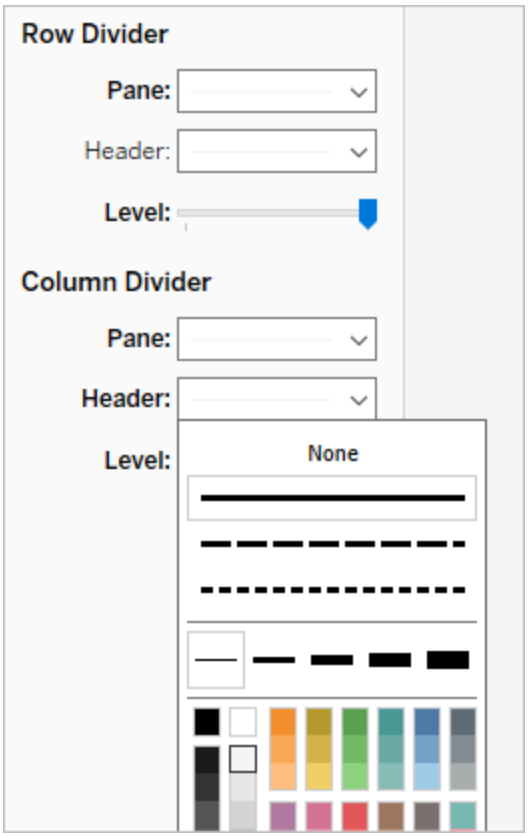


Formatting is optional, but may help make the visual more readable.

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

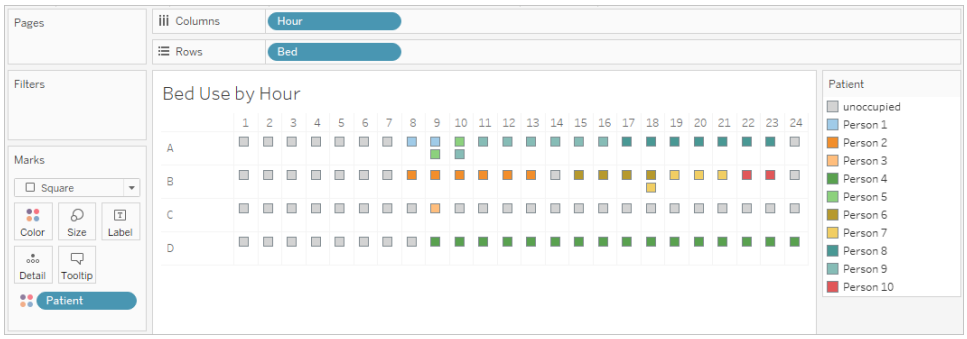


11. Click the **Format** menu and then **Borders**.
12. For **Row Divider**, click the Pane dropdown and choose a very light grey.
13. Adjust the **Level** slider to the second tick mark.
14. Repeat with the **Column Divider**. Set the **Pane** colour to be light grey and the **Level** to the second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



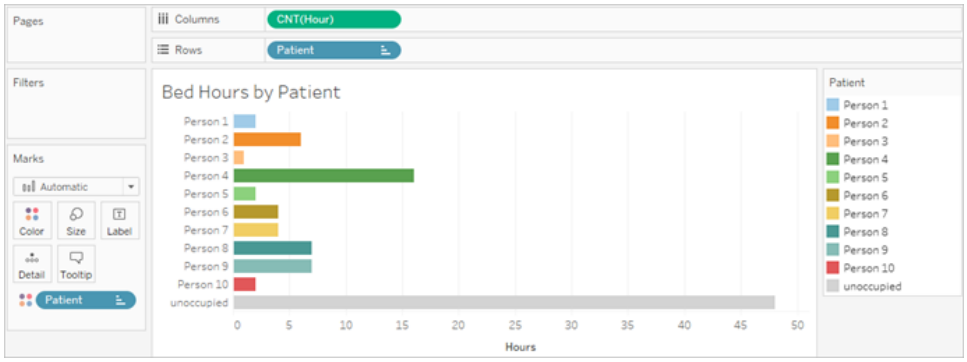
15. Double click the sheet tab at the bottom and rename it **Bed Use by Hour**.

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 the Hour pill to open the menu. Choose **Measure** > **Count**.
19. Drag another copy of the **Patient** field from the **Data** pane to the **Colour** 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 analyse, 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? 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 Builder to restructure the data for our analysis. In the second stage, **Analysis with the Second Date in Tableau Desktop** on page 511, 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 Builder (installed or via the browser) and the data downloaded. For the second portion, you'll also need Tableau Desktop installed.
>
> The data set is Traffic Violations.xlsx. It is recommended to save it in your My Tableau Prep Repository > Datasources folder.

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

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	10/22/2017	Reckless driving	Yes	800	9/8/2018	Speeding	Yes	124

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 Builder.
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.

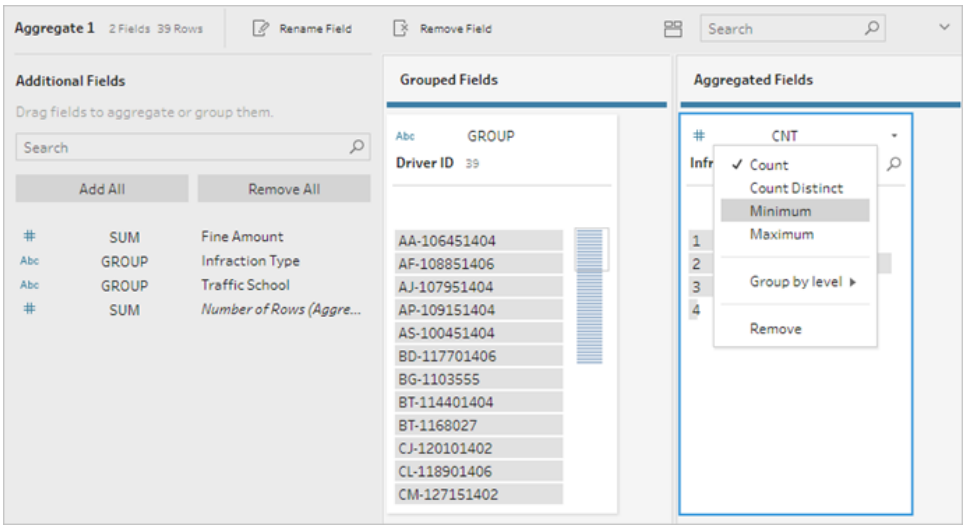
   For more information about connecting to data, see **Connect to Data** on page 75.

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**.

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 **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**.

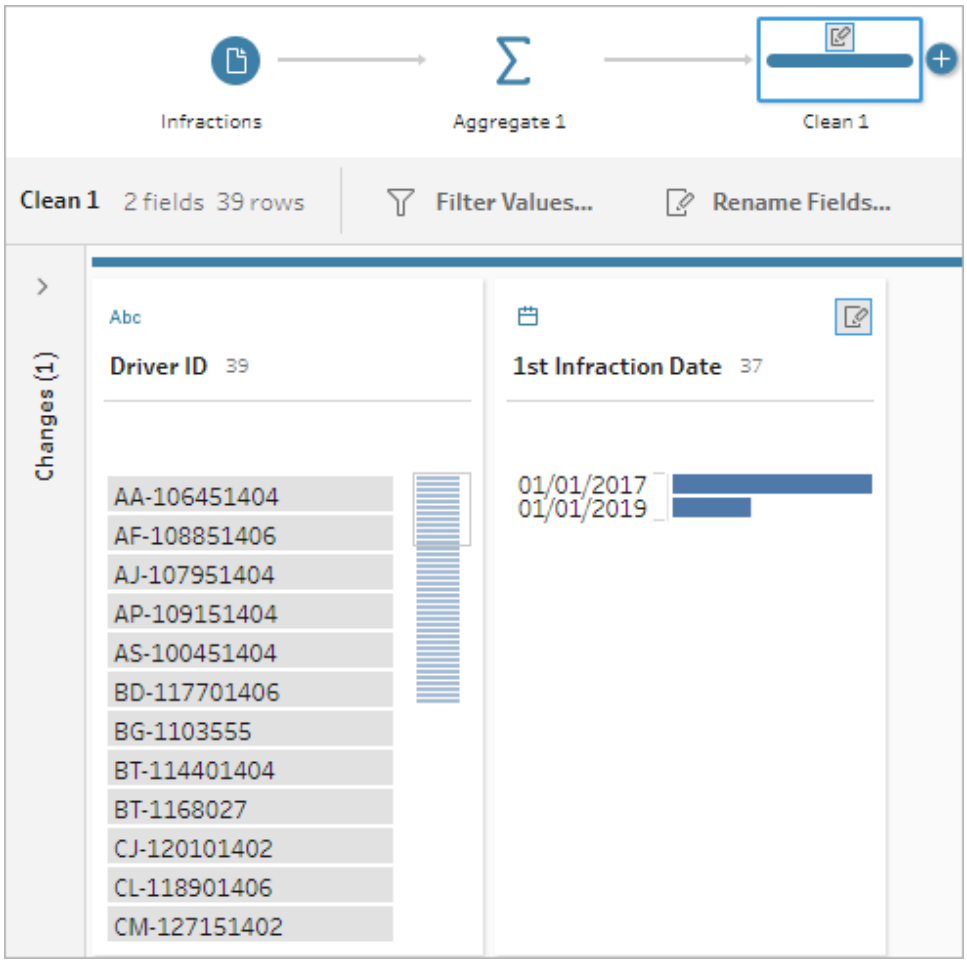


This identifies the smallest (earliest) date, which is the first infraction date per driver.

