USEREADY

A PRACTITIONER'S GUIDE TO

TABLEAU PREP BUILDER

2019.1

Published by
USEReady
1407 Broadway, Suite 721, New York, NY 10018
www.useready.com

ISBN: 978-1-64516-475-3

Manufactured in the United States of America

Table of Contents

About USEReady

USEReady's mission is to help users succeed with data.

We achieve this mission with fanatical customer centricity, humility, and integrity. At USEReady, we love to solve customer problems, contribute to community and continue to improve. We aim to build a strong learning culture and have fun doing it together.

We have harnessed modern Business Intelligence solutions with cutting edge Artificial Intelligence (AI) platforms to unlock the power of data. Our result-oriented solutions ensure that our customers are successful in adopting self-service technologies.

We have partnered with the best in class BI, Data and Cloud products such as Tableau, Snowflake, Alteryx, Informatica, Alation, Collibra, AWS and Microsoft. We have delivered proven success across Financial Services, Insurance, Retail and Media verticals.

We have been nominated and won several awards along this journey. Check us out at www.useready.com

About Tableau

Tableau Software, headquartered in Seattle, Washington, is a leader in data visualization and data analytics solutions. It was founded in 2003 to create software products focused on Business Intelligence with a revolutionary approach.

The company was recognized as the leader of the Gartner Quadrant for five consecutive years from 2012 to 2017 due to its remarkable Ability to Execute and Completeness of Vision.

Hundreds of thousands of data analysts around the world rely on Tableau Software products for their daily data preparation, data visualization, and data exploration needs. Tableau is also recognized as one of the top products in delivering powerful end-user interactivity supported by self-service capabilities.

In simple terms, Tableau Software helps people see and understand data.

For more details, please visit https://www.tableau.com.

Letter from the CEO

Tableau is an important BI partner and the key component to USEReady's goto market strategy. Tableau pioneered self-service BI and as BI is going self-driven, Tableau is emerging into a platform. I meet Francois Ajenstat, CPO, Tableau at least four times a year. During one of our meetings, I asked Francois "Why ETL product from Tableau? It is such a crowded space".

He said "Well, Tableau has over 80K customers and less than 10% are using a self-service ETL product. This means we have an unmet need for the rest of the 90%". I felt that is a pretty accurate view. Most Tableau creators rely on either IT to assist them with data prep or message data using a spreadsheet. Neither approach is ideal for rapid fire BI. With this new product, Tableau Prep Builder, Tableau customers have a tool that assists them with data shaping efforts.

At USEReady, we believe in helping organizations succeed with data. We are very proud of the fact that we carry this mission with a sense of customer centricity, community, integrity and humility. To serve the community we are part of, our consultants conduct several doctor sessions, webinars, author content, write blogs etc. Along with that strategy we have authored this book "A Practitioner's Guide to Tableau Prep Builder". This is the second book in our "Practitioner's Guide" series.

A total of ten practitioners have contributed their expertise to create this book, incorporating illustrations and hands-on exercises.

Finally, I am thankful to Vijai Narasimha, USEReady to lead the book initiative, Jeremy Walsh, Strategic Customer Success Manager, Tableau Software and Jeff Black, Customer Consulting Manager, Financial Services Tableau Software for reviewing and writing the foreword to this book.

I am quite confident that Tableau community is going to embrace Prep with the same love they bestowed to Tableau Desktop. You are welcome to share me your feedback or thoughts at udayh@useready.com

Uday Hegde
CEO
USEReady

Foreword

As data exponentially grows, continues to be created, and is stored in more places, the challenge of preparing data is only becoming more difficult. Often in the world of data analytics, 80% of a person's time is spent getting the data ready, leaving only 20% of time for analysis. Considering this ratio, if you have one hour to obtain data, prepare it, and answer any necessary questions, approximately 48 minutes of that time will be spent obtaining and preparing data. Merely 12 minutes are left to perform the value-add analysis. This ratio is a problem that needs balance.

Having worked with hundreds of enterprises and spending years in the financial services industry, this is one of the biggest challenges I see and have also experienced firsthand. As a former customer, bringing data together to answer questions about the business was a considerable challenge; many individuals do not possess the proper solution or skills to effectively resolve this challenge. At Tableau, I have spent a significant amount of time testing Tableau Prep Builder during its early alpha and beta stages. Since its release, I am now seeing customers balance this prep-to-analysis ratio with the product.

For those wanting to learn Tableau Prep Builder, this practitioner's guide—written by certified BI consultants—will build the foundation needed, whether you're an individual in business, a journalist who reports data findings, or a student at an educational institution.

Given their experience as a certified training partner and collaborating with enterprises on strategic data projects, USEReady is positioned to help you succeed with data.

The aim of this book is to help you learn Tableau Prep Builder, and in doing so, you will be better equipped to solve data challenges.

Jeff Black
Customer Consulting Manager, Financial Services
Tableau Software

Foreword

I have spent nearly a decade helping Tableau customers achieve self-sufficient analytics, most recently as a Strategic Customer Success Manager working closely with Tableau's largest customers on success planning, driving adoption, and enabling users on Tableau's Enterprise Platform. Now more than ever, there is a demand to empower users to clean, combine, aggregate, and prepare data for analysis. USEReady is a leader in this discipline and has once again created a training manual not only to help users adopt Tableau Prep Builder quickly and easily but to further Tableau's mission: Help People See and Understand Data.

USEReady has created a book designed for those who are just beginning their journey on data cleansing and preparation. Tableau Prep Builder is a new tool in the Tableau Product suite designed to make data prep easy and intuitive. By the time you finish working the examples and real world use cases in this easy-to-follow training aid, you will know how to connect to a variety of files or servers, combine data from multiple data sources, easily bring tables into a "flow pane" and perform operations such as validate, filter, pivot, union, and join to clean and prepare, or shape, your data.

Tableau disrupted the BI space in 2003 with the introduction of Tableau Desktop with its "drag and drop" concept that thousands of customers have come to know and love today. With Tableau Prep Builder, Tableau has again found a way to disrupt the BI space, helping those thousands of customers continue their data journey through data preparation.

USEReady fully appreciates Tableau's mission and for many years has helped customers achieve self-service analytics through faster insight, and faster time-to-market with Tableau's Enterprise solution, which now includes Tableau Prep Builder.

Jeremy Walsh
Strategic Customer Success Manager
Tableau Software

Preface

The world of data management has changed a lot in the last decade. Companies are amassing huge amounts of data every time the clock ticks. Maintaining such volumes of data and making sense of it are becoming more challenging every day. But the bigger challenge is using this data for reporting needs.

Data can come in various forms and shapes. There is a constant need for data preparation—a series of steps or processes to combine, shape, clean, and organize the data for easy usage.

Whenever data preparation is the topic of discussion, the words *Extract*, *Transform*, and *Load* (ETL) come into play. ETL simply says this: **extract** data from different sources, **transform** data into the required format, and **load** data into a system that is ready for consumption.

In the end, people who can control their data and use a combination of best Business Intelligence tools will have the right business insights and make better data-driven decisions.

In the world of Tableau, a widely used tool is Tableau Desktop, one of the best data visualization tools for ease of use, adaptability, and attractive features for end users. Tableau developers have faced issues developing the infographics attributed to the data, ranging from the shape of the data to missing information, unwanted and redundant data, mismatched elements, and the need for a basic cleanup. That limited Tableau developers' creativity, and they ended up depending on the data team. But with the introduction of Tableau Prep Builder, a new addition to the Tableau software suite, many

of those issues were resolved, and the Tableau Desktop developers breathed a sigh of relief.

It was a great move on Tableau's part to introduce this simple and fantastic tool with very powerful capabilities needed by Tableau analyst's day in and day out.

To the Reader

This book is written as a hands-on learning manual. Its aim is to cover most of the aspects and functionalities of Tableau Prep Builder. It is very understandable to users who have minimal experience with ETL tools and demonstrates the features through fundamental concepts.

Illustrations are provided throughout the book for each step, so the user can easily follow along as the chapters progress. At the same time, the book also provides great reading material. It is divided into seven chapters, with the first six focusing on content related to data preparation. The last chapter demonstrates the leverage of Tableau Prep Builder Conductor on Tableau Server. All the chapters are treated as use cases with data sets prepared and modified for training purposes.

The exercises will use data that can be download from https://www.useready.com/resources/tableau-prep-builder-book by following the instructions on the website to unpackage the file.

CHAPTER 1

INTRODUCTION to TABLEAU PREP BUILDER

1.1 Tableau Prep Builder

Tableau Prep Builder is a new add-on to the Tableau software family for preparing data by reshaping, combining, cleaning, and bringing them into a structure that is easy for visual analytics.

Here are some typical questions from Tableau designers and artisans:

> *Why do we need Tableau Prep Builder*?
> *Is Tableau Prep Builder able to solve all our preparation needs*?
> *Is Tableau Prep Builder better than other tools to solve our data issues*?

This book is not written to compare multiple tools or elaborate on the importance of Tableau Prep Builder. It has been written keeping in mind the challenges a data analyst is likely to face when the data provided or sourced is in a certain shape or format but is not ready to be pulled into the Tableau worksheet. Some publicly available data sets and sample data sets are created to illustrate solutions for some simple use cases.

While Tableau Prep Builder might not be a one-stop solution for all data problems, it is definitely an important tool from the perspective of simplicity. It brings significant functionalities without sacrificing the visual treats that Tableau Software has been consistently providing data enthusiasts and for which many Tableau developers have found good uses.

***Note:** Tableau Prep Builder comes as a free tool for creator licensees. Tableau Prep Builder Conductor is required for scheduling the workflow built into Tableau Prep Builder, which will be explained in Chapter 7.

The introduction of Tableau Prep Builder has solved many issues that Tableau Desktop developers were and are still facing on a day-to-day basis. In many cases, as mentioned before, Tableau developers have suffered or have been confined in certain methods of visualizing data, either due to the format of the data or lack of tools to prepare data as the need demanded.

Tableau Desktop offers many good features for organizing data and basic cleaning. But there was a constant need for ETL features to support the designers repairing huge data sets.

The biggest challenge was combining data across various databases. Tableau Blend is a great feature but has its own limitations. Cross Database Join, which was introduced much later, is another reliable construct, but some designers had issues using this feature due to difficulties in the collation of data, pivots of huge data sets, cross-data queries, joining live and published sources together, the union of different sources, and so on.

With the introduction of Prep, most of these issues can be solved, and the features offered have a refined and easier approach. One of the best features about Prep is the option to publish to .tde or .hyper files directly onto the Tableau Server.

Tableau Prep Builder is straightforward, with most of the features very intuitive. The major steps that can be performed using Prep are Input, Cleaning, Aggregate, Pivot, Join, Union, and Output. More details about these steps will be explained in the respective chapters. As we progress through the chapters, we can see that some of the information and steps are repeated since the intent is to focus on the topic at hand and only complete the use case. Each chapter is written by a different set of authors, and the authors have used their creativity to explain each of the concepts.

1.2 Tableau Prep Builder Interface

For the purpose of this book, we are using Tableau Prep Builder version 2019.1.2

To install the Tableau Prep Builder software, we can visit the Tableau website at *https://www.tableau.com/products* and search for the product.

When we launch Tableau Prep Builder, the home screen or landing page shows two main segments or panes and one additional hidden pane.

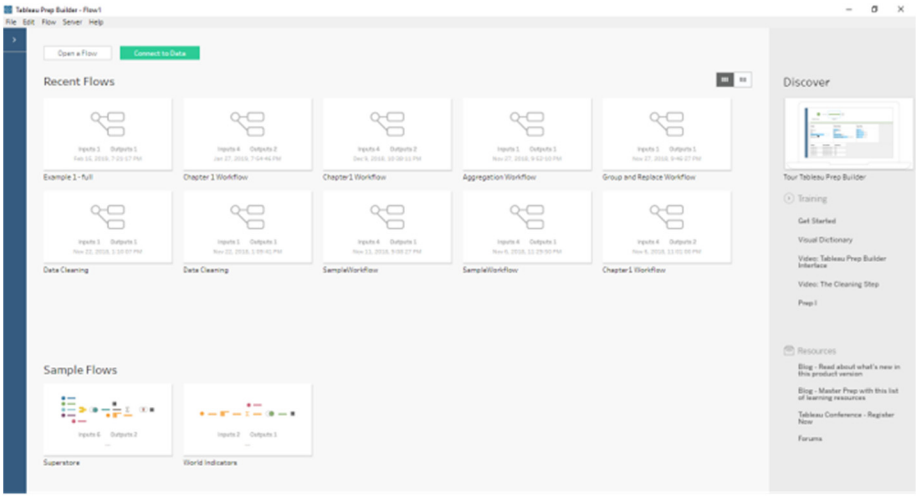


Figure 1.1 – Tableau Prep Builder - Landing Page

a. **Workflows (also called flows) –** A workflow is a sequence of steps executed in the Tableau Prep Builder environment used to prepare a clean and tidy data set for visual analytics. The pane in the center shows some of the existing workflows that are built and saved in the current machine. It also shows some sample workflows inbuilt when the software is installed. We can launch any workflow and continue working on it. We can only open one workflow at any point in time. To open two workflows at the same time, we need to launch the Prep software again.

   The workflows pane can be toggled with the default card view or the list view. The list view also shows the location of the flow.

Figure 1.2a – Recent Flows – Card view

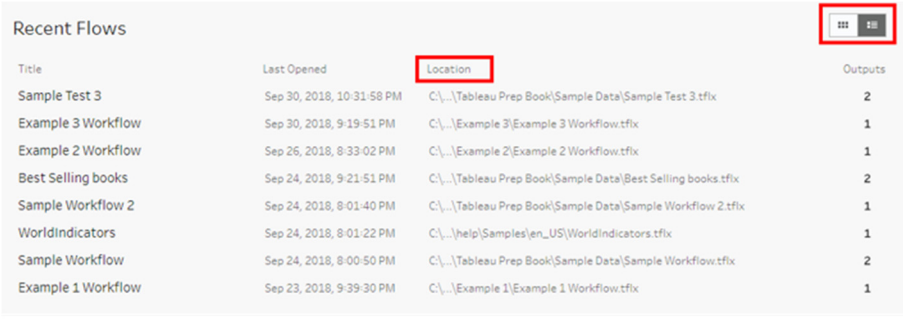


Figure 1.2b – Recent Flows – List view

b. **Discover –** The pane on the right provides a small tour of Tableau Prep Builder, quick access to training videos, and some blogs and forums that provide additional learning material.

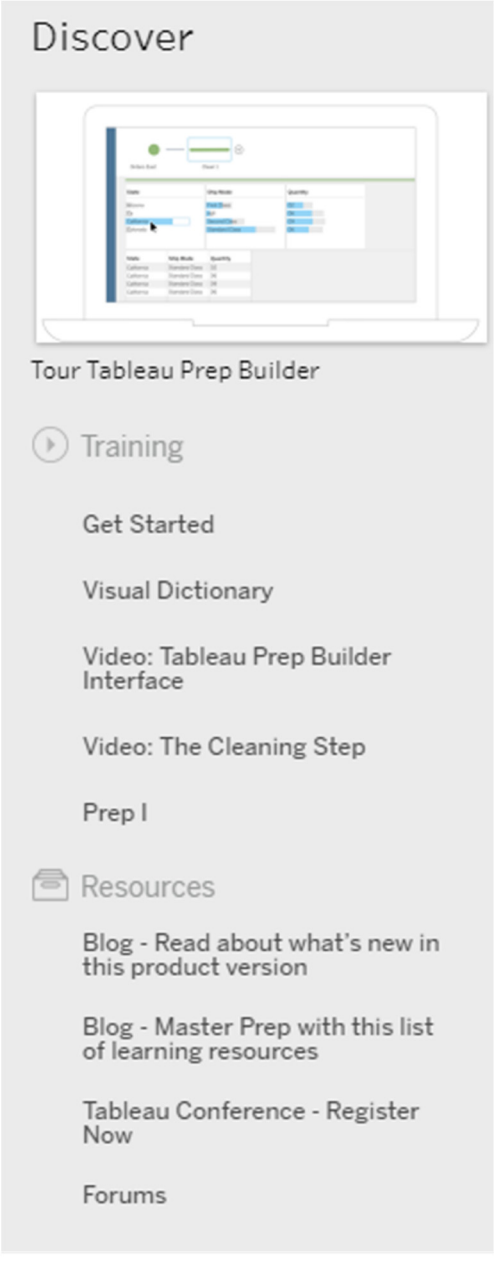


Figure 1.3 – Discover Pane

c. **Connections –** The pane on the left is not directly visible. To unhide it, use the drop-side caret  , which shows all available local flat file connections and data servers. For data server/database connections, we must download and install the appropriate drivers.