For more information about aggregations, see **Clean and Shape Data** on page 227.

8. In the Flow pane, select **Aggregate 1**, click the plus icon, and select **Clean 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 pane should look like this:

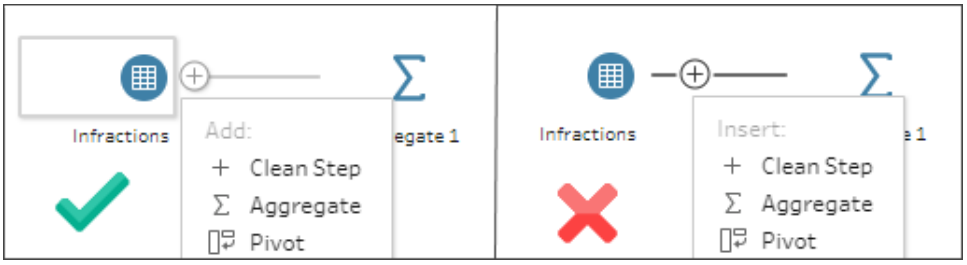


From the Profile pane in this Clean step, we can see that our data now consists of 39 rows and only 2 fields. Any field not used for grouping or aggregation is lost. But we want to be able to keep some of the original information. We could either add those fields to the grouping or aggregation (but doing 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 **Clean Step**.

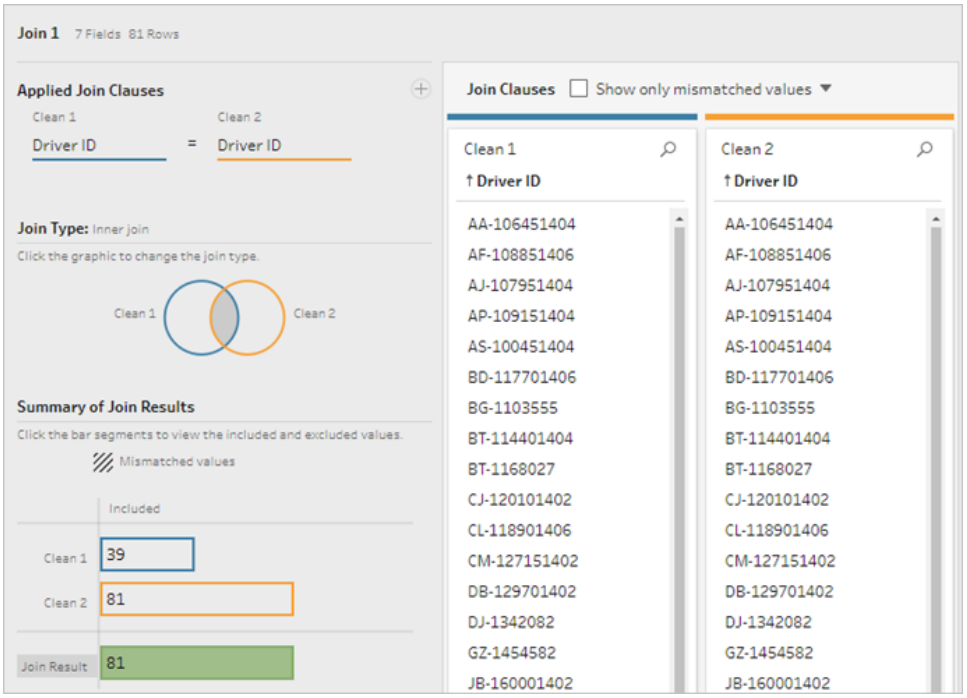
    Make sure you hover over the Infractions step directly, not the line between it and the Aggregation step. If the new Clean step is inserted between the two rather than branching,

use the Undo arrow in the tool bar and try again. The menu should say *Add* not *Insert*.



This branches your flow with 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**.

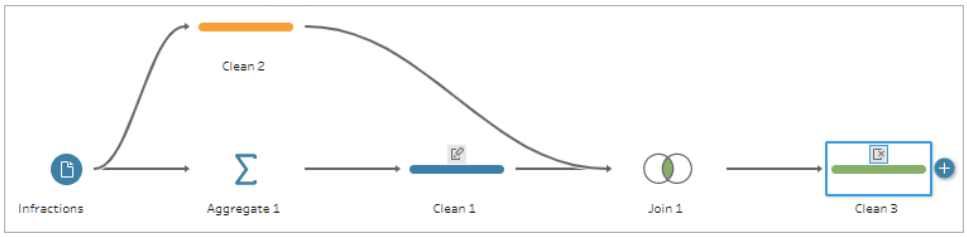


For more information about joins, see **Join your data** on page 360.

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 select **Clean Step**.
14. In the Profile pane, right-click or Ctrl -click (MacOS) the card for **Driver ID-1**, and select **Remove** .
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.

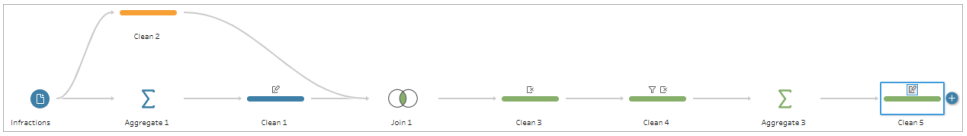
> **Note**: Because we'll want to use the data as it currently is in **Clean 3** later on in the flow, we'll add another **Clean** step to get the second infraction date. This will leave the current state of the data in Clean 3 available later on.

16. In the Flow pane, select **Clean 3**, click the plus icon, and select **Clean 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 **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 **Clean 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 **Clean 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

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

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 **Clean Step**.

    Like in step 10 above, we want to add a branch for the new clean step, not insert it between Clean 3 and Clean 4.

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.

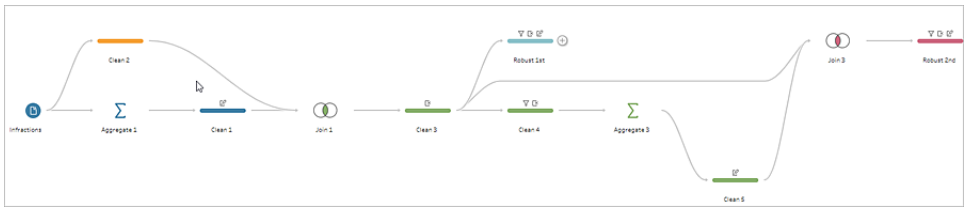
30. In the Flow pane, select **Clean 6**, after the last join.

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.

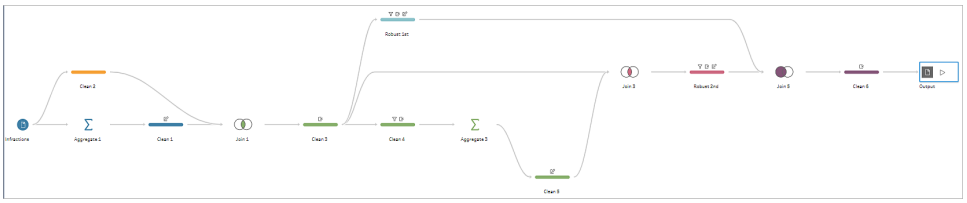
35. Select **Robust 2nd** and drag it on top of **Robust 1st**, dropping it on **Join**.
36. The default join clause should be correct as **Driver ID = Driver ID**.
37. Because we don't want to drop drivers who didn't have a second infraction, we need to make this a left join. In the **Join Type** area, 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 **Clean 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 383.

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 analysed 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 499, 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 Builder 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 Builder and optionally Tableau Desktop installed and the data downloaded.
>
> To install Tableau Prep and Tableau Desktop before continuing with this tutorial, see 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.

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.

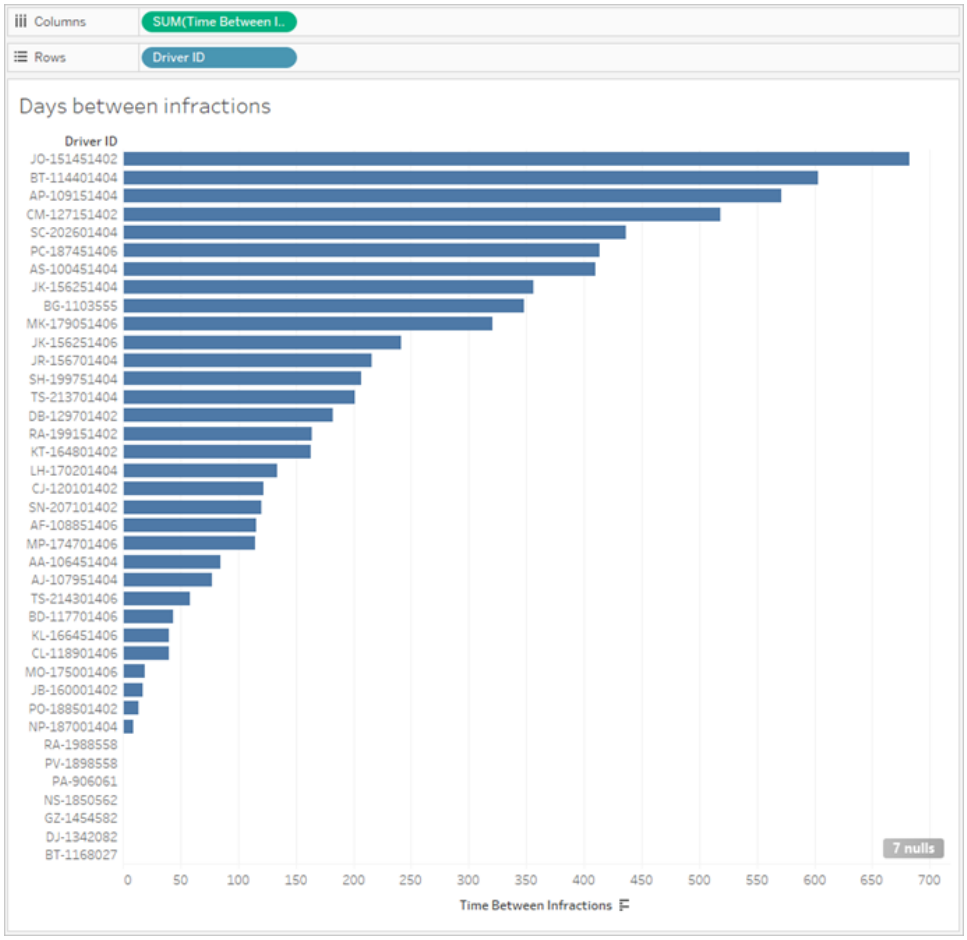
1. What was the length of time in days between the first and second infraction for each driver?

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.

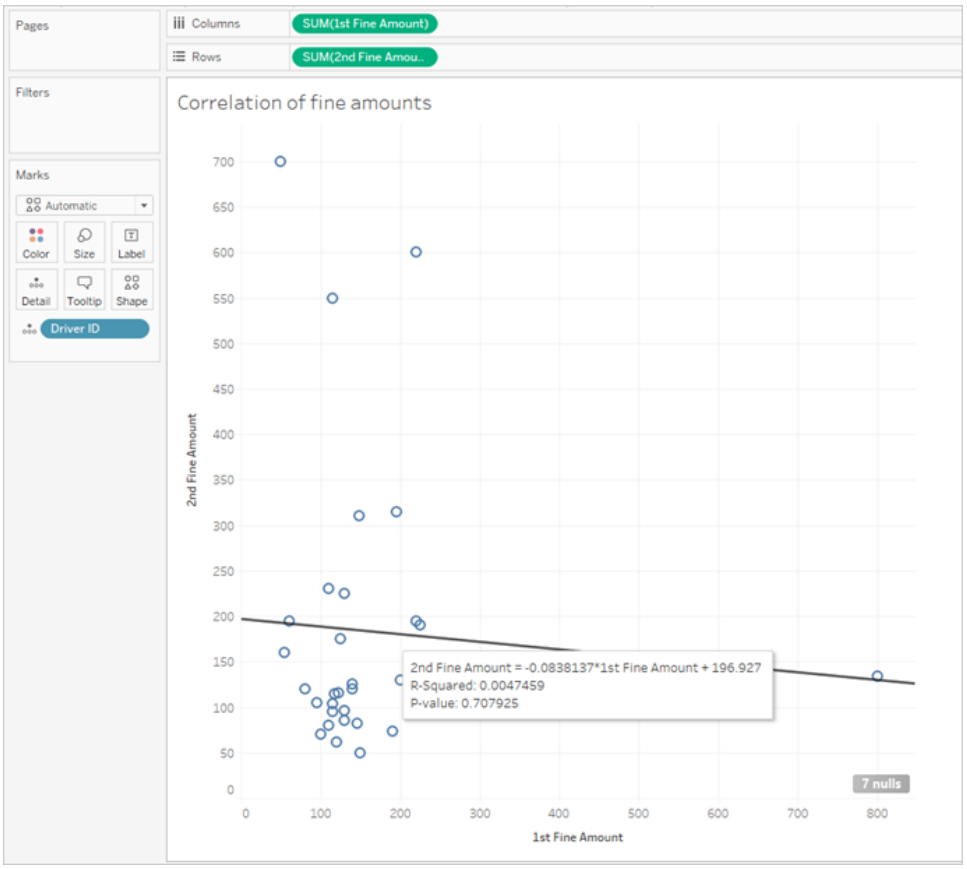


2. Compare the fine amounts for the first and second infractions. Are they correlated?

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.

   If we were to use this scatter plot in a dashboard, the trend line should be removed.



3. Which driver paid the most overall? Who paid the least?

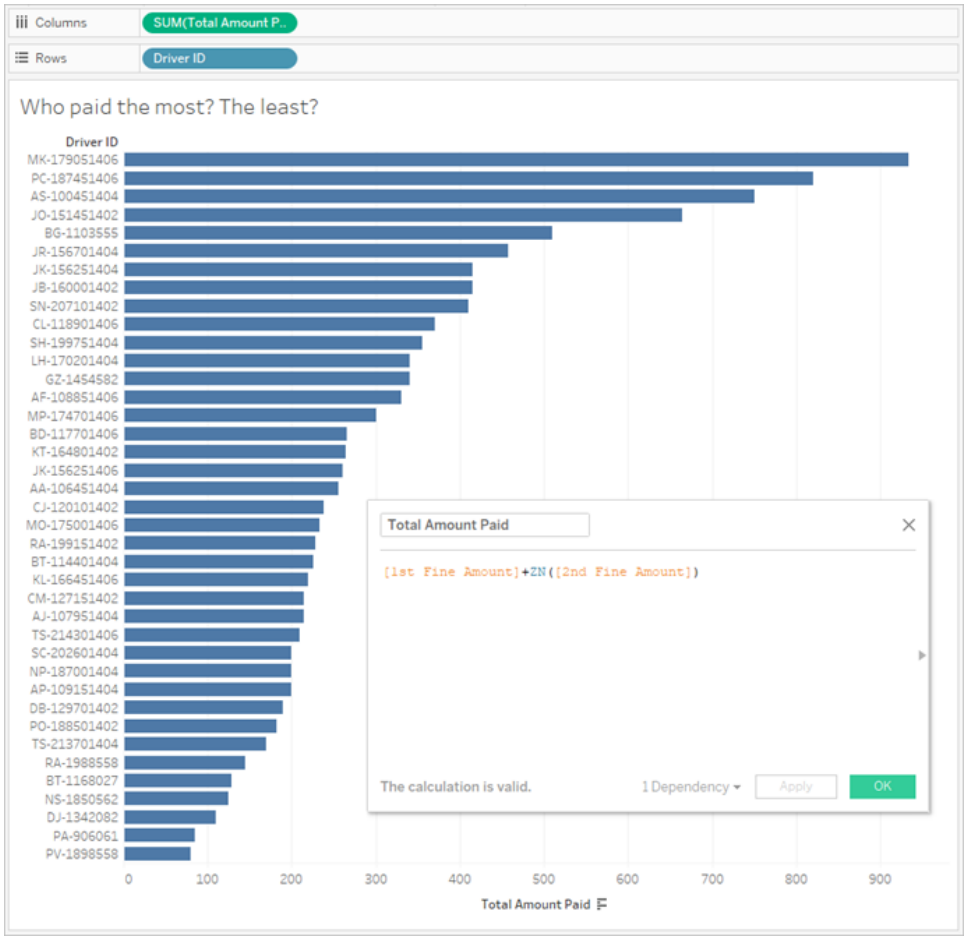
When we want to go deeper in our analysis, we may need to create some calculations.

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. Failing to do this will result in nulls if there isn't a second fine.

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?

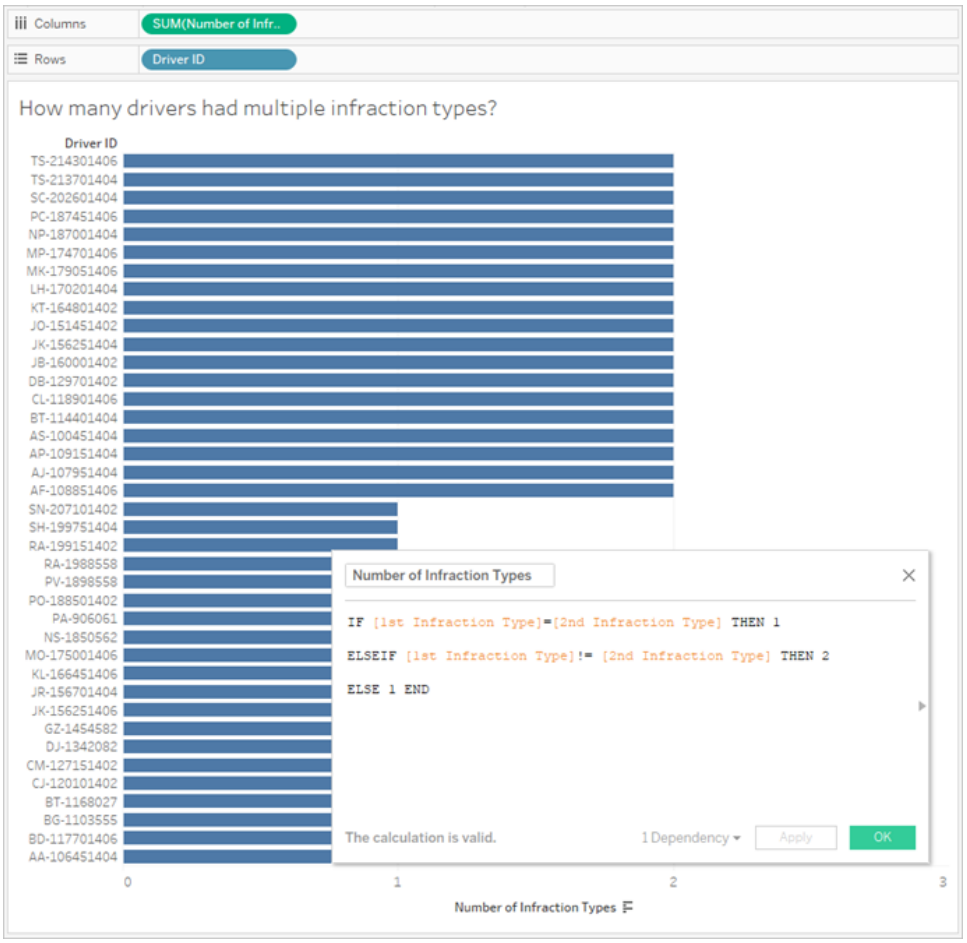
A. To answer this in Tableau Desktop, we need to do a fancier `IF` calculation, comparing if the first and second infraction types are the same. If they are, we want to assign the value "1". If they are not the same, we'll assign "2". Since we only care about multiple infraction types, any other result, 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:

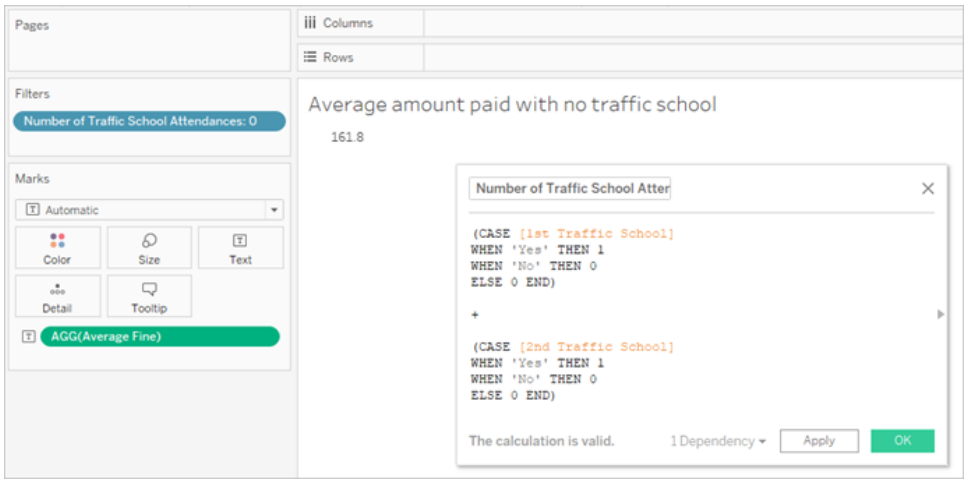
   Number of Traffic School Attendances =

```
(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 (above the line), 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 pill 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 in Tableau Prep Builder can handle this nicely. We can work from the end of the **Driver Infraction** Tableau Prep flow created in the previous tutorial **Finding the Second Date with Tableau Prep** on page 499.

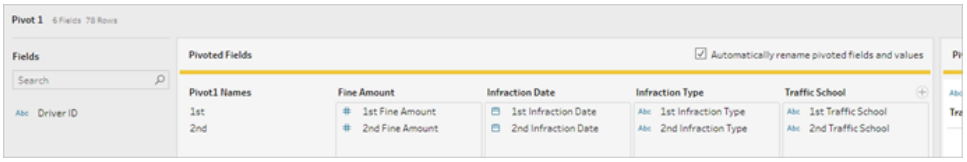
Tip: Make sure you're back in Tableau Prep for these next steps.

1. From the final clean step, add a **Pivot** step that pivots by every duplicated field. 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 **Clean and Shape Data** on page 227.

2. In the Pivoted Fields area, under the **Pivot1 Names** column, double click each value and

rename them to 1st and 2nd.



The results can be tidied by removing null dates as well as renaming and reordering fields.

3. Add a cleaning step after the pivot. In the **Infraction Date** column, right-click on the null bar and choose **Exclude**.
4. Double-click the field name **Pivot1 Names** and rename it **Infraction Number.**
5. 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

6. 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 Date = `IF [Infraction Number] = "1st" THEN [Infraction Date] END`

   2nd Infraction Date = `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 Date], [2nd Infraction Date])`
   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 Date = `{ FIXED [Driver ID] : MIN ( IF [Infraction Number] = "1st" THEN [Infraction Date] END ) }`

2nd Infraction Date = `{ FIXED [Driver ID] : MIN ( IF [Infraction Number] = "2nd" THEN [Infraction Date] END ) }`

> **Note**: The original `IF` 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).

> **Note**: These calculations can also be created in Tableau Prep Builder. For more information on LOD expressions in Prep, see **Create Level of Detail, Rank and Tile Calculations** on page 275.

3. Now we can create the `DATEDIFF` calculation as follows:

   Time Between Infractions = `DATEDIFF('day', [1st Infraction Date], [2nd Infraction Date])`

   - If we want to look at weeks or months, simply modify the date part (currently `'day'`).
   - It would also be possible to create a single calculation for the entire thing by placing the FIXED calcs inside the DATEDIFF directly:

     ```
     DATEDIFF ( 'day',
     { FIXED [Driver ID] : MIN ( IF [Infraction Number] = "1st" THEN [Infraction Date] END ) },
     { FIXED [Driver ID] : MIN ( IF [Infraction Number] = "2nd" THEN [Infraction Date] END ) }
     )
     ```

4. Plot **Time Between Infractions** on Columns and **Driver ID** on Rows.

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 ) }`

      These calculations can also be created in Tableau Prep Builder. For more information on LOD expressions in Prep, see **Create Level of Detail, Rank and Tile Calculations** on page 275.

   3. Create a scatterplot with **1st Fine Amount** on Columns and **2nd Fine Amount** on Rows 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 Date]), '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 Date]), '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 the other, we don't want either infraction to be considered here – that driver has been to traffic school and therefore doesn't fit the parameters of the question.

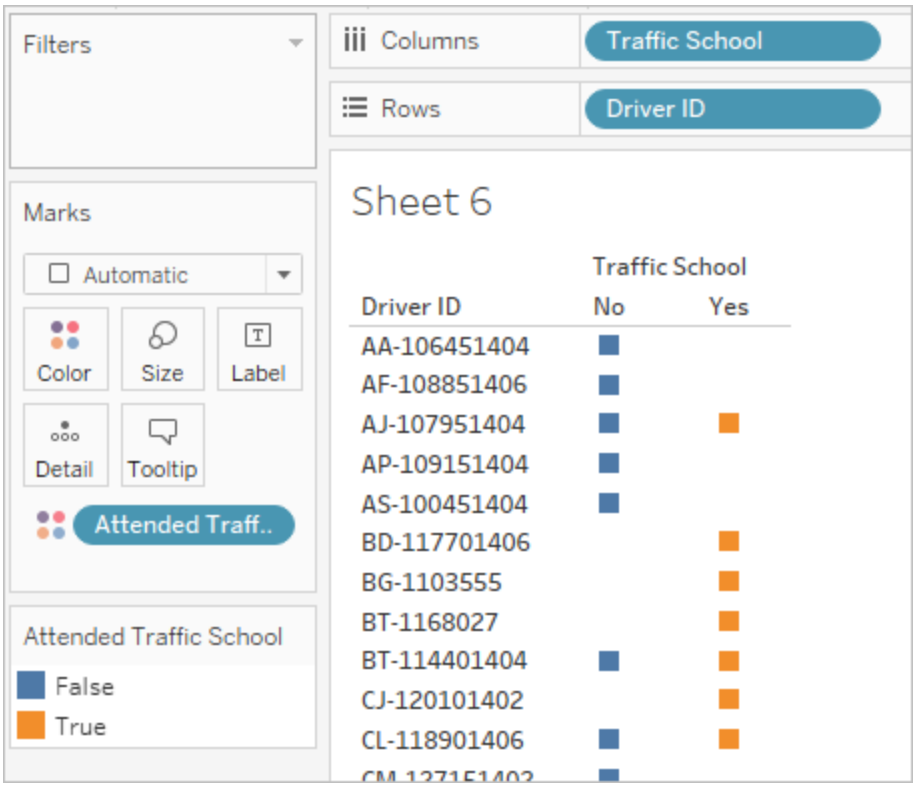
   What we want to do is filter out any driver who attended traffic school. In terms of the data, we want to filter out any driver who has a "Yes" for **Traffic School** on any row, regardless of which infraction it's associated with. Let's build our calculation in stages, using a simple view to help keep 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:

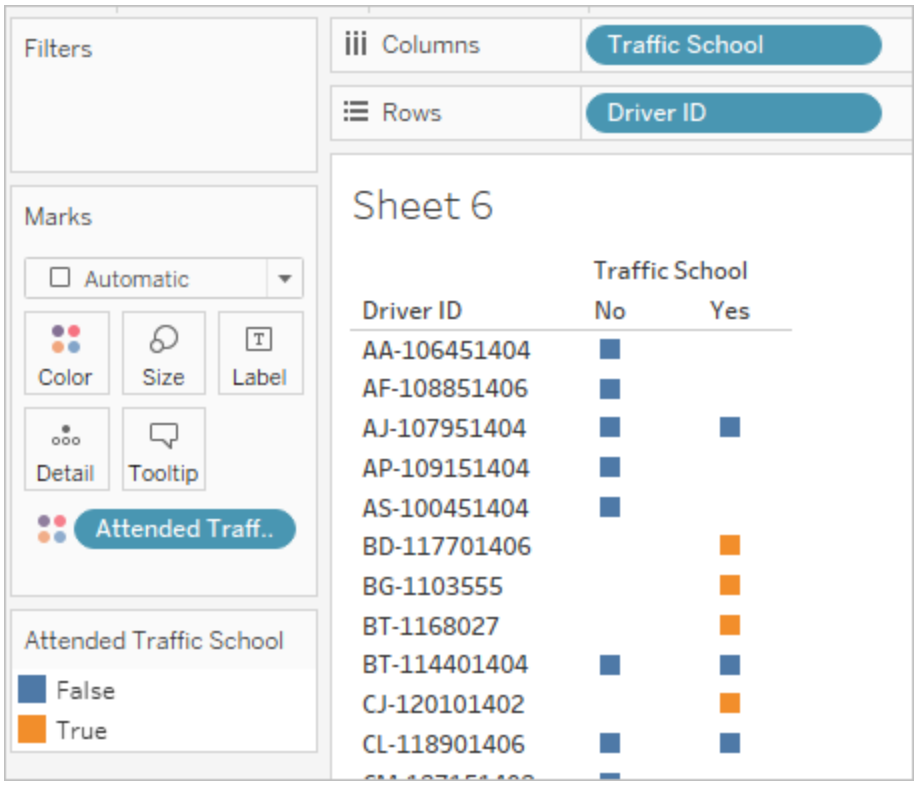
      Attended Traffic School = `CONTAINS([Traffic School), 'Yes')`

      If we bring **Attended Traffic School** to the **Colour** 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:

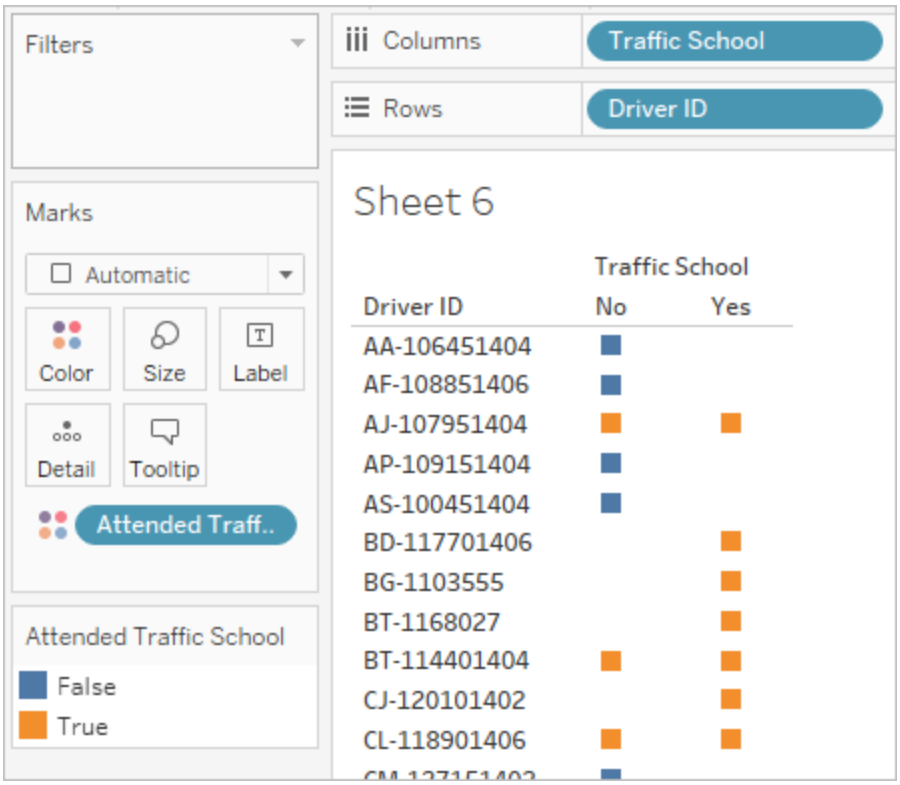
   Attended Traffic School = `{ FIXED [Driver ID] : MIN( CONTAINS ([Traffic School], 'Yes'))}`



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 Builder? 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 realise 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.)

A. 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, summarised 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 512).

- 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 519).
- 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 529).

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 Builder

This article lists problems you might encounter when using Tableau Prep Builder and suggestions for how to resolve them.

Running LogShark

LogShark is a free, open-source command line utility that you can use to extract information from Prep log files to troubleshoot and gain insight into errors and usage. Using the LogShark `Prep.twbx` plugin, you can generate workbooks with an error and flow dashboard to help you analyse and visualise Prep issues.

LogShark requires that the Prep log files that you process are compressed (zipped) files. To find the Prep log files, navigate to the **My Tableau Prep Repository** folder. The location is `/Users/<username>/Documents/My Tableau Prep Repository`.

For information about installing and running LogShark, see Get your Computer Set Up for LogShark.

Common errors when using the command line to run flows

You can run flows from the command line to refresh your output files programmatically instead of opening Tableau Prep Builder to run each flow manually. While this process helps build efficiency in your flow process, if your syntax is incorrect, or you are missing credentials for your connections or output locations, you will receive errors when running this process.

The following table describes common errors and how to resolve them. For information about how to run flows from the command line, see **Refresh flow output files from the command line** on page 425.

Error	Cause	How to fix it
"Missing arguments"	One of the required com-mand line argu-	Use “tableau-prep-cli -help” to see a list of the

	ments is missing.	arguments for the command line.
"Unable to read the connections file."	There are errors in the syntax or format in the credentials.json file for the input connections.	Check the syntax for the input connections in the .json file. For more information and examples, see **Refresh flow output files from the command line** on page 425.
"There are errors in the flow. Unable to run the flow. Check that the credentials .json file includes all required credentials. Open the flow in Tableau Prep Builder to view error details."	There are missing credentials in the credentials.json file for the input connections or the flow has errors.	Check that the .json file has the credentials for all connections, and open the flow file in Tableau Prep Builder to see if there are any errors in the flow. If the flow has errors, you must fix them and republish the flow to Tableau Server, then try running the process again.
"Could not find match for <hostname of inputConnections >"	The credentials.json file is missing an entry for the host-	Make sure the credentials.json file includes the correct

	name (server name).	credentials for the hostname (server name). For more information and examples, see **Refresh flow output files from the command line** on page 425