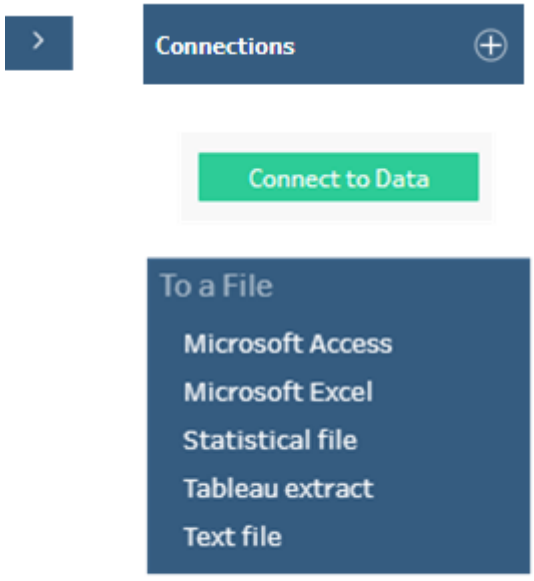


Figure 1.4a – Local files

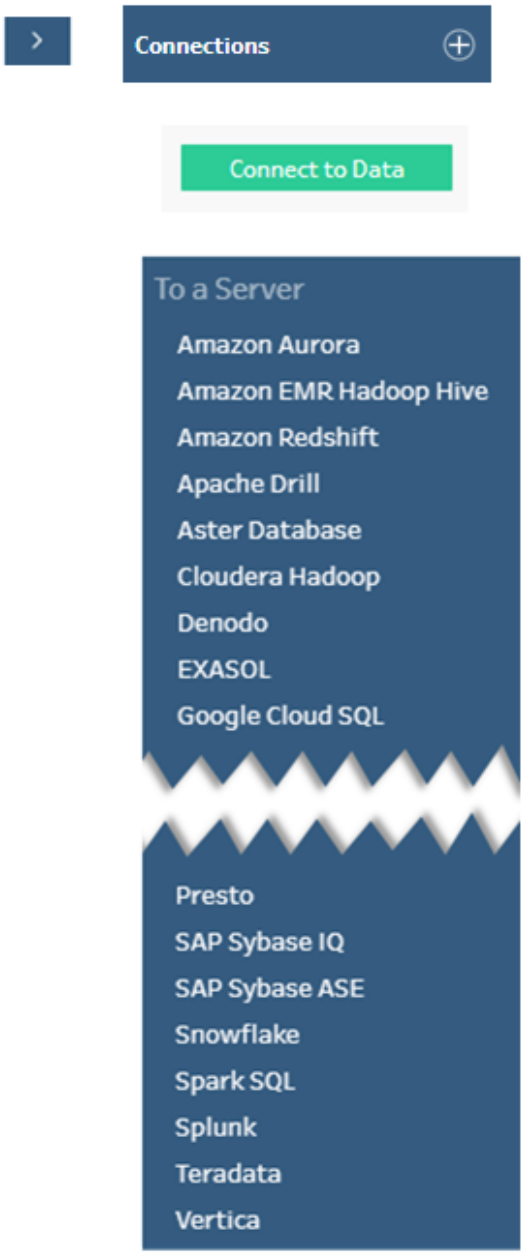


Figure 1.4b – Data Servers

1.3 Steps in Tableau Prep Builder

a. **Add Input –** Although adding input is not explicitly mentioned as a step, it is the first step that initiates a workflow, starting with a data connection.
b. **Add Step –**This is the cleaning step and the most widely used in the workflow for most cases because it offers multiple functions.
c. **Add Aggregate –** This step helps aggregate data for selected dimensions with the required summaries.
d. **Add Pivot –** This step helps shape the data. It takes a wide data set and converts it to a long data set.
e. **Add Join –** This step helps combine two streams of data based on a common field and clause.
f. **Add Union –** This step helps append two streams of data, one below the other.
g. **Add Output –** This step helps create an output after the workflow has executed.

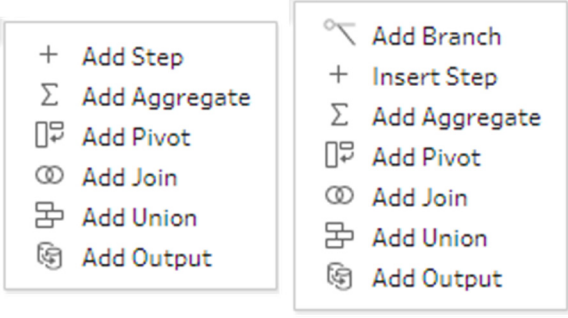


Figure 1.5 – Steps in Tableau Prep

An extra step—Add Branch—becomes available when a couple of steps are used in the workflow. This step helps branch out from the existing workflow and build another workflow using the existing input nodes to run the workflows in parallel.

For Tableau Desktop users, it easy to understand the functions because most Tableau developers have used similar features in Tableau Desktop for their data needs.

1.4 Features of a Workflow

All the features of a workflow are very intuitive and involve some graphics to help understand some complex scenarios.

As mentioned before, the first step is to use a data source.

We will skip this step and go a couple of steps ahead. We will look at the menu options and the toolbar.

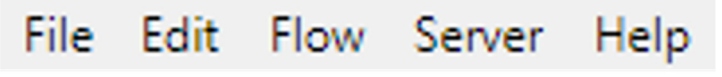


Figure 1.6 – Menu options

The options offered by the *File* dropdown are straightforward and self-explanatory like most other software tools.

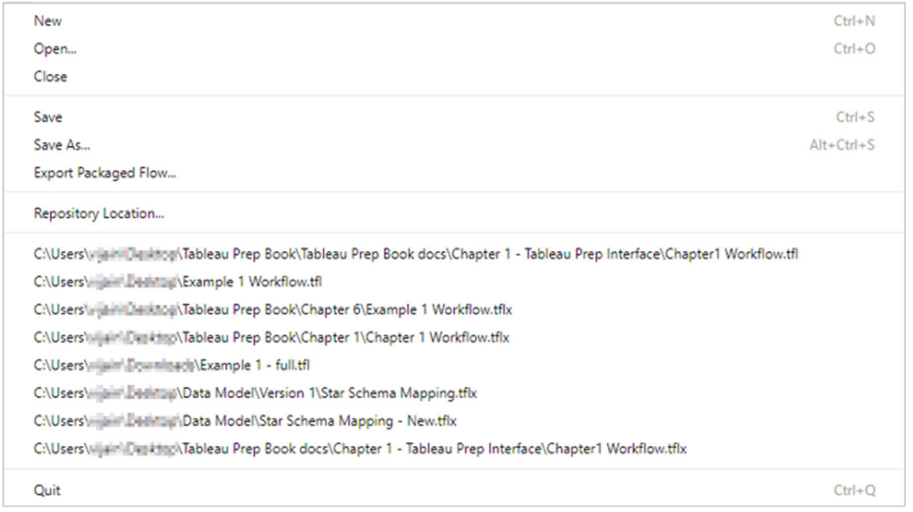


Figure 1.7 – File menu

a. **New –** This opens a new Tableau Prep Builder instance. When we use this option, it closes the existing workflow. If we have not saved it, a prompt appears to save the work in progress.
b. **Open –** This opens an existing saved workflow from the machine.
c. **Close –** This closes the existing workflow and redirects to the landing page to start with data connections.
d. **Save –** This saves a workflow or overwrites the work in progress.

e. **Save As –** We can save the workflow as a .tfl or .tflx extension.

.tfl – Prep Workflow	Saves the workflow only
.tflx – Prep Packaged Workflow	Saves the workflow along with local data files as a bundle

Now we need to go a little deeper with this concept. Most Tableau Desktop users are familiar with the concept of .twb and .twbx. The same principle is applied to .tfl and .tflx. The .tfl is more like a skeleton or framework that contains only the components of the workflow. When a .tfl is sent to another Tableau Prep Builder user, that user must have access to the data files required to run the flow. The .tflx works as a zip file that contains the workflow and the local files. Any Tableau Prep Builder user can run the workflow without navigating to the required data files.

.tfl file

.tflx file

f. **Export Packaged Flow –** This directly saves as a Packaged Workflow.

g. **Repository Location –** This allows us to change the Prep Builder repository location needed to store some data sources or flows which by default is under Documents.

h. **Recent Workflows –** This opens any of the recently saved workflows.

i. **Quit –** This exits from the Prep application.

The options available on the *Edit* dropdown are also very easy to interpret. There are some basic options to use on the existing workflow. There are also some keyboard shortcuts.

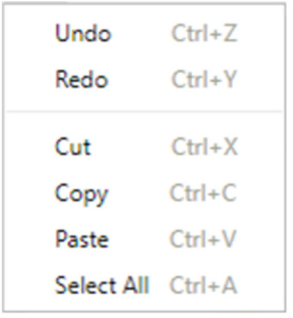


Figure 1.8 – Edit menu

The *Flow* menu has only one option: Run All. In Tableau Prep Builder, we have the advantage of running each of the outputs individually or all at once.

Figure 1.9 – Flow menu

The *Server* menu has the option to sign in to a Tableau Server or Tableau Online (as shown in the image). It also allows to Publish a workflow which is discussed in Chapter 7. Publish option is available only for Prep Builder version 2019.1.1 and above. The feature is not available for lower versions.

Figure 1.10 – Server menu

The *Help* menu, as the name suggests, provides some options for assistance while learning Tableau Prep Builder.

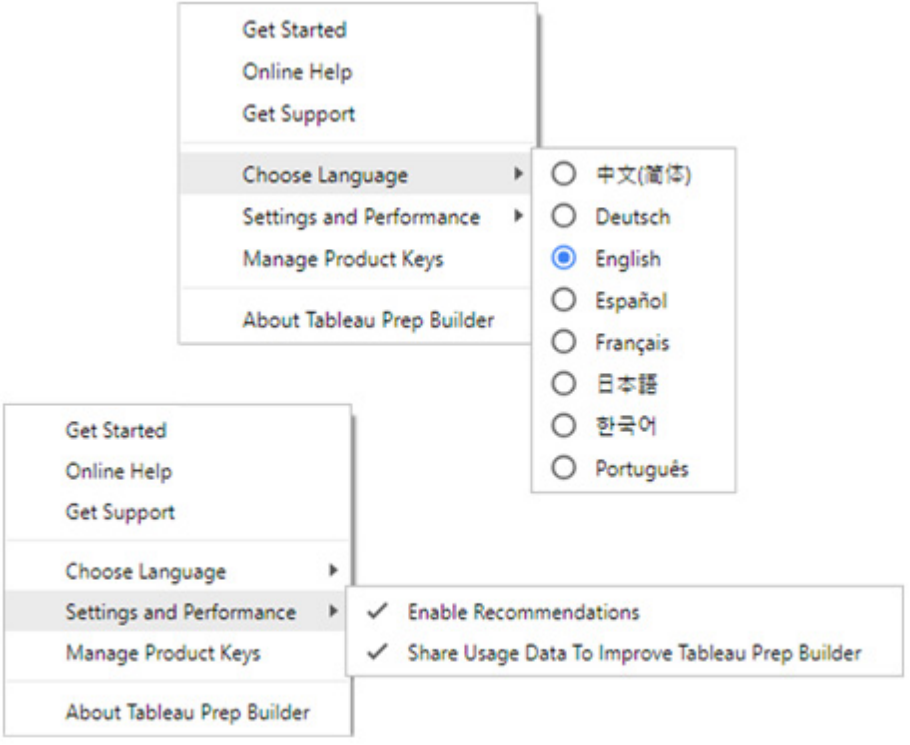


Figure 1.11a – Help menu

a. **Get Started –** Shows the Tableau Prep Builder tour video, which is the same as the one in the Discover pane of the Landing page.
b. **Online Help –** Launches the Tableau website that has information about Tableau Prep Builder, more like an online manual.
c. **Get Support –** Launches Tableau's support page to find answers to specific product-related issues.
d. **Choose Language –** Allows us to choose our language of preference. In this case, the default is English.
e. **Settings and Performance –** Allows us to select or deselect Workflow recommendations for Data type, Data Role, Geographic role, Nulls etc. and share usage information with Tableau.
f. **Manage Product Keys –** Opens a dialog box to enter the product keys.

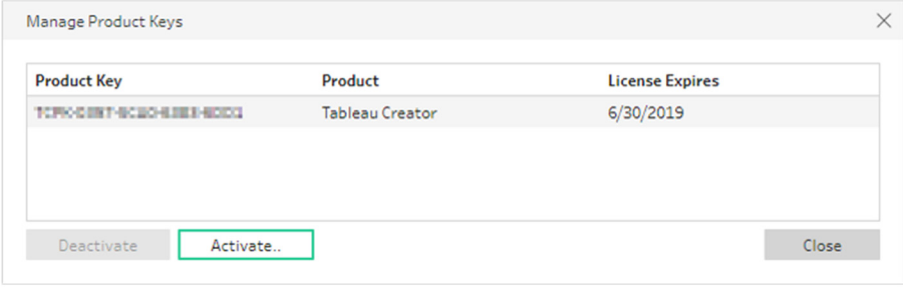


Figure 1.11b – Product keys

g. **About Tableau Prep Builder –** Shows the current version of Tableau Prep Builder being used. This book is written using Prep Builder 2019.1.2. Every new version and release will have small variations in the User Interface.

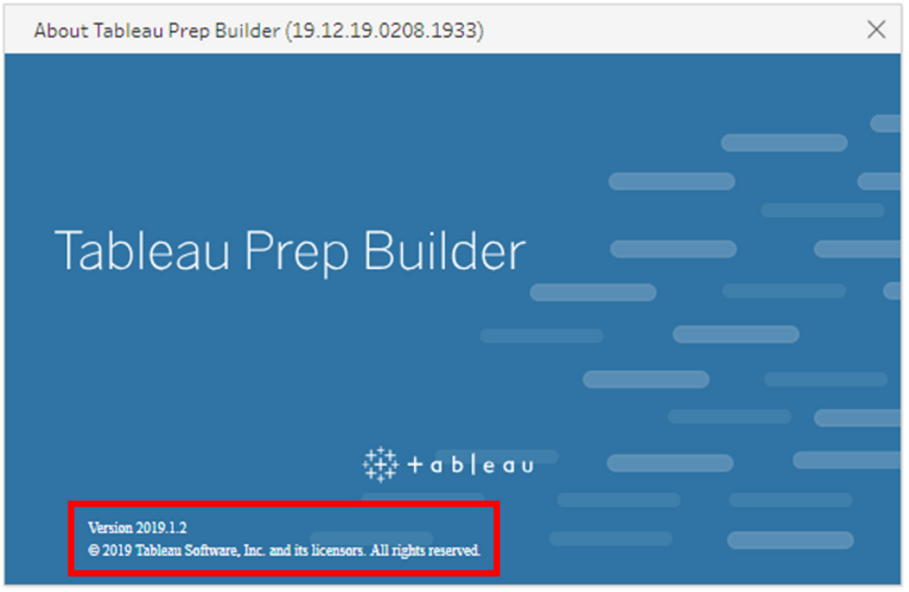


Figure 1.11c – Tableau Prep Builder Version

Once the data connection is established, we can see all the available tables in that particular data set or database. If we are using a single table, it is automatically brought into the workflow.

To quickly skim through the parts of a workflow, we will use the Chapter 1 Workflow.tflx, a packaged workflow, as an example. The workflow is stored in the Chapter 1 folder. We can double-click on the workflow or launch the Tableau Prep Builder software and open it as a workflow from the Landing page or by using the File menu.

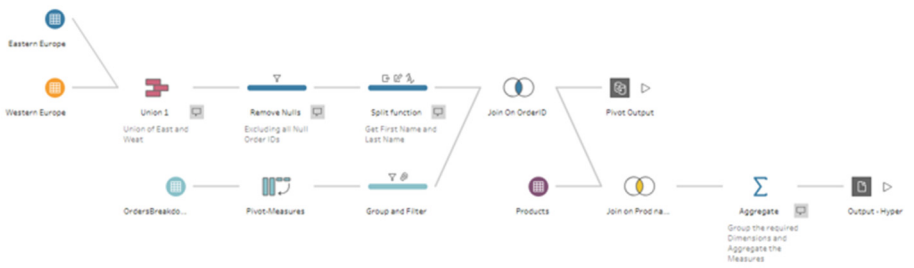


Figure 1.12 – Chapter 1 Workflow

This is a simple workflow with four Input steps, a Union step, a few Cleaning steps, a Pivot step, two Join steps, an Aggregate step, and two Output steps.

The entire canvas is divided into two parts. The main part on the top is the workflow area. The bottom part provides access to configuration, field settings, profile pane, and preview. The bottom part only opens up after a particular step is selected.

A couple of features get repeated in every step other than the standard configuration features.

1. Changes tab that helps in tracking all the actions performed in that step which is not available in the Output step.
2. Filter option on the contextual toolbar to remove unwanted values which is not available in the Output step.
3. Calculated field option for customized fields on the contextual toolbar which is not available in the Input and Output steps.

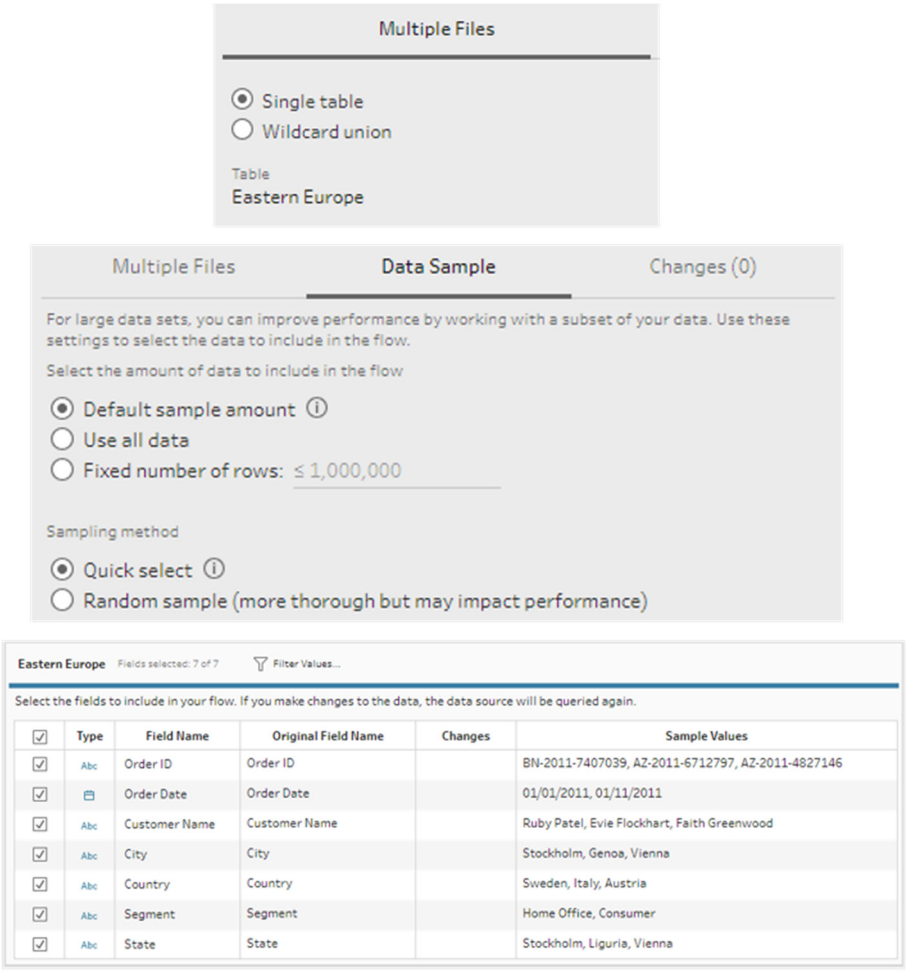


Figure 1.13 – Input step features

The Input step provides options for sheet selection, data sample, field configuration, and properties. The details will be discussed in Chapter 2.

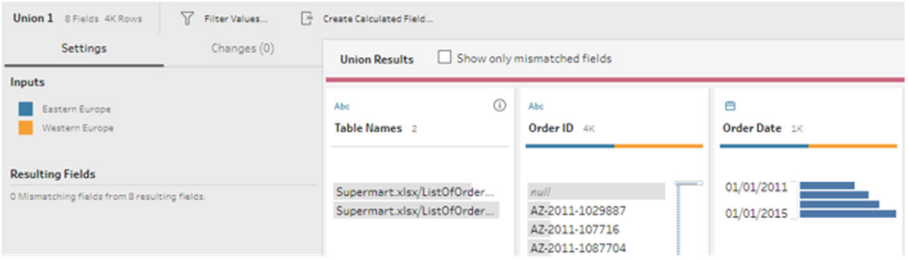


Figure 1.14 – Union step features

The Union step shows the selected tables and some field properties. More details are covered in Chapter 6.

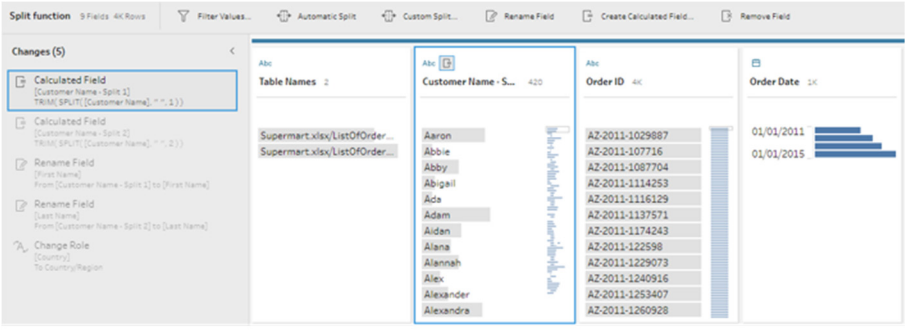


Figure 1.15 – Cleaning step features

The Cleaning step is the most widely used step since it offers a wide variety of functions, including calculations, renaming fields, filters, removal of fields, grouping and so on. Chapter 3 discusses all features of cleaning.

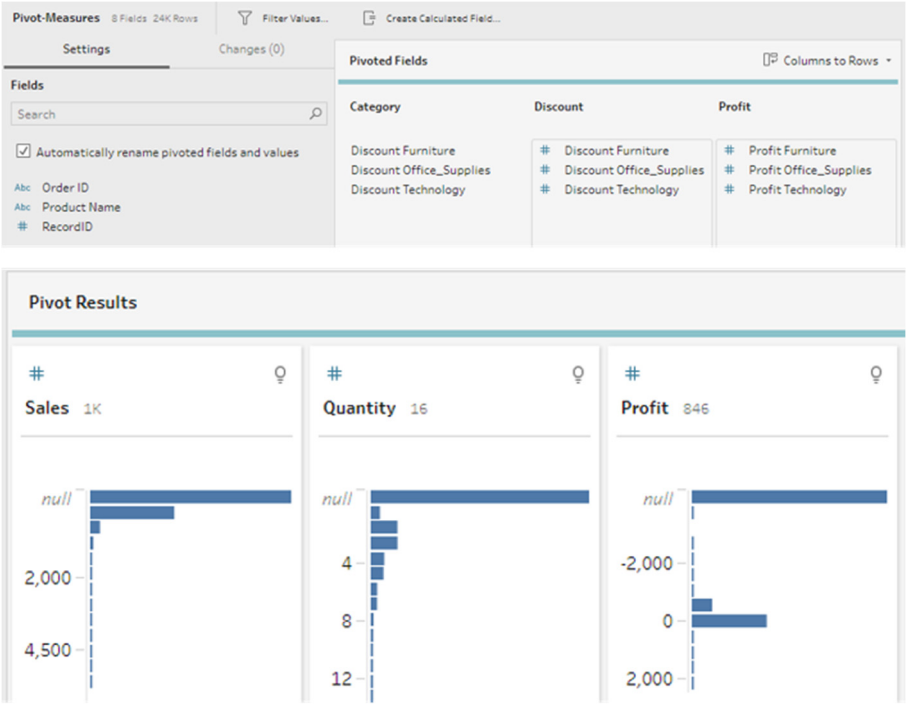


Figure 1.16 – Pivot step features

The Pivot step is a great feature that transposes a wide data set into a columnar structure. The grouping feature for the required Dimensions and selection of required Pivot measures will be discussed in Chapter 5.

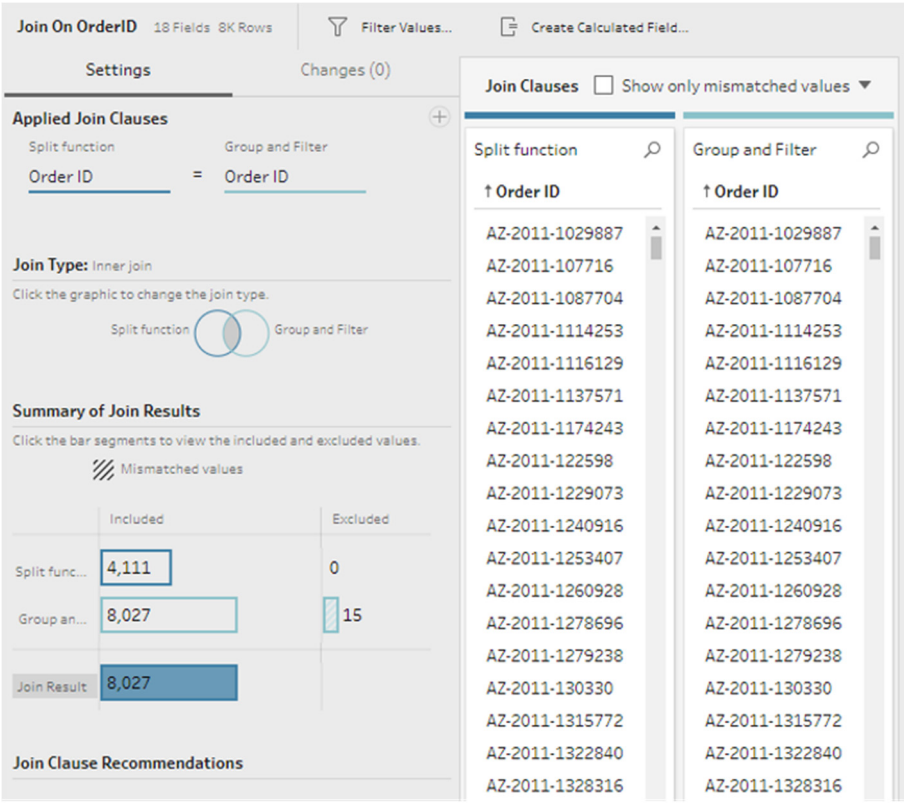
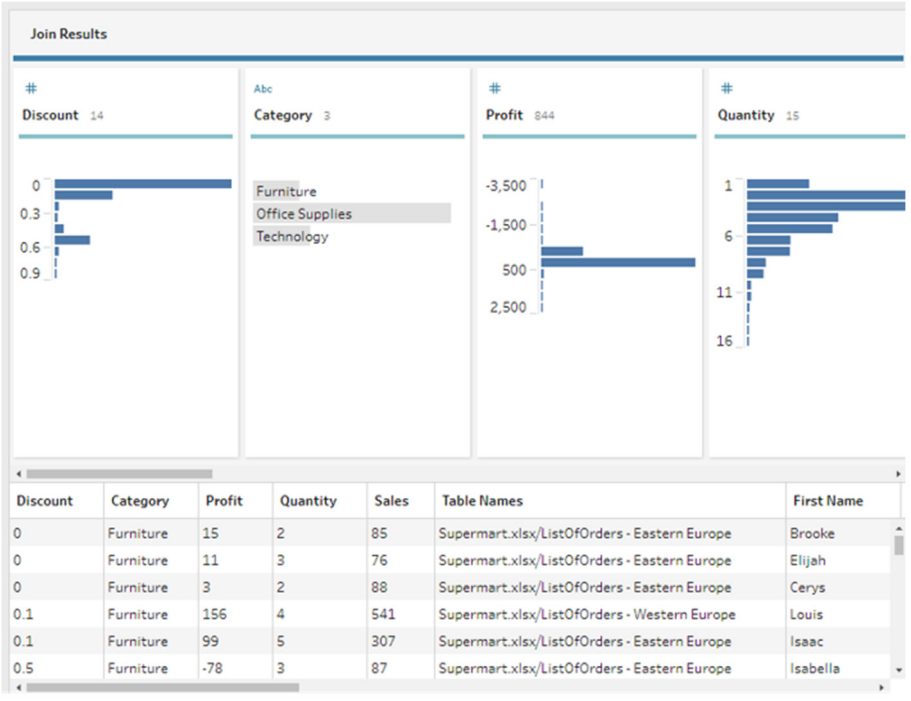


Figure 1.17a – Join step features



Discount	Category	Profit	Quantity	Sales	Table Names	First Name
0	Furniture	15	2	85	Supermart.xlsx/ListOfOrders - Eastern Europe	Brooke
0	Furniture	11	3	76	Supermart.xlsx/ListOfOrders - Eastern Europe	Elijah
0	Furniture	3	2	88	Supermart.xlsx/ListOfOrders - Eastern Europe	Cerys
0.1	Furniture	156	4	541	Supermart.xlsx/ListOfOrders - Western Europe	Louis
0.1	Furniture	99	5	307	Supermart.xlsx/ListOfOrders - Eastern Europe	Isaac
0.5	Furniture	-78	3	87	Supermart.xlsx/ListOfOrders - Eastern Europe	Isabella

Figure 1.17b – Join step features

The Join step provides options to join two tables based on common fields. The details on various types of joins will be discussed in Chapter 6.

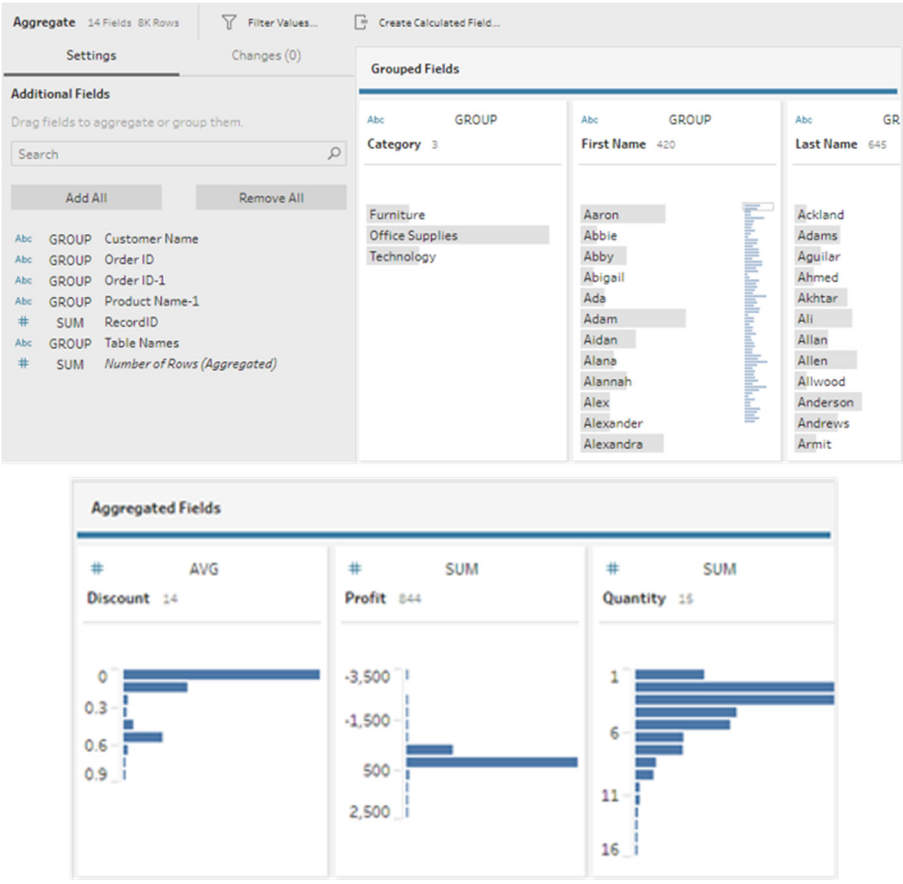


Figure 1.18 – Aggregate step features

The Aggregate step provides an option to Group by required Dimensions and Summarize the required Measures, which are explained in Chapter 5.

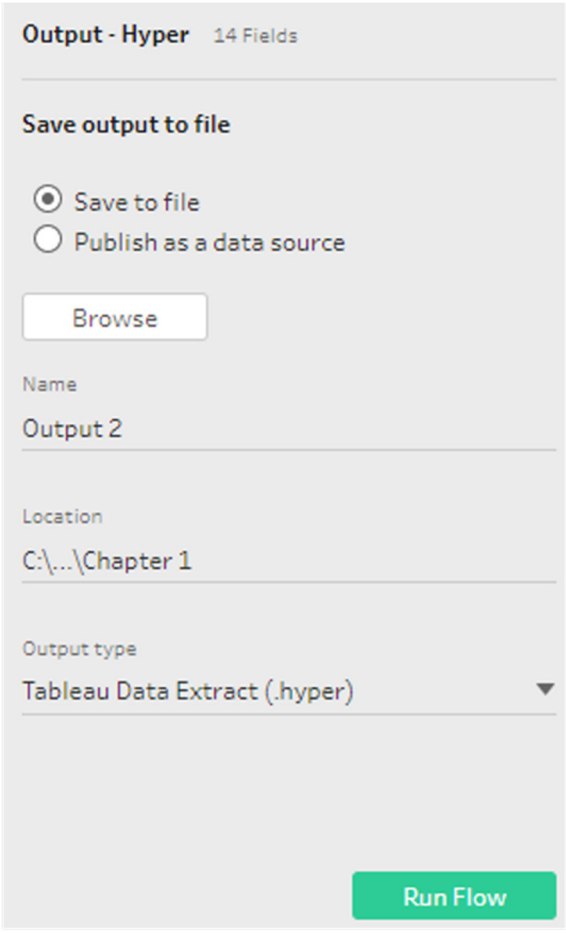


Figure 1.19 – Output step features

Finally, the Output step shows a preview of the output and the location to save the output, which is explained in Chapter 2.

There are some other features that might be of interest to developers. All the steps in the workflow have an option to rename and write a description of 200 characters. That helps other developers understand the features of the workflow at a quick glance.

We can also right-click on a step and change some features like color, description, copy etc.