"We don't have credentials of all connections in tfl/tflx file. The following connection(s) were not found: <hostname of inputConnections>"	The credentials.json file is missing or has incorrect credentials for the hostname (server name) shown in the error message.	Make sure credentials.json file includes the correct credentials for the hostname (server name) listed in the error message. For more information and examples, see **Refresh flow output files from the command line** on page 425.
"Error signing in server <serverUrl> as a user <userName>. Please check the credentials."	The credentials.json file has the incorrect credentials for Tableau Server.	Make sure the credentials.json file includes all the correct credentials and elements for the output connection.

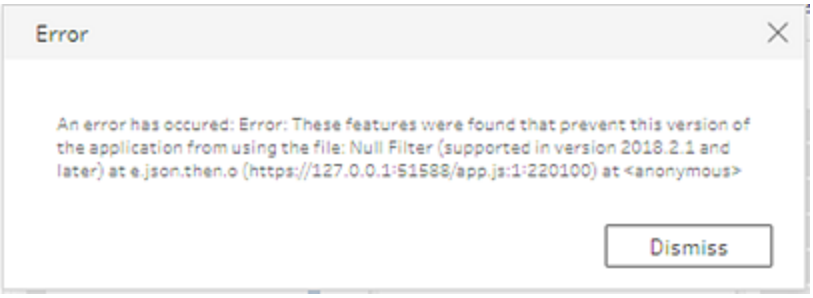
		For more information and examples, see **Refresh flow output files from the command line** on page 425
"Could not sign in successfully as <userName> to server <serverUrl>(<contentUrl>)"	The credentials.json file has the incorrect credentials for Tableau Server.	Make sure the credentials.json file includes all the correct credentials and elements for the output connection. For more information and examples, see **Refresh flow output files from the command line** on page 425
"We don't have credentials for Tableau Server to publish extract for one or more output nodes in tfl/tflx file."	The credentials.json file was not passed in as a command line argument or it is missing the credentials for the output connection.	Make sure the path to the credentials.json file is included in the command line and verify that the credentials.json file includes all the correct credentials and elements for the output

		connection. For more information and examples, see **Refresh flow output files from the command line** on page 425
"Loom rest api server not started"	The installation or environment setup is incorrect.	Make sure that Tableau Prep Builder is installed correctly and that you are running the command as an Administrator. For information about how to install Tableau Prep Builder, see Install Tableau Desktop or Tableau Prep Builder from the User Interface.
"Error. Flow file does not exist."	The path to the flow file is incor-rect.	Make sure that the correct path to the flow file is included in the command line.
"Error. Connections file does not exist."	The path to the credentials.json file is incorrect.	Make sure that the correct path to the cre-

		dentials.json file is included in the command line.
"Could not find match for <mapr01:5181>,<mapr02:5181>,<mapr03:5181>"	You must specify a specific Port ID when connecting to Apache Drill using ZooKeeper .	Include a cre-dentials.json file in the command line that specifies "port": 31010 for the input cre-dentials.

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 Builder, 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 features. 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 Builder.

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 Builder, you must have a root certificate installed on the computer where Tableau Prep Builder 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 Builder is installed. For more information, see System requirements in the Tableau Desktop and Tableau Prep Builder Deployment Guide.

Maintain Licences for Tableau Desktop and Tableau Prep

Tableau Desktop and Tableau Prep Builder can be licensed under a term licence model. When you purchase a new Tableau Server or a new Tableau Cloud subscription, however, product keys are no longer issued for Tableau Desktop or Tableau Prep Builder. Instead, you use login-based licence management to activate and sign in to Tableau Server or Tableau Cloud. For more information, see Activate Tableau using Login-based Licence Management.

Term licences must be renewed and the product key refreshed to continue providing uninterrupted service. You can continuously renew the term licence as each specified period expires. If you don't renew your term licence and the term expires, Tableau will stop working and you will no longer have access to the software. For more information about renewing your licence, see How to Renew your Tableau Licences.

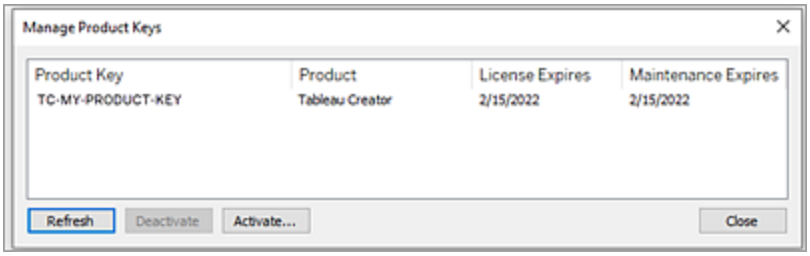
> **Note**: Trial licences for Tableau Desktop or Tableau Prep expire after a set period of time, usually 14 days. After the trial period expires, you'll need to purchase a licence to continue using the product.

View data about your licence

After you install Tableau Desktop or Tableau Prep open the application and then navigate to **Help** > **Manage Product Keys** from the top menu to see information about the type of licence you have and when it expires.

You can also activate or deactivate a product key or refresh a maintenance product key from this dialog if you are not using the Virtual Desktop (ATR) option.

> **Note**: Tableau offers term licences that provide a range of capabilities. The type of licence that you have is displayed in the **Product** field. For more information about the different type of user-based licences that are available, see User-based licences in the Tableau Server help.
>
> Existing Tableau Desktop users may have a perpetual (permanent) licence. Perpetual licences don't expire and their **Licence Expires** field in the **Manage Product Keys** dialog box displays "Permanent". However, to get access to product updates and technical support you must purchase Support and Maintenance services. These services must be renewed to continue receiving the service. Perpetual (permanent) licences are no longer available for Tableau Desktop.



Use the following buttons to take action on your product key:

- **Refresh** (Non-login-based licence management and non-Virtual Desktop only): Click the **Refresh** button to refresh a maintenance licence that is expiring, then close and restart Tableau Desktop. If the **Maintenance Expires** date doesn't update, check with your licence administrator as the key or maintenance agreement may have changed.

  A product key whose Licence Expires value is listed as "Permanent", as shown in the Manage Product Keys dialog box above, is a legacy product key. You can refresh a Permanent product key at any time as long as the maintenance end date listed in the

Tableau Customer Portal is higher than the date reflected in the Desktop Manage Product Keys dialog box.

If the product key has reached its expiry date (non-permanent product keys), you cannot refresh the product key. Visit the Tableau Customer Portal to obtain an updated subscription product key and perform a new activation. If the product key has not reached its expiration date, you can refresh the product key. When you refresh a product key that has not yet expired, only the "Licence Expires" value will change and not the product key. The product key will change when it reaches its expiration date.

To refresh a maintenance key from the command line see Refresh the product key in the Tableau Desktop and Tableau Prep Deployment guide.

> **Note:** You cannot refresh the product key if Tableau Desktop is offline. If you are activating Tableau Desktop in offline mode, you must obtain and activate a new key from the Tableau Customer Portal.

- **Deactivate** (Non-login-based licence management and non-Virtual Desktop only): Select a product key in the list then click **Deactivate** to deactivate the product key. Deactivate a product key if you need to move the product key to another computer or when you no longer need the product key on this computer.

  For more information about deactivating a product key, see Move or Deactivate Product Keys in the Tableau Desktop and Tableau Prep Deployment guide.

- **Activate**: After Tableau Desktop or Tableau Prep is installed, click **Activate** to open the activation dialog and enter your product key. If you get an error and can't activate Tableau Desktop or Tableau Prep using your product key, contact Tableau Support.

  For more information about activating a product key, see Activate and Register your Product in the Tableau Desktop and Tableau Prep Deployment guide.

Automatically refresh product keys using zero downtime licensing

Beginning in Tableau version 2021.1, internet-connected Tableau Desktop and Tableau Prep Builder users may not have to manually refresh product keys. Term licences are automatically refreshed without requiring any action starting 14 days before subscription expiration if the user is signed onto Tableau Desktop or Tableau Prep Builder. Permanent product keys are not

automatically refreshed and must be refreshed manually using the **Manage Product Keys** menu option.

Tableau Desktop and Tableau Prep Builder will attempt to silently refresh an active product key and will warn users 14 days before their licence is set to expire if the silent refresh was unsuccessful. Tableau will attempt to refresh a product key three times (at 14 days, 2 days and 1 day before licence expiry) to reflect licence end date extensions as a result of your subscription renewal. The product key is not refreshed unless a Tableau Desktop user signs onto Tableau Desktop during those times. For users who do not sign onto Tableau Desktop every day, you must refresh their product keys using the **Manage Product Keys** menu option.

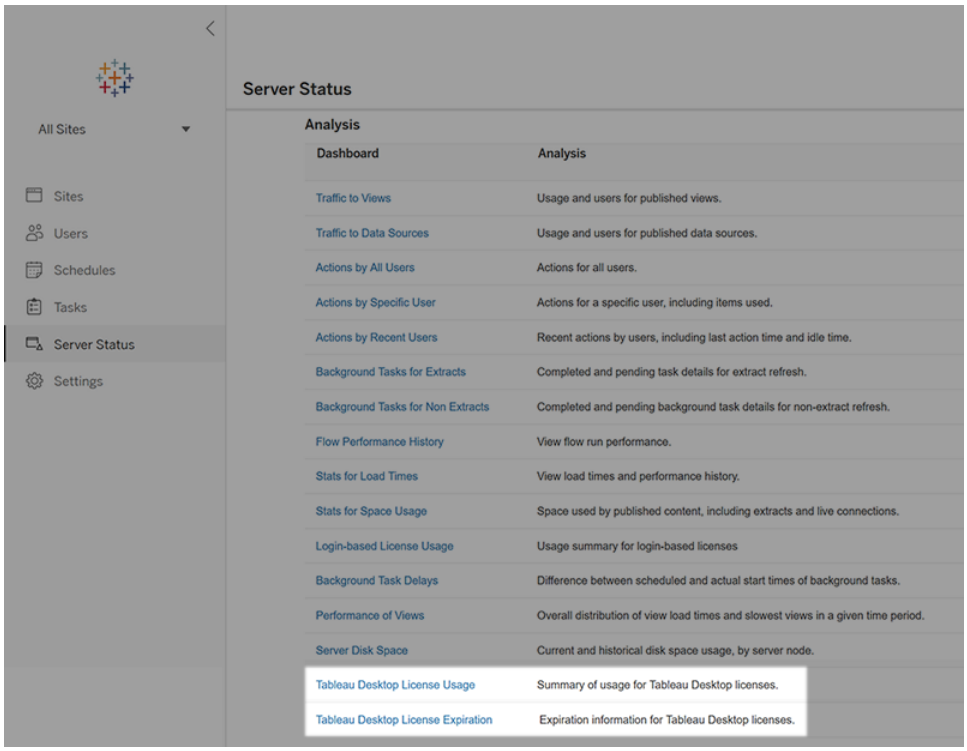
Track Tableau Desktop licence usage and expiration data

If you want to track and view licence usage and expiration data for Tableau Desktop in Tableau Server you must configure Tableau Desktop to send licence data to Tableau Server on a set interval, and then enable reporting on Tableau Server.

This enables server administrators to access two reports:

- **Desktop Licence Usage**: This report lets server administrators see usage data for Tableau Desktop licences in your organisation.
- **Desktop Licence Expiration**: This report gives server administrators information about which Tableau Desktop licences in your organisation have expired or need maintenance renewal.

If Tableau Desktop and Tableau Server are configured for licence reporting, when signed in to Tableau Server as an Administrator, you will see these two reports listed on the **Server Status** page in the **Analysis** section.



If you don't see these reports listed, then Tableau Desktop and Tableau Server may not be configured for Tableau Desktop usage reporting.

For information about how to configure Tableau Desktop and Tableau Server for usage reporting, see Manage Tableau Desktop Licence Usage in the Tableau Desktop and Tableau Prep Deployment guide.

Additional resources

For more information about managing your licence refer to the following topics:

- To find your product key and activate Tableau Desktop or Tableau Prep Builder, see Where's my product key.
- To deactivate a product key or move it to another computer, see Move or Deactivate Tableau Desktop.
- To learn more about product keys for non-persistent virtual desktops or for computers that are regularly re-imaged, see Configure Virtual Desktop Support.
- To learn more about product key management for Tableau Server or Tableau Cloud, see Licensing Overview (Linux | Windows)

- To learn more about the licence renewal process or to renew a licence, see How to Renew your Tableau Licences.

Tableau Prep Functions Reference

Tableau Prep calculation functions allow you to use calculated fields to create new data using data that already exists in your data source.

Number functions

Syntax	Description

`ABS(number)`	Returns the absolute value of the given number.
	Examples:
	`ABS(-7) = 7`
	`ABS([Budget Variance])`
	Returns the absolute value for all the numbers contained in the Budget Variance field.

`ACOS(number)`	Returns the arc cosine of the given number. The result is in radians. Example: `ACOS(-1) = 3.14159265358979`
`ASIN(number)`	Returns the arc sine of a given number. The result is in radians. Example: `ASIN(1) = 1.5707963267949`
`ATAN(number)`	Returns the arc tangent of a given number. The result is in radians. Example: `ATAN(180) = 1.5652408283942`
`ATAN2(y number, x number)`	Returns the arc tangent of two given numbers (`x` and `y`). The result is in radians.

	Example: `ATAN2(2, 1) = 1.10714871779409`
`CEILING(number)`	Rounds a number to the nearest integer of equal or greater value. Example: `CEILING(3.1415) = 4`
`COS(number)`	Returns the cosine of an angle. Specify the angle in radians. Example: `COS(PI( ) /4) = 0.707106781186548`
`COT(angle)`	Returns the cotangent of an angle. Specify the angle in radians. Example: `COT(PI( ) /4) = 1`
`DIV(integer1, integer2)`	Returns the integer part of a division operation, in which `integer1` is divided by `integer2`. Example: `DIV(11,2) = 5`
`EXP(number)`	Returns e raised to the power of the given number. Examples: `EXP(2) = 7.389 EXP(-[Growth Rate]* [Time])`
`FLOOR(number)`	Rounds a number to the nearest integer of equal or lesser value. Example: `FLOOR(3.1415) = 3`

`HEXBINX(number, number)`	Maps an x, y coordinate to the x-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately. Example: `HEXBINX([Longitude], [Latitude])`
`HEXBINY(number, number)`	Maps an x, y coordinate to the y-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately. Example: `HEXBINY([Longitude], [Latitude])`
`LN(number)`	Returns the natural logarithm of a number. Returns Null if number is less than or equal to 0. Example: `LN(EXP(5)) = 5`
`LOG(number [, base])`	Returns the logarithm of a number for the given base. If the base value is omitted, base 10 is used. Example: `LOG(256, 2) = 8`

`MAX(number, number)`	Returns the maximum of the two arguments, which must be of the same type. Returns Null if either argument is Null. `MAX` can also be applied to a single field in an aggregate calculation.
	Examples:
	`MAX(4,7)`
	`MAX(Sales,Profit)`
	`MAX([First Name],[Last Name])`

`MIN(number, number)`	Returns the minimum of the two arguments, which must be of the same type. Returns Null if either argument is Null. `MIN` can also be applied to a single field in an aggregate calculation.
	Examples:
	`MIN(4,7)`
	`MIN(Sales,Profit)`
	`MIN([First Name],[Last Name])`

`PI( )`	Returns the numeric constant pi: 3.14159. Example: `PI() = 3.14159265358979`

`POWER(number, power)`	Raises the number to the specified power.
	Examples:
	`POWER(5,2) = 52 = 25 POWER (Temperature, 2)`
	You can also use the ^ symbol:
	`5^2 = POWER(5,2) = 25`

`RADIANS(number)`	Converts the given number from degrees to radians. Example: `RADIANS(180) = 3.14159`
`ROUND(number, [decimals])`	Rounds numbers to a specified number of digits. The decimals argument specifies how many decimal points of precision to include in the final result. If decimals is omitted, number is rounded to the nearest integer. Example:

	This example rounds every Sales value to an integer: `ROUND(Sales)`

`SIGN(number)`	Returns the sign of a number: The possible return values are -1 if the number is negative, 0 if the number is zero or 1 if the number is positive.
	Example:
	If the average of the profit field is negative, then
	`SIGN(AVG(Profit)) = -1`

`SIN(number)`	Returns the sine of an angle. Specify the angle in radians.
	Examples:
	`SIN(0) = 1.0`
	`SIN(PI( )/4) = 0.707106781186548`

`SQRT(number)`	Returns the square root of a number. Example: `SQRT(25) = 5`
`SQUARE(number)`	Returns the square of a number. Example: `SQUARE(5) = 25`
`TAN(number)`	Returns the tangent of an angle. Specify the angle in radians. Example: `TAN(PI ( )/4) = 1.0`
`ZN(expression)`	Returns the expression if it is not null, otherwise returns zero. Use this function to use zero values instead of null values.

	Example: `ZN([Profit]) = [Profit]`

String functions

Syntax	Description