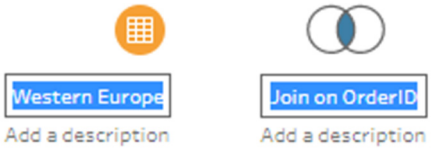


Figure 1.20a – Renaming a step

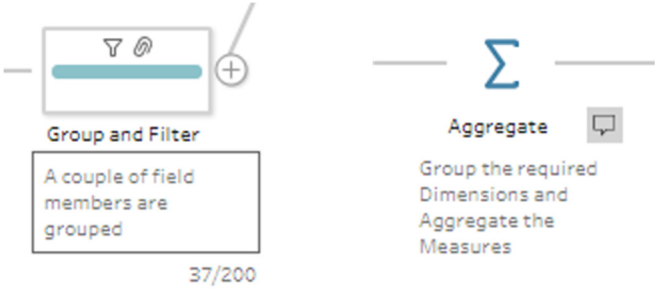


Figure 1.20b – Description for a step

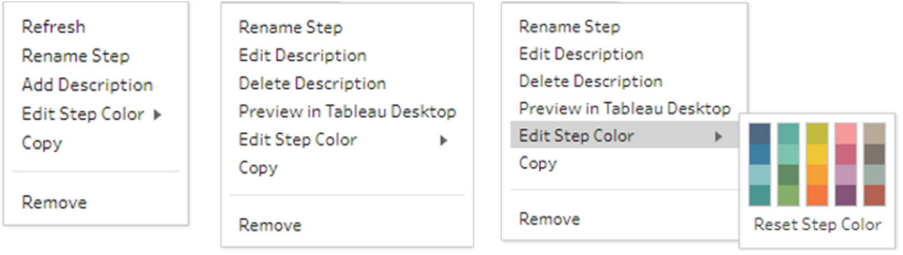


Figure 1.20c – Right-click properties

As previously discussed, the bottom window of the canvas provides a small toggle button to either display or hide the Preview pane. That option is not available for the Input or Output steps.

There is also a small search bar in the right corner of the settings window.

Figure 1.21a – Toggle for Preview

Figure 1.21b – Search bar

There is also a very simple toolbar in the left corner that provides options to Undo or Redo some steps. There are also options to refresh individual data sources used in the flow. (The data source refresh does not provide new values if it is a packaged workflow.) Similarly, we can run or execute individual outputs.

Figure 1.22 – Toolbar

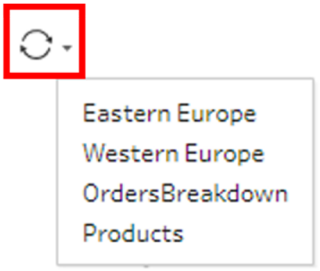


Figure 1.22a – Data Refresh

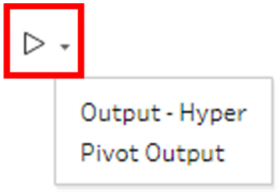


Figure 1.22b – Workflow Execution

In the top right corner is a small bell icon that displays errors and workflow execution progress. A small dialog box opens at the same time to show the progress of the workflow.

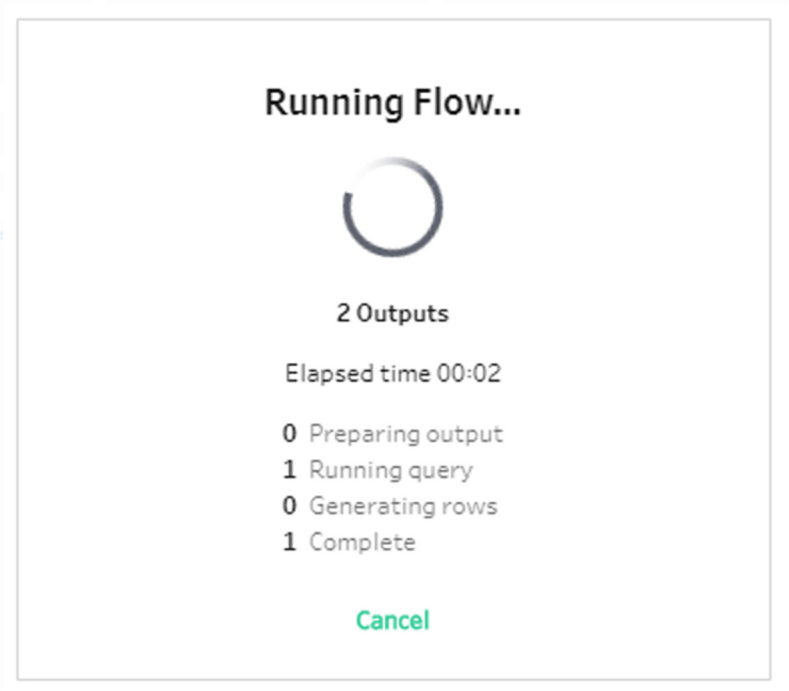


Figure 1.23a – Workflow Progress

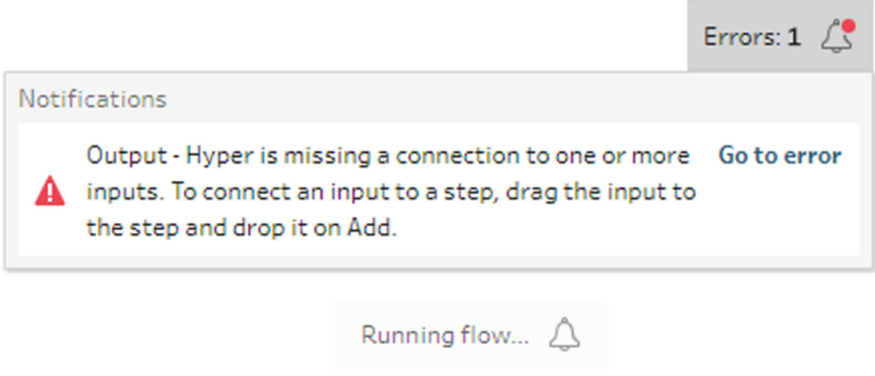


Figure 1.23b – Workflow notifications

CHAPTER 2

INPUT and OUTPUT

When we need to clean data or make amendments to a data set, the first step is to import the required amount of data into the workflow. In this chapter, we will discuss the Input and Output steps, perform basic functions in the Input configuration window, detail how to connect to data, input data, filter on a field, output to a file, and preview in Tableau.

As we saw in Chapter 1, a Tableau Prep Builder workflow starts with the Input step. Since Prep cannot create data, we have to identify sources such as files or databases to make the data visualization-ready.

This chapter is divided into sections that cover various types of data sources, input step features, and configuration options. First, we will explain how to connect to a Microsoft Excel (.xlsx) file. Then we will explain how to connect to other files such as Tableau Data Extract (.tde), Text file (.csv), Snowflake Server, and Microsoft SQL Server.

This chapter introduces all the data sets that are used in the upcoming chapters. Although the SQL Server and Snowflake Server are used as examples, all topics in each chapter will be explained using only local files.

2.1 Connect to Microsoft Excel

We will be connecting to an Excel file called *Cancels and Corrections.xlsx* found in the Chapter 2 folder. It is sample financial trade risk data that contains information about the status of the trade, which can be either Correction or Cancellation

based on the progress of a trade transaction. The important fields in the data are Trade ID (similar to transaction ID), Trade Date (date of trade initiation), Modified Date, Settlement Date, Category of the Trade, Stock Name, Geographic Details (country, state, city, etc.), Internal and External Trade values, and so on. The data is collected for 2014–2017.

After launching the Tableau Prep Builder tool, the Connections option is displayed on the left side. Clicking the + icon lets us connect to a File or a Server. We will use Microsoft Excel and navigate to the file Cancels and Corrections.xlsx.

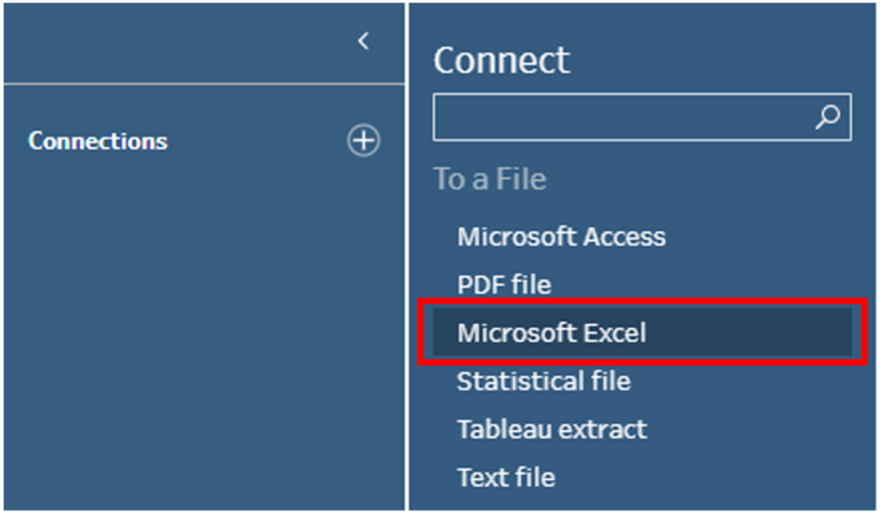


Figure 2.1 – Connection to Microsoft Excel (.xlsx)

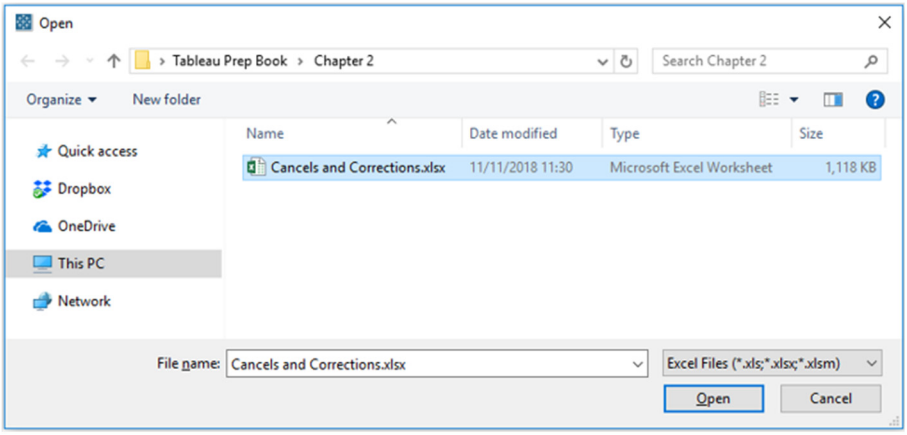


Figure 2.2 – Connection to .xlsx

We are now connected to the Excel source file. As we can see, the data set contains four tables, one for each year. More functions of Tableau Prep Builder using this data set will be covered in Chapter 6, Joins and Unions.

The next step is to import and preview the data set, which is covered in the following sections.

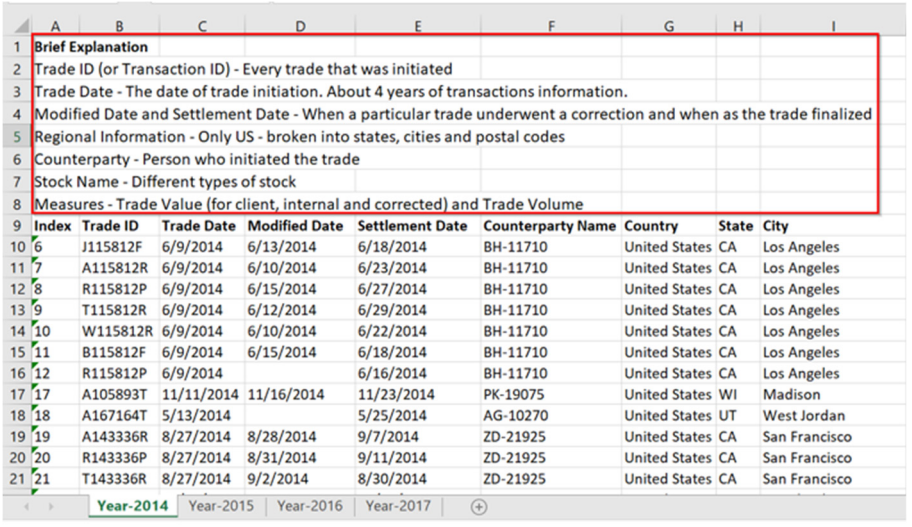
2.2 Data Interpreter

Like Tableau Desktop, Tableau Prep Builder also comes with Data Interpreter. It is very useful in cleaning and detecting unwanted headers or reporting information in the file.

> *__Note:__ The functionality is available only for Microsoft Excel at this point.

When we open the Excel file outside of Tableau Prep Builder, we see that the first few rows of the file contain the description of the

columns. We do not need these rows. Deleting the rows manually is not the right way to approach it. We can use the *Use Data Interpreter* option that Tableau Prep Builder offers (similar to the feature in Tableau Desktop) to focus on the data to be imported.



	A	B	C	D	E	F	G	H	I
1	**Brief Explanation**								
2	Trade ID (or Transaction ID) - Every trade that was initiated								
3	Trade Date - The date of trade initiation. About 4 years of transactions information.								
4	Modified Date and Settlement Date - When a particular trade underwent a correction and when as the trade finalized								
5	Regional Information - Only US - broken into states, cities and postal codes								
6	Counterparty - Person who initiated the trade								
7	Stock Name - Different types of stock								
8	Measures - Trade Value (for client, internal and corrected) and Trade Volume								
9	**Index**	**Trade ID**	**Trade Date**	**Modified Date**	**Settlement Date**	**Counterparty Name**	**Country**	**State**	**City**
10	6	J115812F	6/9/2014	6/13/2014	6/18/2014	BH-11710	United States	CA	Los Angeles
11	7	A115812R	6/9/2014	6/10/2014	6/23/2014	BH-11710	United States	CA	Los Angeles
12	8	R115812P	6/9/2014	6/15/2014	6/27/2014	BH-11710	United States	CA	Los Angeles
13	9	T115812R	6/9/2014	6/12/2014	6/29/2014	BH-11710	United States	CA	Los Angeles
14	10	W115812R	6/9/2014	6/10/2014	6/22/2014	BH-11710	United States	CA	Los Angeles
15	11	B115812F	6/9/2014	6/15/2014	6/18/2014	BH-11710	United States	CA	Los Angeles
16	12	R115812P	6/9/2014		6/16/2014	BH-11710	United States	CA	Los Angeles
17	17	A105893T	11/11/2014	11/16/2014	11/23/2014	PK-19075	United States	WI	Madison
18	18	A167164T	5/13/2014		5/25/2014	AG-10270	United States	UT	West Jordan
19	19	A143336R	8/27/2014	8/28/2014	9/7/2014	ZD-21925	United States	CA	San Francisco
20	20	R143336P	8/27/2014	8/31/2014	9/11/2014	ZD-21925	United States	CA	San Francisco
21	21	T143336R	8/27/2014	9/2/2014	8/30/2014	ZD-21925	United States	CA	San Francisco

Year-2014 | Year-2015 | Year-2016 | Year-2017

Figure 2.3 – Need for Data Interpreter

If we see the preview pane after importing the 2014 table, the field names are obscured and shown as F2, F3, and so forth, which is not useful. All the column headers are also missing and all sample values are Null. That is because the Excel sheet has extra information as headers that must be cleansed before importing.

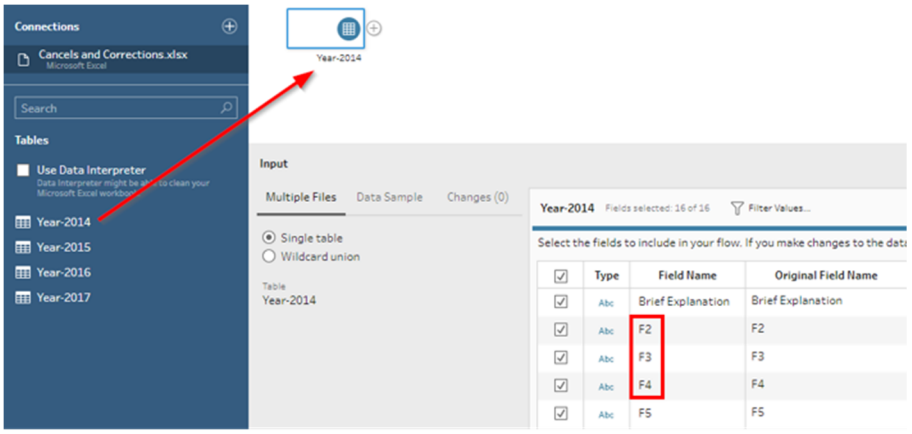


Figure 2.4 – Importing data without Data Interpreter

When we check the *Use Data Interpreter* checkbox, the tool automatically cleans the Excel sheet and imports the correct data.