`ASCII(string)`	Returns the ASCII code for the first character of string. Example: `ASCII('A') = 65`
`CHAR(number)`	Returns the character encoded by the ASCII code number. Example: `CHAR(65) = 'A'`
`CONTAINS(string, substring)`	Returns true if the given string contains the specified substring. Example: `CONTAINS("Calculation", "alcu") = true`
`ENDSWITH(string, substring)`	Returns true if the given string ends with the specified `substring`. Trailing white spaces are ignored. Example: `ENDSWITH("Tableau", "leau") = true`
`FIND(string, substring, [start])`	Returns the index position of `substring` in string, or 0 if the substring isn't found. If the optional argument start is added, the function ignores any instances of `substring` that appear before the index position start. The first

	character in the string is position 1. Examples: `FIND("Calculation", "alcu") = 2`
`FINDNTH(string, substring, occurrence)`	Returns the position of the nth occurrence of substring within the specified string, where n is defined by the occurrence argument. Example: `FINDNTH("Calculation", "a", 2) =7`
`ISDATE(string)`	Returns true if a given string is a valid date. Example: `ISDATE("2004-04-15") = True`
`LEFT(string, number)`	Returns the left-most number of characters in the string. Example: `LEFT("Matador", 4) = "Mata"`
`LEN(string)`	Returns the length of the string. Example: `LEN("Matador") = 7`
`LOWER(string)`	Returns string, with all characters lower case. Example: `LOWER("ProductVersion") = "productversion"`
`LTRIM(string)`	Returns the string with any leading spaces removed. Example: `Example: LTRIM(" Sales") = "Sales"`

`(MID(string, start, [length])`	Returns the string starting at index position start. The first character in the string is position 1. If the optional argument length is added, the returned string includes only that number of characters.
	Examples:
	`MID("Calculation", 2) = "alculation"`
	`MID("Calculation", 2, 5) ="alcul"`

`PROPER(string)`	Converts a text string so the first letter of each word is capitalised and the remaining letters are lowercase. Spaces and non-alphanumeric characters such as punctuation also act as separators. Example: `PROPER("PRODUCT name") = "Product Name"`
`REGEXP_EXTRACT(string, pattern)`	Returns a substring of the given string that matches the capturing group within the regular expression pattern. The regular expression pattern requires exactly one capturing group Example: `REGEXP_EXTRACT('abc 123', '[a-z]+\\s+(\\d+)') = '123'`
`REGEXP_EXTRACT_NTH(string, pattern, index)`	Returns a substring of the given string using the regular expression pattern. The substring is matched to the nth capturing group, where n is the given index. Example: `REGEXP_EXTRACT_NTH('abc 123', '([a-z]+)\\s+(\\d+)', 2) = '123'`

`REGEXP_MATCH(string, pattern)`	Returns true if a substring of the provided string matches the regular expression pattern. Example: `REGEXP_MATCH('-([1234].[ The.Market ])-', '\\[\\s*(\\w*\\.)(\\w*\\s*\\])') = true`
`REGEXP_REPLACE(string, pattern, replacement)`	Returns a copy of the given string where the matching pattern is substituted with the replacement string. Example: `REGEXP_REPLACE('abc 123', '\\s', '-') = 'abc--123'`
`REPLACE(string, substring, replacement)`	Searches string for substring and replaces it with replacement. If substring is not found, the string is not changed. Example: `REPLACE("Calculation", "ion", "ed") = "Calculated"`
`RIGHT(string, number)`	Returns the right-most number of characters in string. Example: `RIGHT("Calculation", 4) = "tion"`
`RTRIM(string)`	Returns string with any trailing spaces removed. Example: `Example: RTRIM("Market ") = "Market"`
`SPACE(number)`	Returns a string that is composed of the specified number of repeated spaces.

	Example: `SPACE(1) = " "`
`SPLIT(string, delimiter, token number)`	Returns a substring from a string, as determined by a delimiter extracting the characters from the beginning or end of the string. Example: `SPLIT('a-b-c-d', '-', 2) = 'b'`
`STARTSWITH(string, substring)`	Returns true if string starts with `substring`. Leading white spaces are ignored. Example: `STARTSWITH(“Joker”, “Jo”) = true`
`TRIM(string)`	Returns the string with leading and trailing spaces removed. Example: `TRIM(" Calculation ") = "Calculation"`
`UPPER(string)`	Returns string, with all characters uppercase. Example: `UPPER("Calculation") = "CALCULATION"`

Aggregate functions

Syntax	Description
`AVG(expression)`	Returns the average of all the values in the expression. AVG can be used with numeric fields only. Null values are ignored Example;

	`AVG([Profit])`
`COUNT(expression)`	Returns the number of items in a group. Null values are not counted. Example: `COUNT([Customer ID])`
`COUNTD(expression)`	Returns the number of distinct items in a group. NULL values are not counted. Each unique value is counted only once. Example: `COUNTD([Region])`
`MEDIAN(expression)`	Returns the median of a single expression. MEDIAN can be used with numeric fields only. Null values are ignored. Example: `MEDIAN([Profit])`
`PERCENTILE(expression, number)`	Aggregate calculation that returns the percentile value from the given expression corresponding to the specified number. Valid values for the number are 0 to 1. PERCENTILE ([expression], 0.50) will always return the median value in the expression. Example: `PERCENTILE([Sales], 0.90)`
`STDEV(expression)`	Returns the sample standard deviation of the expression. Example: `STDEV([Profit])`
`STDEVP(expression)`	Returns the population standard deviation of

	the expression. Example: `STDEVP([Profit])`
`SUM(expression)`	Returns the sum of all values in the expression. SUM can be used with numeric fields only. Null values are ignored. Example: `SUM([Profit])`
`VAR(expression)`	Returns the statistical variance of all values in the given expression based on a sample of the population. Example: `VAR([Profit])`
`VARP(expression)`	Returns the statistical variance of all values in the given expression based on a sample of the population. Example: `VARP([Profit])`
`ZN(expression)`	Returns the expression if it is not null, otherwise returns zero. Use this function to use zero values instead of null values. Example: `ZN([Profit]) = [Profit]`

Type conversion functions

Syntax	Description
`DATE`	Returns a date, given a number, string or date expression.

	Example: `DATE("2006-06-15 14:52") = 2006-06-15` Note that the quotation marks are required.
`DATETIME(expression)`	Returns a datetime given a number, string or date expression. Example: `DATETIME("April 15, 2004 07:59:00")` Note that the quotation marks are required.
`FLOAT(expression)`	Returns a float given an expression of any type. This function requires unformatted numbers, which means excluding commas and other symbols. Example: `FLOAT("3")= 3.000`
`INT(expression)`	Returns an integer given an expression. This function truncates results to the closest integer towards zero. Example: `INT(8.0/3.0) = 2 or INT(-9.7) = -9`
`MAKEDATE(year, month, day)`	Returns a date value constructed from a year, a month and a day of the month. Example: `MAKEDATE(2014, 3, 18)`
`MAKEDATETIME(date, time)`	Returns a date and time value given a date expression and a time expression. Example:

	`MAKEDATETIME(#2012-11-12#, #07:59:00#)`
`MAKETIME(hour, minute, second)`	Returns a time value constructed from hours, minutes and seconds. Example: `MAKETIME(14, 52, 40)`
`STR(expression)`	Returns a string given an expression. Example: `STR([Age])` Returns all the values of the Age measure as strings.

Date functions

Syntax	Description
`DATEADD(date_part, interval, date)`	Adds an increment to the specified date and returns the new date. The increment is defined by the interval and the date_part. Example: `DATEADD('month', 3, #2004-04-15#) = 2004-07-15 12:00:00 AM`