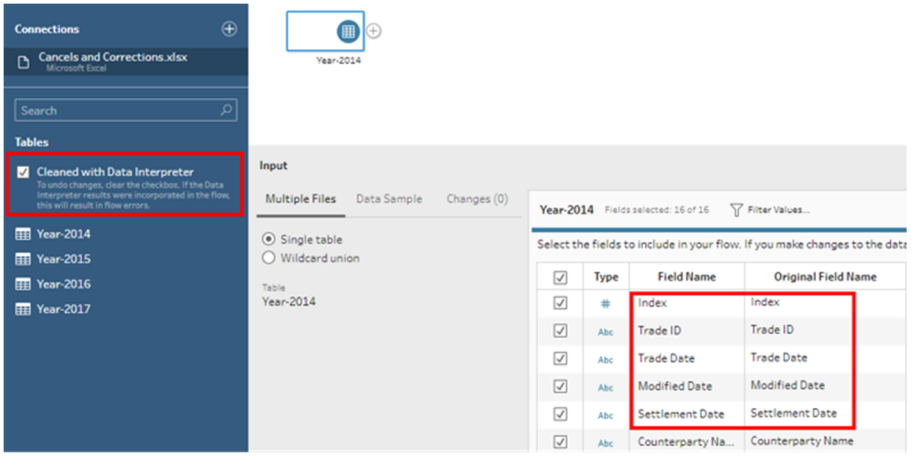


Figure 2.5 – Importing data with Data Interpreter

2.3 Configuration Window

The Configuration window holds the metadata profile of the data set. It also provides features to search for fields, select or deselect fields to be included, change the field type, rename the fields, filter on a condition, and inspect sample values, which we will discuss one by one.

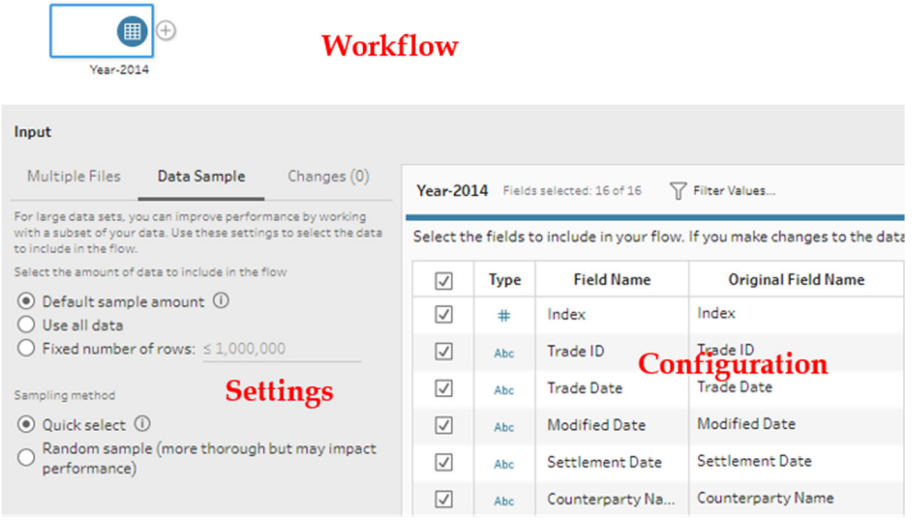


Figure 2.6 – Panes in the Input step

a. **Field Selection or Removal**

Often, we may want to select only specific fields from a data source file according to the business need. Tableau Prep Builder makes it very easy to select only the required fields and omit the rest while importing.

To the right, we can see the Configuration Window. By default, all fields are selected. We can uncheck the fields that are not

required—in our case, *Index* and *Country*. We see that 14 of 16 fields are selected. Changes can be tracked in the Changes tab. When a field is removed or changed, we can see the visual cure. Though field selection can be done in the Input step, it is advisable to see the preview in the Cleaning step and remove the fields there. Thist way, we have a better picture of the fields that are being removed especially when the Data Interpreter is used.

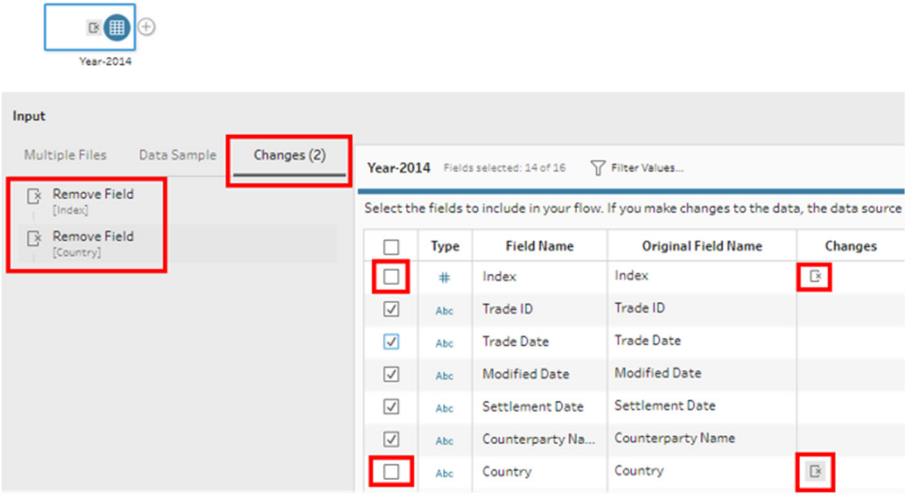


Figure 2.7 – Field selection in Input step and the Changes tab

b. Filtering on a Field

Filtering helps set criteria and conditions on a large dataset or a data table so only the required data is brought into the rest of the data flow. It is a very important step in data profiling and helps focus on content-specific information.

For this example, we will filter to show only stocks having Trade Volume Units greater than 2. We will use the *Filter Values* option on the contextual toolbar.

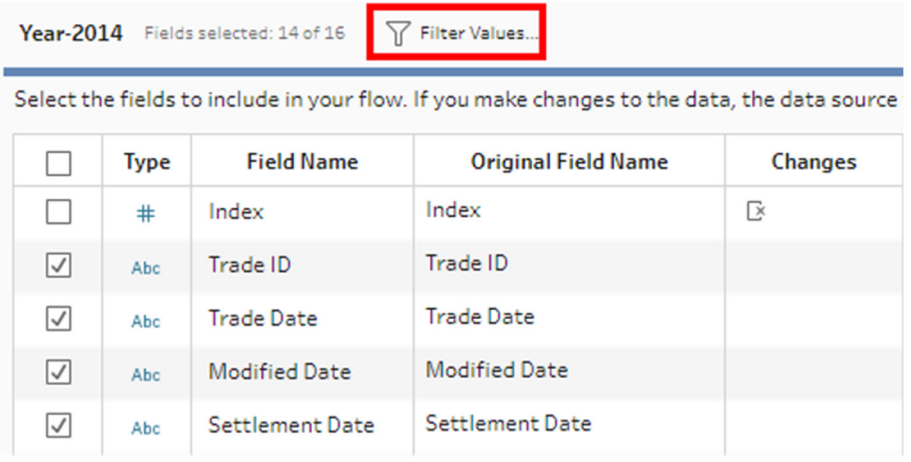


Figure 2.8 – Filter using the contextual toolbar

This option opens a new calculation editor window. We can enter the condition (in this case, *[Trade Volume (Units)]* > 2), which returns a Boolean value. The filter will be applied to every row.

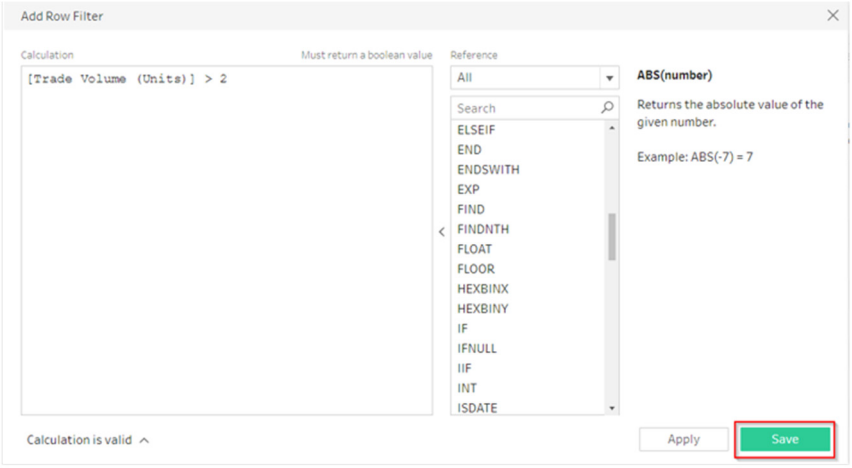


Figure 2.9 – Filtering on a Field – Calculation Window

When we click on *Save*, the filter will be applied. We can click on the filter to edit or remove it from the dropdown menu.

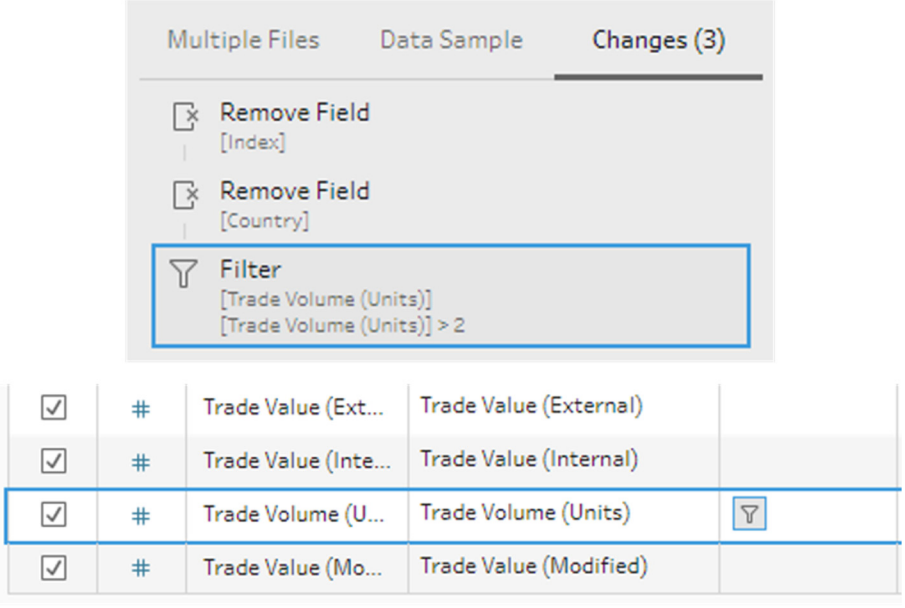


Figure 2.10 – Filter display

2.4 Data Sampling

After data cleansing, it is necessary to sample the data (large amounts of data in most cases). There are two settings available in Tableau Prep Builder for data sampling—*Amount of Data* and *Sampling Method*.

The following sampling options are displayed in the Input pane.

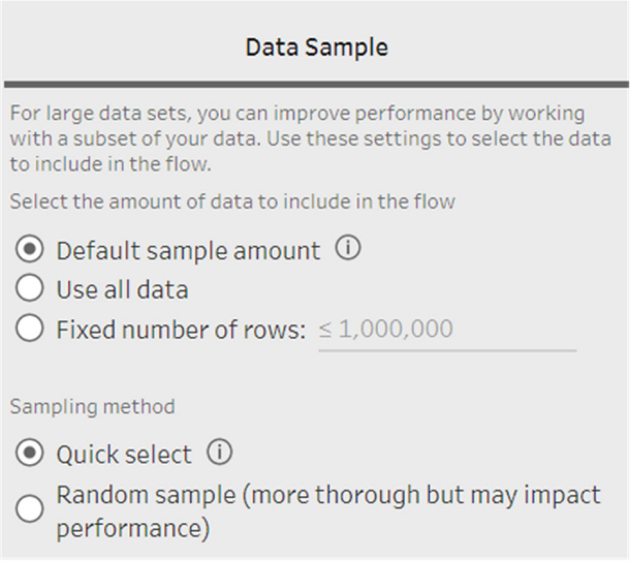


Figure 2.11 – Data Sampling in Tableau Prep

For the first setting option, the sampling amount determines the number of records. Here, we get the following three options:

a. **Default Sampling Amount –** It does not have a fixed row count; it is based on the total number of fields and field data types in the data set.

b. **Use All Data –** That includes all data without sampling. It forces Tableau Prep Builder to retrieve all rows in the data.
c. **Fixed Number of Rows –** This includes only the specified amount of records in the sample.

The second section in the Data Sampling method determines how the records are chosen. We have two options:

a. **Quick Select –** This option returns the specified number of records as quickly as possible. Although it is faster than random sampling, it results in a biased sample. Only the first few rows of sorted data in the database or cached in memory are included.
b. **Random Sample –** This option looks at each row in the dataset and returns the requested records randomly to make a better representative sample. Since it scans the entire data set, performance issues may persist.

***Note:** Quick Select is not a good representation of the actual data. As the name suggests, it is used for faster analytics.

For this example, we will use *Default Sample Amount* and *Quick Select*.

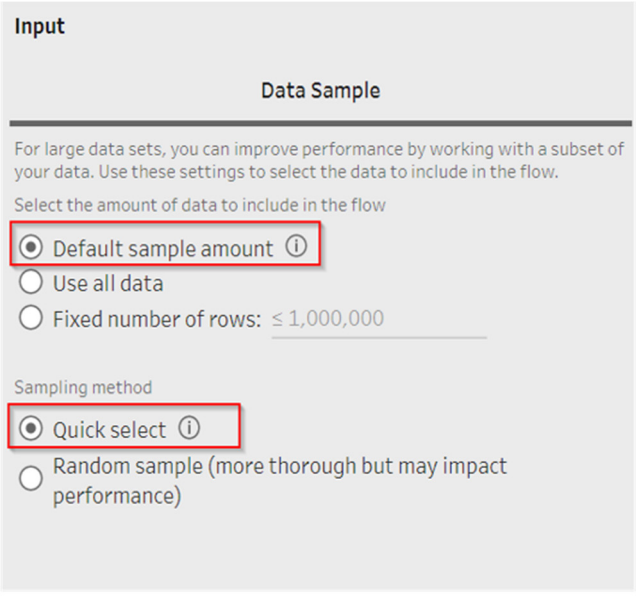


Figure 2.12 – Data Sampling - Default Sampling and Quick Select

To quickly verify the data, click on the + next to the *Input* step and select *Add Step*, which is the cleaning step. More features of the cleaning step will be explained in Chapter 3. By default, it is named Clean 1.

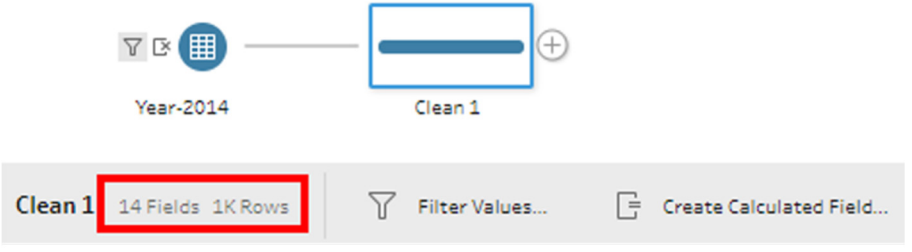


Figure 2.13 – Data Sampling – 1K rows

We can notice that the number of rows after sampling is still 1K. Since the number of rows in the input is small, Tableau Prep Builder included all the rows of the flow. As mentioned earlier,

there is no fixed row count for a sampling amount in Tableau Prep Builder.

Now we will try *Fixed Number of Rows* and *Random Sample* by entering the required number of rows (e.g., 500).

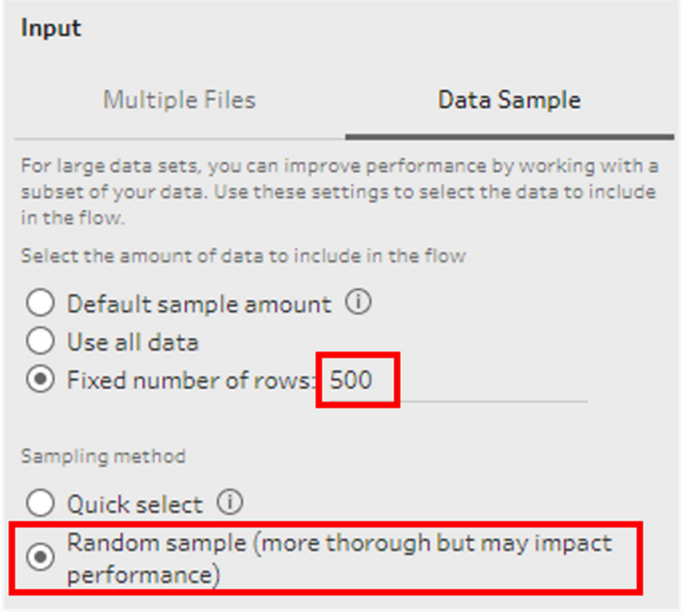


Figure 2.14 – Fixed Rows and Random Sampling

Now, the configuration window of *Clean 1* shows 500 sample rows.

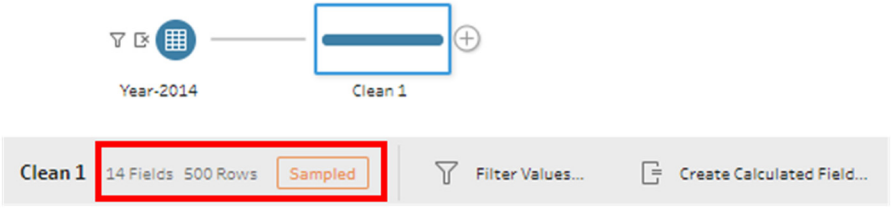


Figure 2.15 – Fixed Rows and Random Sampling – Imported data

The order of indexes indicates that the rows are randomly sampled. Random sampling may affect the performance but provides a good representation of the population.

Tableau Prep Builder users may experiment choosing *number of records* and *sampling methods* to optimize performance and form the right sample.

When discussing data sampling, performance is a major player. *Quick Select* will contain whatever rows the data source provides. It may run into sampling issues such as not covering the full range of values in the data set. For example, it may consider and show only one month's worth of data when the original data spans a year. In contrast, we can select *Random Sample*. That will take longer to generate and may cause the workflow to slow down and impact the operation since it requires the sample to be regenerated over and over again. However, it might result in a superior representation of data if not the ideal one. To get the most representative sample, we apply all pertinent filters and deselect unwanted fields in the *Input* step. Narrowing the scope of fields or rows in the input before sampling will maximize the sample's usefulness and ensure the sample is as robust or holistic as it can be.

2.5 Connect to Tableau Data Extract File

The required file for this example is in the Chapter 2 folder. For the purpose of demonstration, we will work with Resource Utilization data (found in *Unorganized and Associates* Tableau Hyper Extract file). The data set is about a professional services firm and its consultants. The fields used are *Date*, *Consultant*, and *Client* (the one the consultant is working for), and *Number of*

Hours (the number of hours spent and billable, whether the assignment is billable).

As the previous example in the *To a File* section, we use the *Tableau Extract* and navigate to the file *Unorganized and Associates.hyper*.

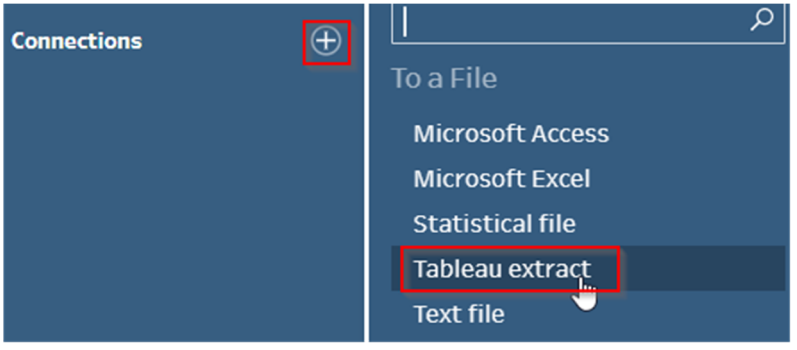


Figure 2.16 - Connecting to Tableau Data Extract (.tde)

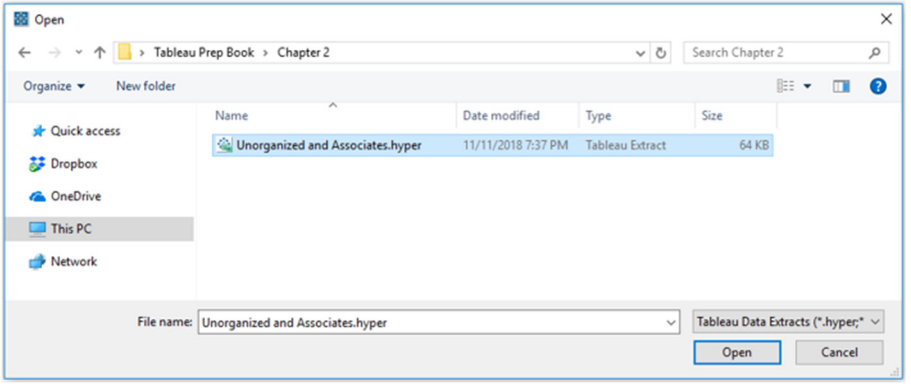


Figure 2.17 – Connection to Hyper file

We are now connected to the data. We will do more detailed operations on the data in Chapter 4, Group and Replace.

2.6 Connect to Text File

The .csv file *Recycling_Diversion_and_Capture_Rates.csv* found in the Chapter 2 folder contains data about Recycling Diversion Rates and Capture Rates captured at each district level. A very important field in the data, *The Diversion Rate-Total*, is the percentage of total solid waste collected by the Sanitation Department and disposed of by recycling. The *Capture Rate Total* is the percentage of total paper or metal/glass/plastic in the waste stream that is disposed of by recycling.

Let us import the data into Tableau Prep Builder. Similar to the previous example, we will use *Text File* as the data connection option to navigate to the file *Recycling_Diversion_and_Capture_Rates.csv*.

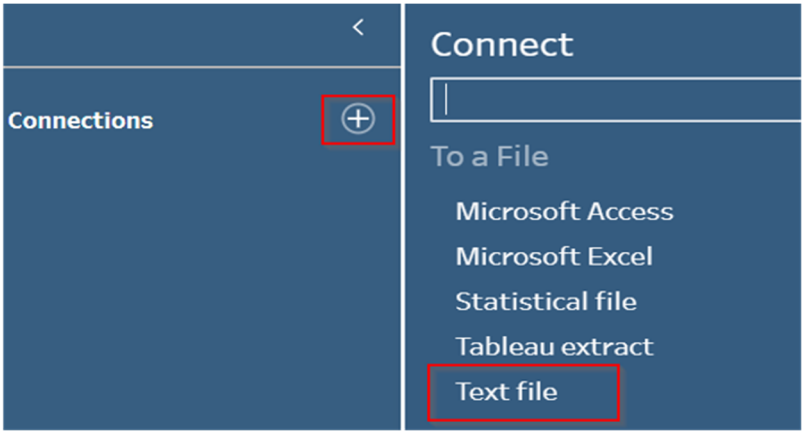


Figure 2.18 - Connection to Text File (.csv)

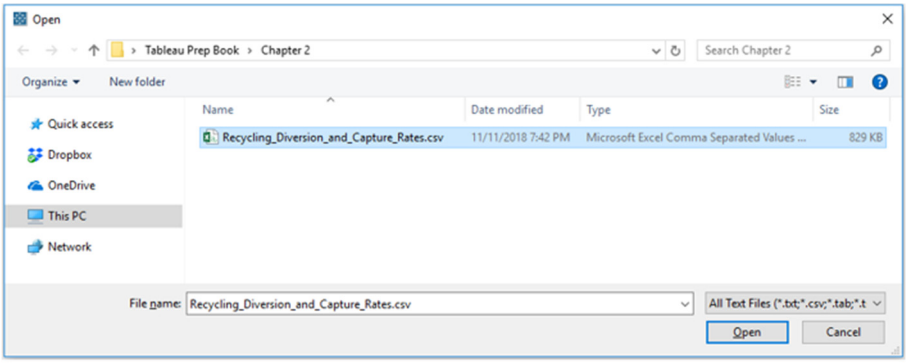


Figure 2.19 – Connection to CSV file

We are now connected to the text source file. Further exploration of this data set will be covered in Chapter 5, Aggregation and Pivot.

2.7 Connect to Snowflake Server

In this example, we will look at the connection to a Snowflake data server. The data set is *Telco Churn Dataset*, which contains customer-churn details. Customer churn is one of the main problems in the telecommunications industry. Several studies have shown that attracting new customers is much more expensive than retaining existing ones. The results from the data should be very valuable for management and marketing departments of telecommunication companies worldwide.

The data is preloaded into the Snowflake Server, so this is only a demonstration example.

> ***Note:** Importing data into Snowflake will not be covered in this book.

In Tableau Prep Builder under the *To a Server* section, we can see that a lot of data server connections are offered. For the Data Server connection, there are specific drivers that need to be downloaded and installed beforehand. In this case, we are using Snowflake ODBC driver version 2.16.8.

We can choose Snowflake from the Connections menu. We will need to fill in the requested information to connect to the Snowflake Server. After we sign in, we will need to select the appropriate Warehouse, Database, and Schema in order for the tables to be displayed.

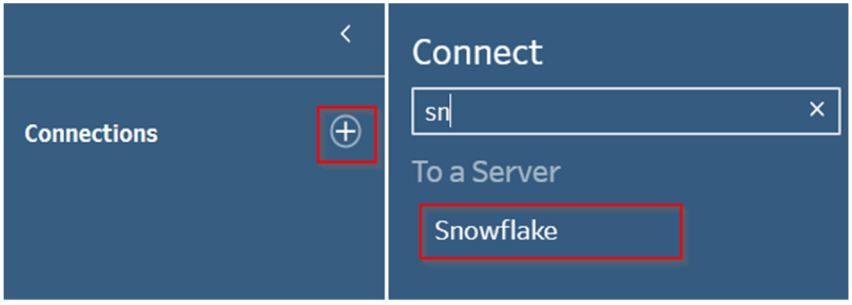


Figure 2.20 - Connection to Snowflake Server

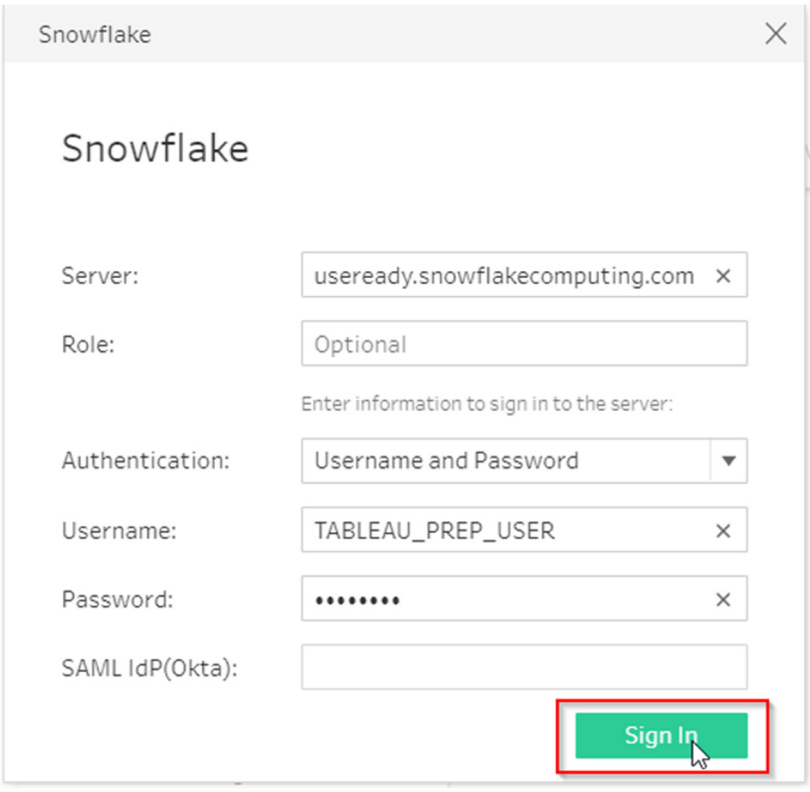


Figure 2.21 - Connection to Snowflake – Connection details

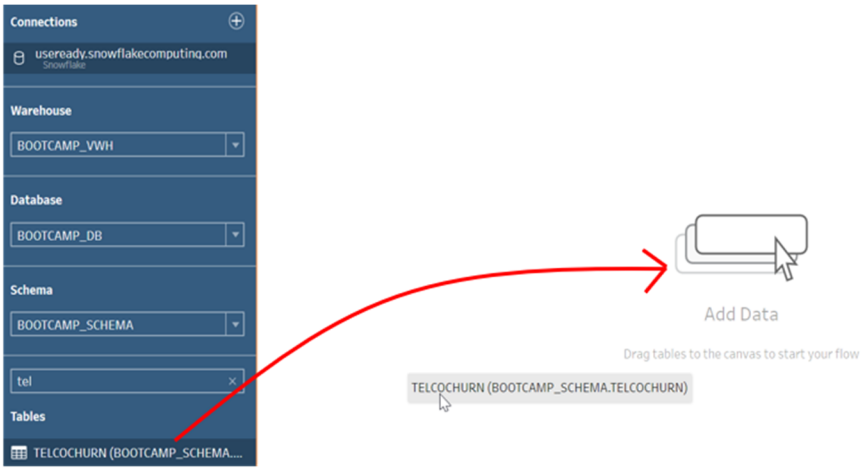


Figure 2.22 - Connection to Snowflake – Select the data

Now the data set is imported to Tableau Prep Builder.

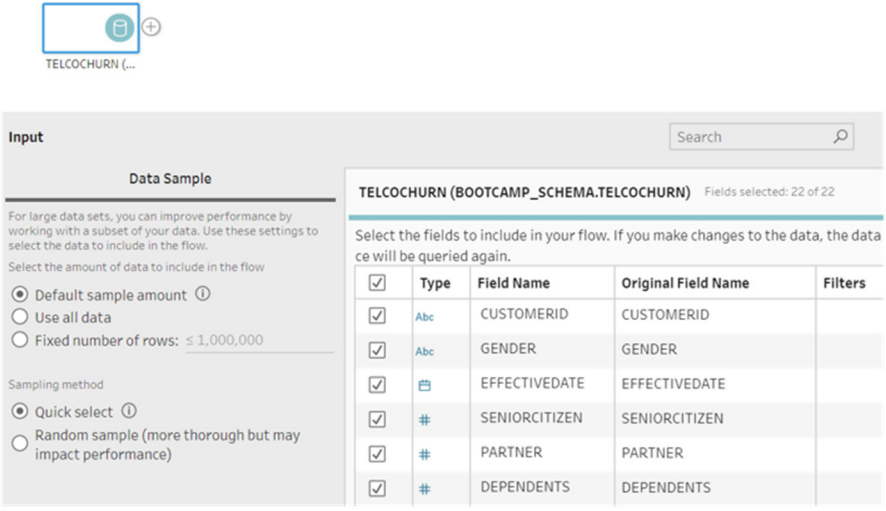


Figure 2.23 - Connection to Snowflake – Configuration Window

There is also a Custom SQL option to import selective data. We can drag the Custom SQL option to the workflow.

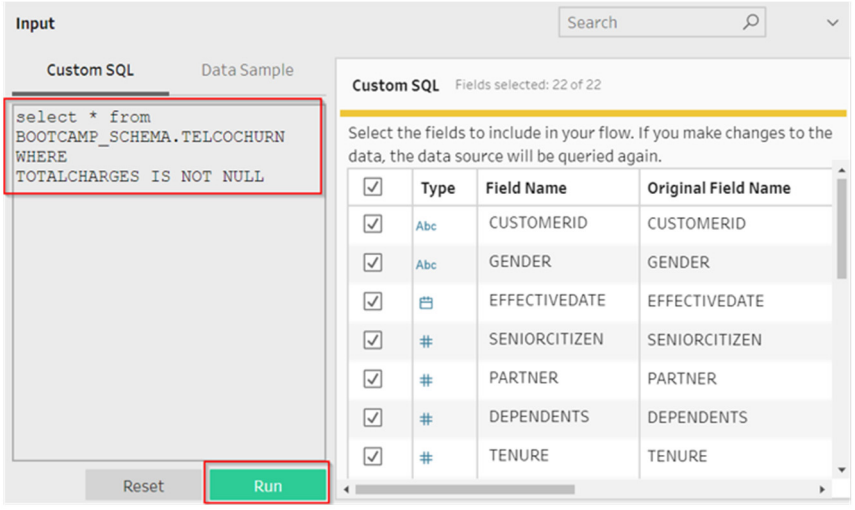


Figure 2.24 - Connection to Snowflake – Using Custom SQL

Further exploration of this data set will be covered in Chapter 3, Data Cleaning.

2.8 Connect to Microsoft SQL Server

Since Microsoft SQL Server is one of the most widely used data servers for storage, we have included an example. The data used is called *Air Passenger Survey Data*, an Airline Passenger Survey sample data that captures the details of its customers' trips and reviews. The data contains Passenger ID, Age Group, Airport, Survey Date, Travel Class, and Ratings for Facilities in terms of ease of booking, food and snacks, entertainment, cabin crew, and so on. The ratings are in the range of 1 (very dissatisfied) to 5 (very satisfied). It is important to analyze the data to understand customer satisfaction and areas of improvement.

The data is already loaded into the SQL Server will be imported to Tableau Prep Builder.

> ***Note:** Importing data into the SQL Server will not be covered in this book.

On Tableau Prep Builder under the *To a Server* section, we can see that Tableau Prep Builder offers a lot of data server connections. When Tableau Desktop is installed, Microsoft SQL Server drivers are automatically installed. For this example, we are connecting to Microsoft SQL server.