`DATEDIFF(date_part, start_date, end_date, [start_of_week])`	Returns the difference between two dates where start_date is subtracted from end_date. The difference is expressed in units of date_part. If start_of_week is omitted, the week start day is determined by the start day configured for the data source. Example: `DATEDIFF('month', #2004-07-15#, #2004-04-03#, 'sunday') = -3`

`DATENAME(date_part, date, [start_of_week])`	Returns a part of the given date as a string, where the part is defined by date_part. If start_of_week is omitted, the week start day is determined by the start day configured for the data source. Example: `DATENAME('month', #2004-04-15#) = "April"`
`DATEPARSE(format, string)`	Converts a string to a date in the specified format. Example: `DATEPARSE ("dd.MMMM.yyyy", "15.April.2004") = 2004-04-15 12:00:00 AM`
`DATEPARTT(date_part, date, [start_of_week])`	Returns a part of the given date as an integer where the part is defined by date_part. If start_of_week is omitted, the week start day is determined by the start day configured for the data source. Example: `DATEPART('month', #2004-04-15#) = 4`
`DATETRUNC(date_part, date, [start_of_week])`	Truncates the specified date to the accuracy specified by the date_part and returns the new date. If start_of_week is omitted, the week start day is determined by the start day configured for the data source. Example: `DATETRUNC('quarter', #2004-08-15#) = 2004-07-01 12:00:00 AM`

`DAY(date)`	Returns the day of the given date as an integer. Example: `DAY(#2004-04-12#) = 12`
`ISDATE(string)`	Returns true if a given string is a valid date. Example: `ISDATE("2004-04-15") = True`
`MONTH(date)`	Returns the month of a given date as an integer. Example: `MONTH(#2004-04-12#) = 4`
`NOW()`	Returns the current date and time. Example: `NOW() = 2004-05-12 1:08:21 PM`
`TODAY()`	Returns the current date. Example: `TODAY() = 2004-05-12`
`YEAR(date)`	Returns the year of a given date as an integer. Example: `YEAR(#2004-04-12#) = 2004`

Logical functions

Syntax	Description
`IFNULL(expr1, expr2)`	Returns `<expr1>` if it is not null, otherwise returns `<expr2>`. Example: `IFNULL([Profit], 0)`

`IIF(test, then, else, [unknown])`	Checks whether a condition is met, and returns one value if TRUE, another value if FALSE and an optional third value or NULL if unknown. Example: `IIF([Profit] > 0, 'Profit', 'Loss')`
`ISDATE(string)`	Returns true if a given string is a valid date. Example: `ISDATE("2004-04-15") = True`
`ISNULL(expression)`	Returns true if the expression does not contain valid data (`Null`). Example: `ISNULL([Profit])`
`ZN(expression)`	Returns the expression if it is not null, otherwise returns zero. Use this function to use zero values instead of null values. Example: `ZN([Profit]) = [Profit]`

Analytic functions

Syntax	Description
`LAST_VALUE(expression, [return_last_non-null_value])`	Returns the last value of the given expression, from the first row in the partition to the current row. Accepts an optional second boolean parameter to return the last non-null value. Example: `{ ORDERBY [Row ID] ASC: LAST_VALUE ([Category], true) } = returns the`

	`last non-null value of [Category]`
`LOOKUP(expression, [offset])`	Returns the value of the given expression in a target row, specified as a relative offset from the current row. Returns NULL if the target row cannot be determined. The [`offset`] parameter of the `LOOKUP` function can only take a literal integer such as 1. Example: `{ORDERBY [DATE] ASC : LOOKUP ([SALES], -1)}` Returns the SALES value from the previous row.
`NTILE(number)`	Distributes rows in a selected partition into a specified number of groups or tiles. With this function, the set of values (6,9,9,14) distributed to 3 tiles would be assigned to the tiles (1,2,2,3) in ascending order. Example: `{PARTITION [Customer]: {ORDERBY [Sales] ASC: NTILE(3) }}`
`RANK()`	Returns the standard competition rank for the current row in the partition. Identical values are assigned an identical rank. Example: `{PARTITION [Customer]: {ORDERBY [Sales] ASC: RANK() }}`
`RANK_DENSE()`	Returns the dense rank for the current row in the partition. Identical values are assigned an identical rank, but no gaps are inserted into the number sequence. Example:

	`{PARTITION [Customer]: {ORDERBY [Sales] ASC: RANK_DENSE() }}`
`RANK_MODIFIED()`	Returns the modified competition rank for the current row in the partition. Identical values are assigned an identical rank. Example: `{PARTITION [Customer]: {ORDERBY [Sales] ASC: RANK_MODIFIED() }}`
`RANK_PERCENTILE()`	Returns the percentile rank for the current row in the partition. Example: `{PARTITION [Customer]: {ORDERBY [Sales] ASC: RANK_PERCENTILE() }}`
`ROW_NUMBER()`	Returns a unique sequential row ID to each row. Example: `{PARTITION [Customer]: {ORDERBY [Sales] ASC: ROW_NUMBER() }}`
`RUNNING_AVG(expression)`	Returns the running average of the given expression, from the first row in the partition to the current row. Example: `{ PARTITION [Category] : { ORDERBY [Row ID] ASC : RUNNING_AVG([Sales]) } } = running average of Sales, per Category.`
`RUNNING_SUM(expression)`	Returns the running sum of the given expression, from the first row in the partition to the current row. Example:

	`{ ORDERBY [Row ID] ASC : RUNNING_SUM([Sales]) } = running sum of all Sales`

Keywords

Function	Description
`AND`	Performs a logical conjunction on two expressions Example: `IF (ATTR([Market]) = "New Business" AND SUM([Sales]) > [Emerging Threshold] )THEN "Well Performing"`
`ASC`	Defines the ascending order for `ORDERBY` operations. Example `{ORDERBY [Sales] ASC:RANK()}`
`CASE`	Finds the first <`value`> that matches `<expr>` and returns the corresponding <`return`>. Example: `CASE [RomanNumeral] WHEN "|" THEN 1 WHEN "II" THEN 2 ELSE 3 END`
`DESC`	Defines the descending order for `ORDERBY` operations. Example: `{ORDERBY [Sales] DESC:RANK()}`
`ELSE`	Test a series of expressions returning the <`then`> value for the first true<`expr`>.

	Example: `IF [Profit] > 0 THEN 'Profitable' ELSE 'Loss' END`
`ELSEIF`	Test a series of expressions returning the <`then`> value for the first true<`expr`>. Equivalent to putting an `if` statement inside an ELSE operation. Example: `IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END`
`END`	Terminates an operation introduced by the corresponding directive. Example: `IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END`
`{ FIXED }`	FIXED level of detail expressions compute values using the specified dimensions without reference to the view level of detail. Example: `{FIXED[Customer ID]:MIN([Order Date])}`
`IF`	Creates conditional statement (`IF` statement) and allows you to execute only if a condition is True. Example: `IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END`

`NOT`	Performs a logical negation on an expression. Example: `IF NOT [Profit] > 0 THEN "Unprofitable" END`
`OR`	Performs a logical disjunction on an expression. Example: `IF [Profit] < 0 OR [Profit] = 0 THEN "Needs Improvement" END`
`{ ORDERBY }`	Defines the order to apply analytic functions. Use the optional asc | desc argument to specify ascending or descending order for each field. Example: `{ORDERBY [Customer Name] ASC, [Sales] DESC: RANK() }`
`{ PARTITION }`	Defines the group to analytic functions. ORDERBY must be inside the partition. Example: `{PARTITION [Customer]: {ORDERBY [Sales]: RANK()}}`
`THEN`	Test a series of expressions returning the <`then`> value for the first true<`expr`>. Example: `IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Break even' ELSE 'unprofitable' END`
`WHEN`	Finds the first <`value`> that matches <`expr`> and returns the corresponding <`return`>. Example:

	`CASE [RomanNumberal] WHEN 'I' THEN 1 WHEN 'II' THEN 2 ELSE 3 END`