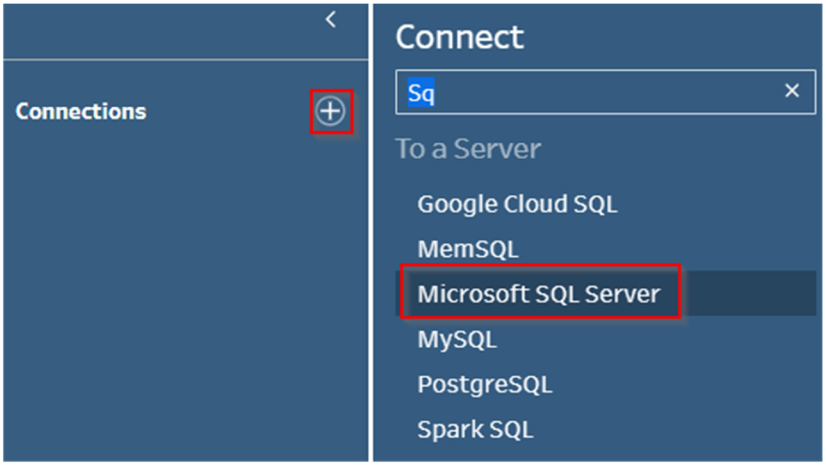


Figure 2.25 - Connection to Microsoft SQL Server

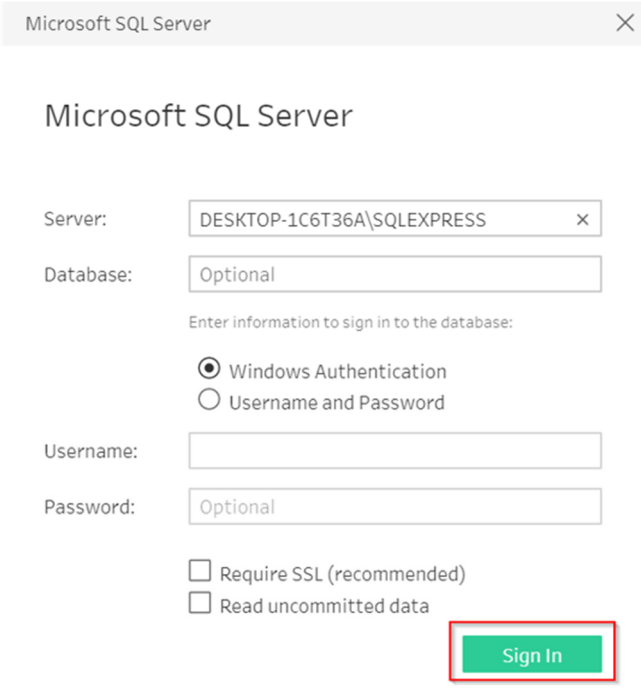


Figure 2.26 - Connection to SQL Server – Connection details

After filling in the required login credentials, choose the appropriate database, enter the table name in the search box, and select the table or scroll down to select the required table. Then we can import the table into Tableau Prep Builder by dragging it to the workflow.

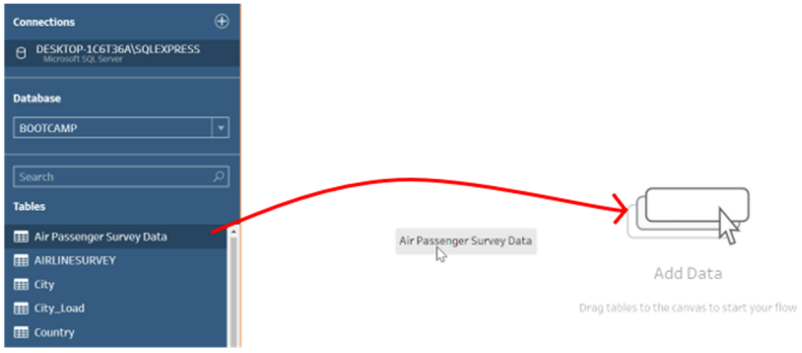


Figure 2.27 - Connection to SQL Server – Importing Data

As discussed in the previous section, Custom SQL is also available here to import selective data.

2.9 Other Features

a. **Save a Flow**

As explained in Chapter 1, data prep flow in Tableau Prep Builder is stored in .tfl or .tflx (packaged workflow) format. Data file formats that are supported for Tableau Prep Builder packaging include Text files, Statistical files, Microsoft Excel, Microsoft Access, and Tableau Extracts.

We can use the File menu and the *Save* or *Save As* option to save in the desired location.

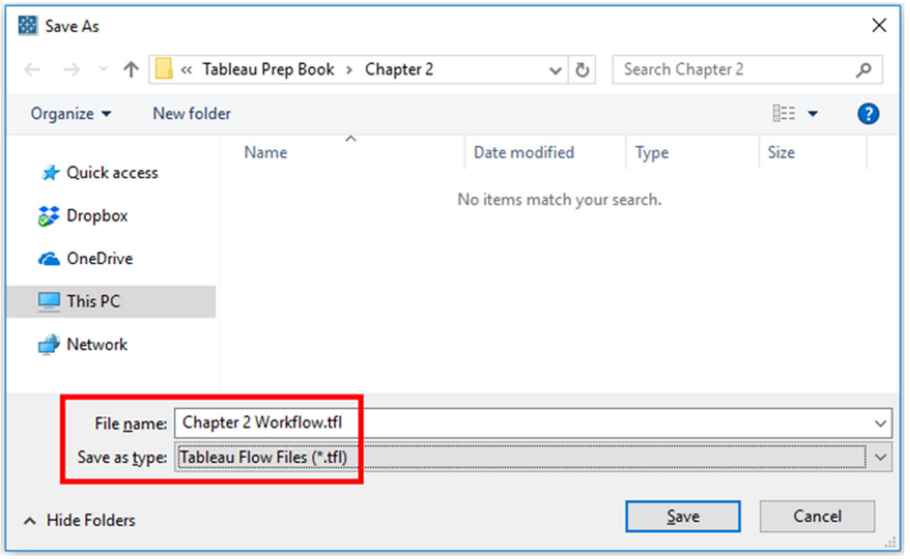


Figure 2.28 – Save Workflow as .tfl

The file can be saved as a Tableau Flow with a .tfl extension or Tableau Packaged Flow with a .tflx extension.

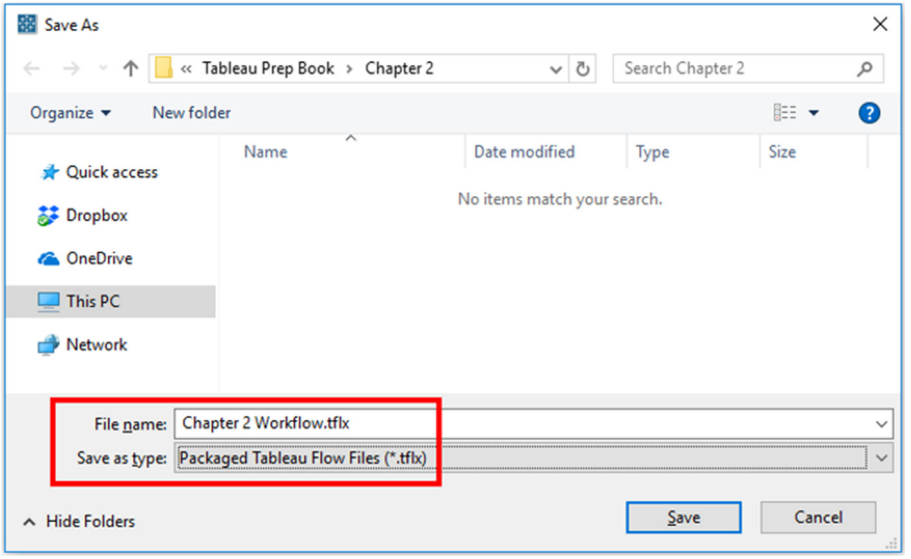


Figure 2.29 – Save Workflow as .tflx – Packaged Workflow

We can notice that the tool displays the following information when saved as a Packaged Workflow. The image shows an example in which multiple files are used in the same workflow.

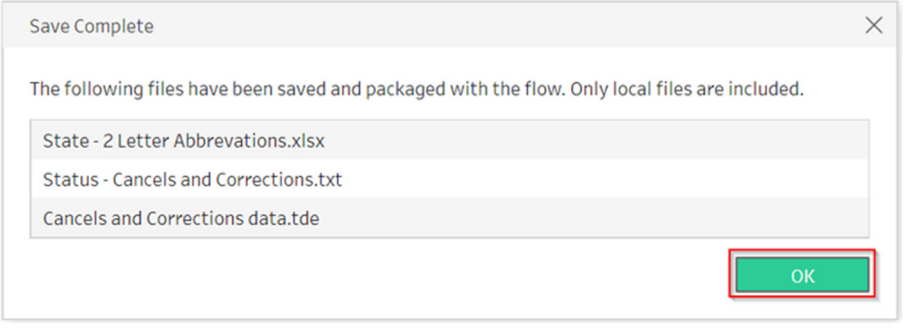


Figure 2.30 – Local Files packaged with the flow

b. **Output to File/Server**

Now we will discuss the output functionality of Tableau Prep Builder.

The file required for this example is in Chapter 2, SampleWorkflow.tflx. For now, let us open the Packaged Workflow in Tableau Prep Builder.

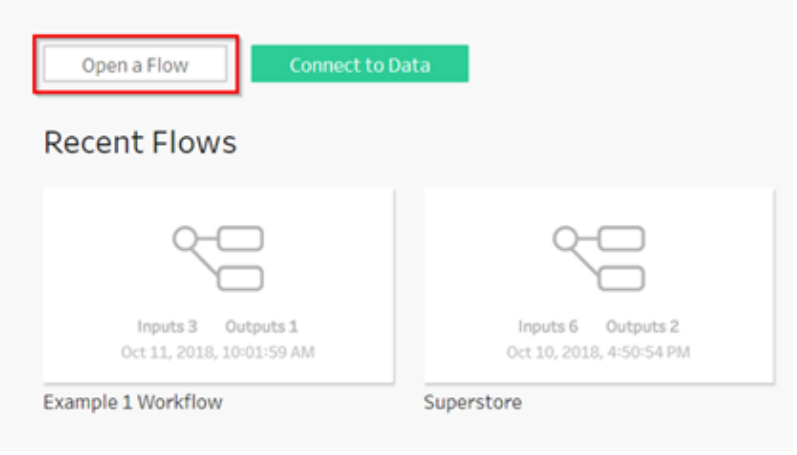


Figure 2.31 – Opening a Packaged Flow

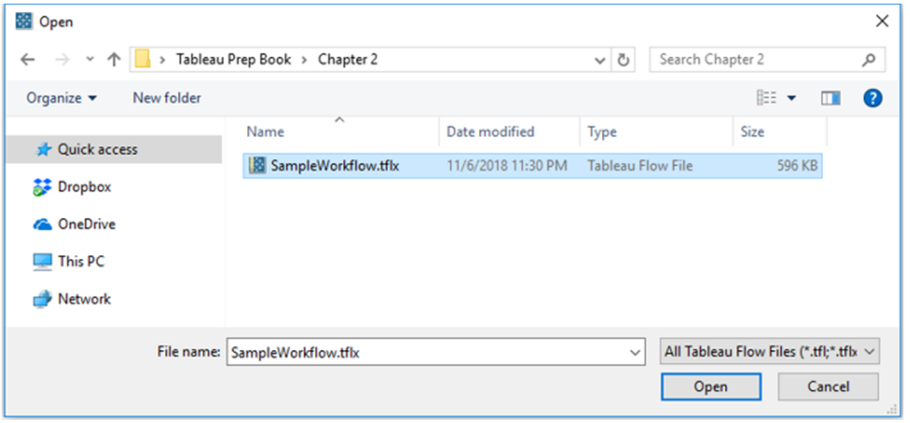


Figure 2.32 – Open a Workflow

The prep flow opens in the Flow pane, as shown below.

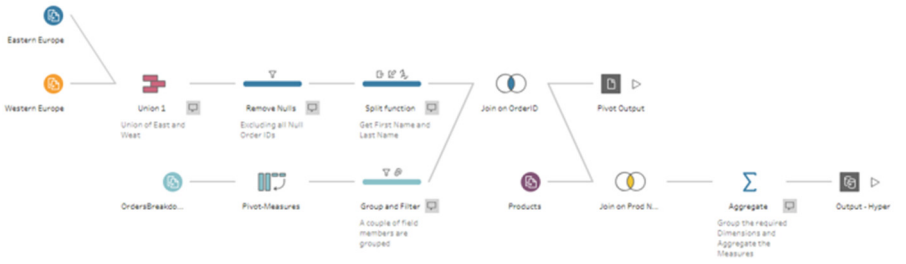


Figure 2.33 – Imported Workflow

c. Preview in Tableau

Tableau Prep Builder provides a very powerful option to preview our data in Tableau Desktop at any point in the flow, which gives us an understanding of the data quality, consistency, and shape needed for Tableau visualization.

We can right-click on the step where we want to view the data and select *Preview in Tableau Desktop*. Once the flow processing completes, the data opens in Tableau Desktop.

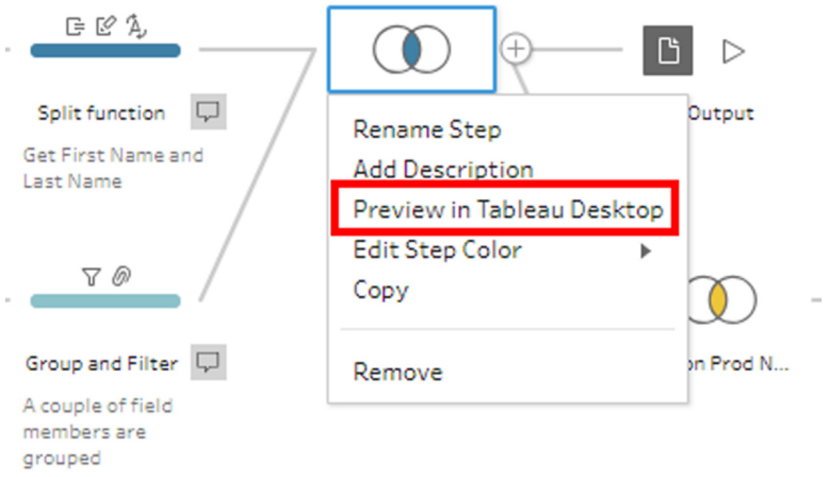


Figure 2.34 – Preview in Tableau Desktop

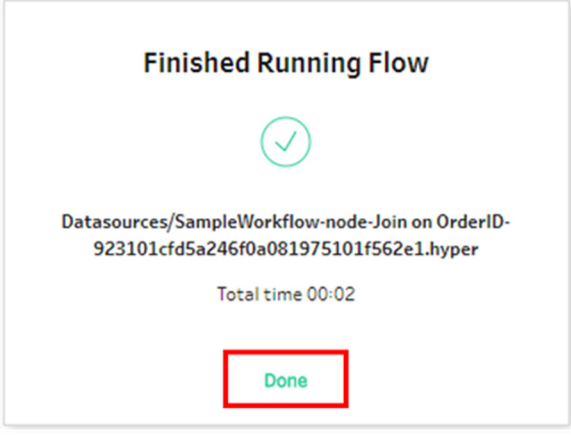


Figure 2.35 – Workflow progress before Preview

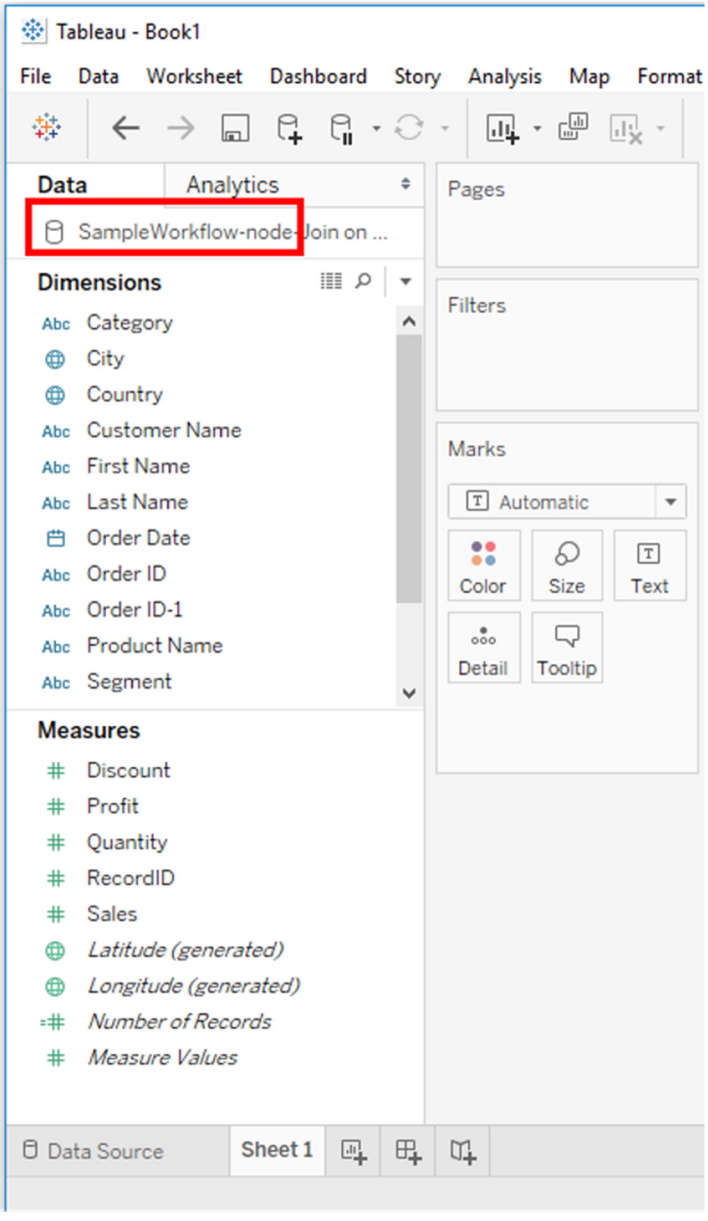


Figure 2.36 – Data ready in Tableau Desktop

Tableau Desktop opens on the *Sheet* tab directly. We can notice that the data source contains the updated dimensions and measures fields.

d. **Output Step**

Once the workflow is ready after all the steps, we can add the *Output* step. We can click on the + of the final step and select *Add Output*. The output step can be added at any step.

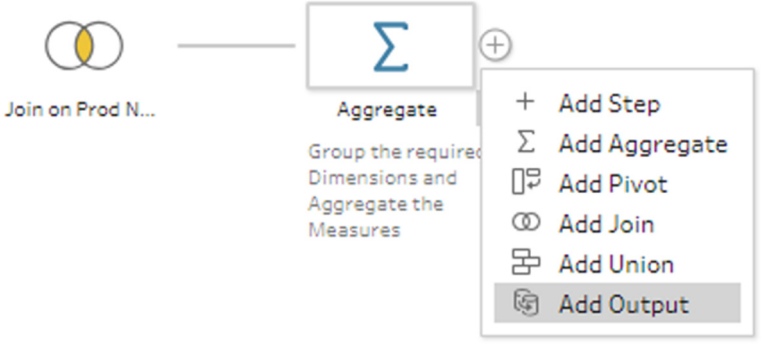


Figure 2.37 – Adding Output Step

The following screen appears when we select the *Output* step, which shows the preview of the output. If the output is not the expected one, we can go back and add more steps or perform more repairs. It also gives us the option to save the output file.

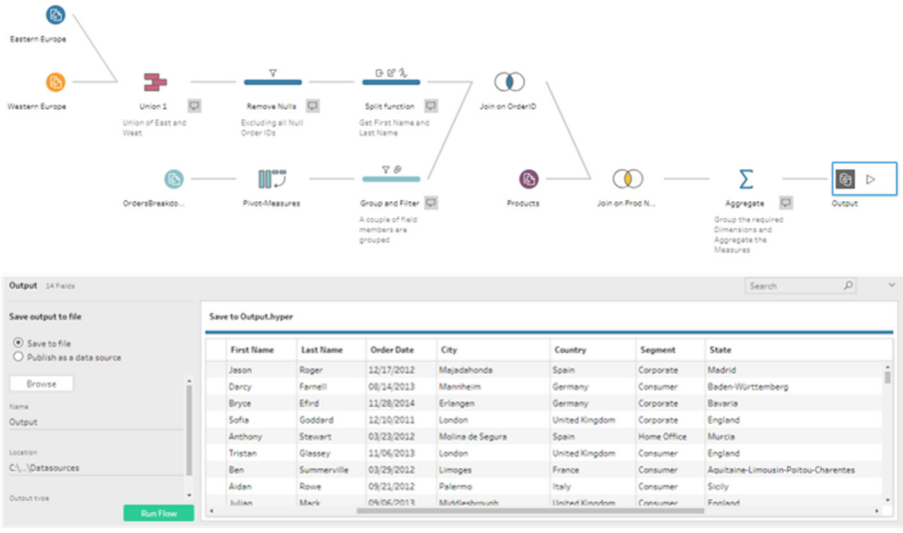


Figure 2.38 – Output preview and saving file

At this point, Tableau Prep Builder supports the creation of only three types of file extracts explained below:

i. *Comma Separated Values (.csv)* – This is ideal for sharing with external teams or third parties. The file will be encoded in UTF-8 with BOM.

ii. *Tableau Data Extract (.tde)* – This is supported by Tableau Desktop/Tableau Server versions 8, 9, and 10.0–10.4.

iii. *Hyper Extract (.hyper)* – This is available only with Tableau Server/Tableau Desktop 10.5 and later.

To create an extract file of Tableau Prep Builder flow, we must run the flow at least once so the latest changes are reflected in the data. Similar to saving any file, we can save the output in the required destination.

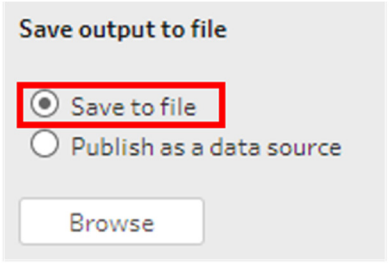


Figure 2.39 – Saving the output file

We can navigate to the folder where we need to save the file, name the file accordingly, and click *Accept*.

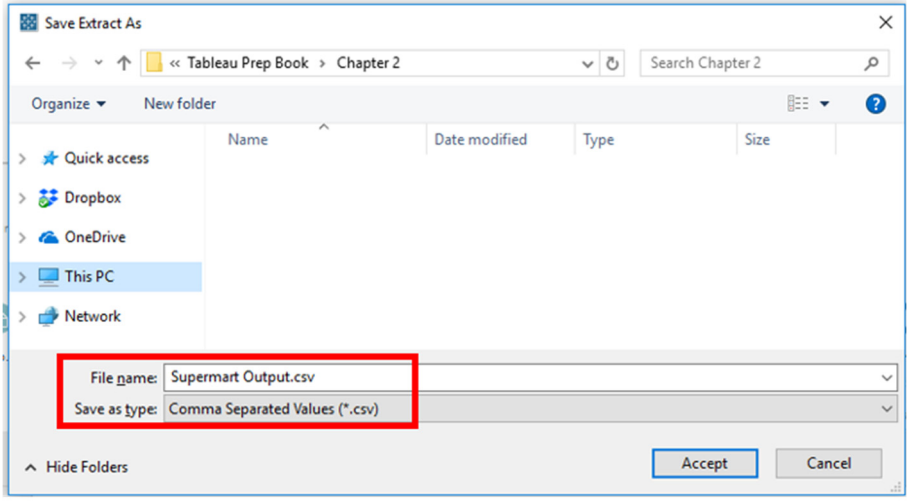


Figure 2.40 – Saving the output

Upon selecting *Run Flow*, the .csv file will be saved in the selected location.

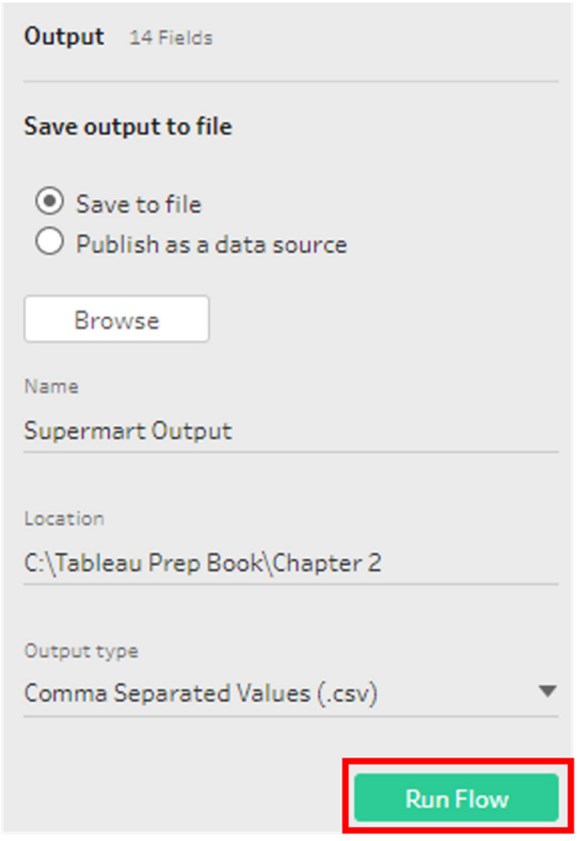


Figure 2.41 – Running the workflow to generate the output

Similarly, Output Type can be saved as an Extract (.tde or .hyper). Publishing to Tableau Server or Tableau Online will be covered in Chapter 7.

CHAPTER 3
DATA CLEANING

Customizing data is one of the most important features of data visualization. Although Tableau Desktop has very powerful features of data repair functions, Tableau Prep Builder makes it easier to customize data and prepare clutter-free data for rapid analysis. It also helps improve performance and get optimized data extracts.

As introduced in Chapter 2, we will use the *Telco Churn Dataset.xlsx* found in the Chapter 3 folder. This chapter will deal with most of the important features available in the *Cleaning* step.

As we have already seen, customer churn is a major problem in the telecommunications industry. Several studies have shown that gaining new customers costs much more than retaining them. So, a service provider has to focus more on developing accurate and reliable predictive models in order to identify potential customers who may leave in the near future. The results from the data should be extremely valuable for management and marketing departments of all telecommunication companies.

In Tableau Prep Builder, it is always recommended to first see the preview of the data in the Profile pane before performing any operation. The Input step only shows sample values and the associated data types. The Profile pane lets us see the structure of data and the composition of a particular field in the data set.

3.1 Profile Pane: An In-Depth Analysis

The Profile pane appears after connecting to data and adding a cleaning step. It basically provides a summary of the data set and

makes it easy to quickly spot the outliers. Since we use multiple cleaning steps, the Profile pane lets us view the data format at any given cleaning step in the workflow.

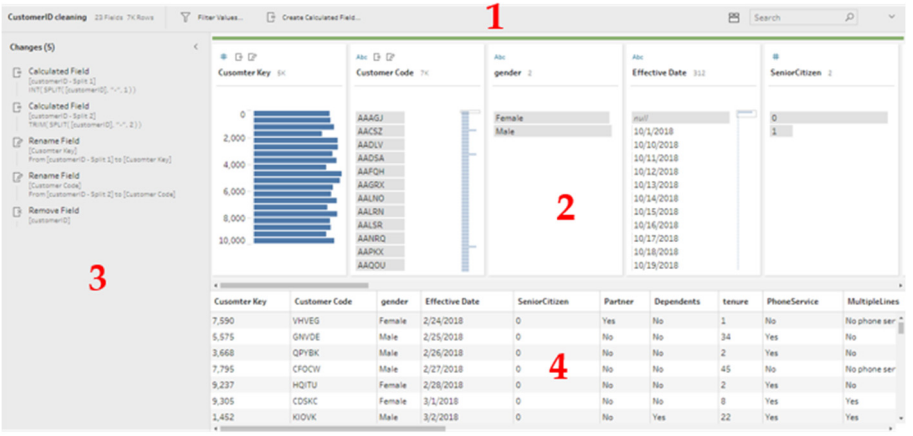


Figure 3.1 – Profile Pane snippet

The Profile pane has four sections labeled 1, 2, 3, and 4 as shown in the image. Each of them has a vital role, as discussed below.

1. **Contextual Toolbar –** As seen in the Input step, the toolbar has quick access options such as Filter Values, Create Calculated Fields, Search Fields, Hide Profile Pane, and Hide both Profile Pane and Data Grid (shown in Chapter 1). On the left side, it shows the name of the Cleaning step we are in. The quick access options change based on what is selected in the view.
2. **Profile Card –** This represents a field in the data source that can be discrete or continuous, with various types of representations in the Profile pane.

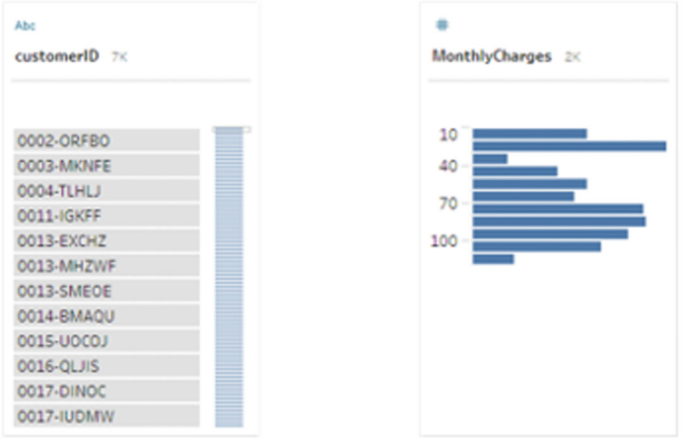


Figure 3.2 – Discrete field (left) and Continuous field (right)

For Discrete Fields, each grey bar represents a value, and the length of the bar represents the number of records with that value. The blue scroll bar to the right of the grey bar is a visual cue of overall distribution that only appears when a field has a large number of values.

For Continuous Fields, blue bars appear where data is divided into bins, represented as a histogram that shows the frequency distribution in that case. That type of visualization provides an aggregated view. If we want to view the actual values, we can change from Summary to Detail using the dropdown menu (More options - ellipsis) or right-click on the card.

We can notice that the color of the bar changes from blue to grey, and a blue scrollbar appears to the right of the grey bars, which is similar to what we noticed in Discrete Fields.

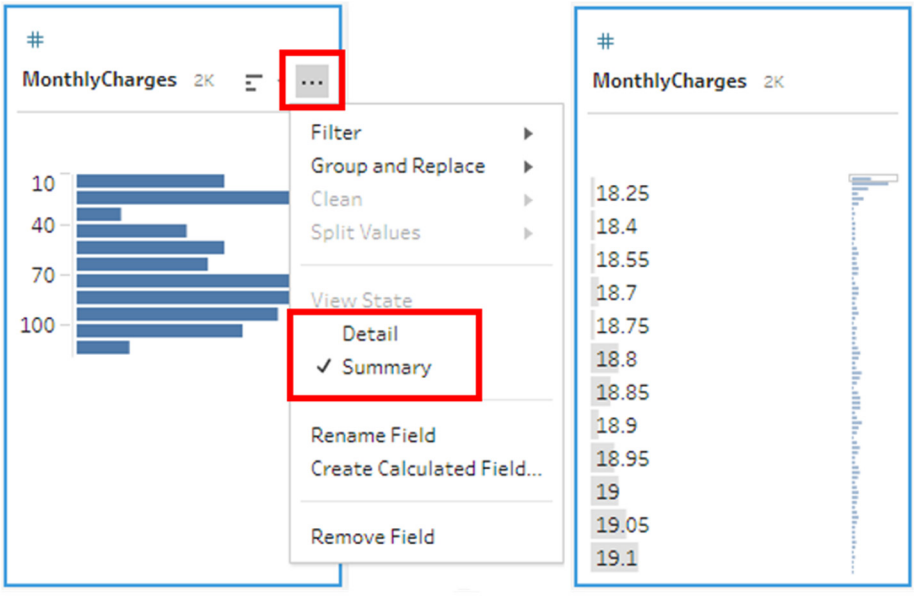


Figure 3.3 – Changing from Summary to Detail

The Profile pane also allows for Instant Analytics based on the selection. If we click on a particular bar in any of the profile cards, it highlights the portion of the other bars containing values associated with the selection from the entire data. For example, if we select *Female*, the portion associated with it in the other fields will also be highlighted.

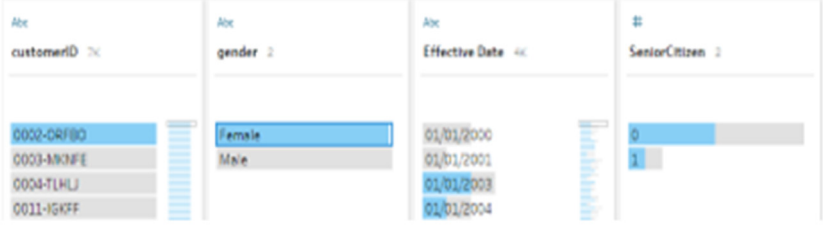


Figure 3.4 – Highlighting 'Female' in gender field

3. **Changes –** The Changes pane (similar to the changes tab in Input tool) helps us check logs and track all the changes performed. We will discuss that a little more in the section called *Changes* later in this chapter.
4. **Preview –** The Preview shows the actual data in the workflow. As we make changes to the data, the Preview gets updated.

3.2 Functions

We can see that the Profile pane lets us study the data in detail and makes it easier to apply a cleaning operation when we have a vision of the underlying data. There are various types of cleaning operations, including Filtering, Splitting Values, Renaming Fields, Removing Fields, Changing Data Types, and so on.

1. **Split Values**

In a system-generated data set, sometimes a field can have more than one piece of information. Those pieces of information can be separated into new fields that make more sense when considered individually. It is the same case when we consider the *customerID* field in the Telco Churn data set. That field is an alphanumeric field whose numeric and alphabetical parts are separated by a hyphen. The numeric part is the customer key, which is used for back-end purposes. The alphabetical part is a customer code that is similar to a customer identification code.

In the analysis, it is better if the parts are separated first and then used. That can be easily established using the split values option.

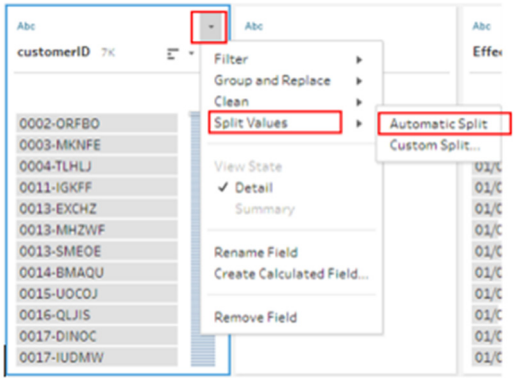


Figure 3.5 – Using Automatic Split

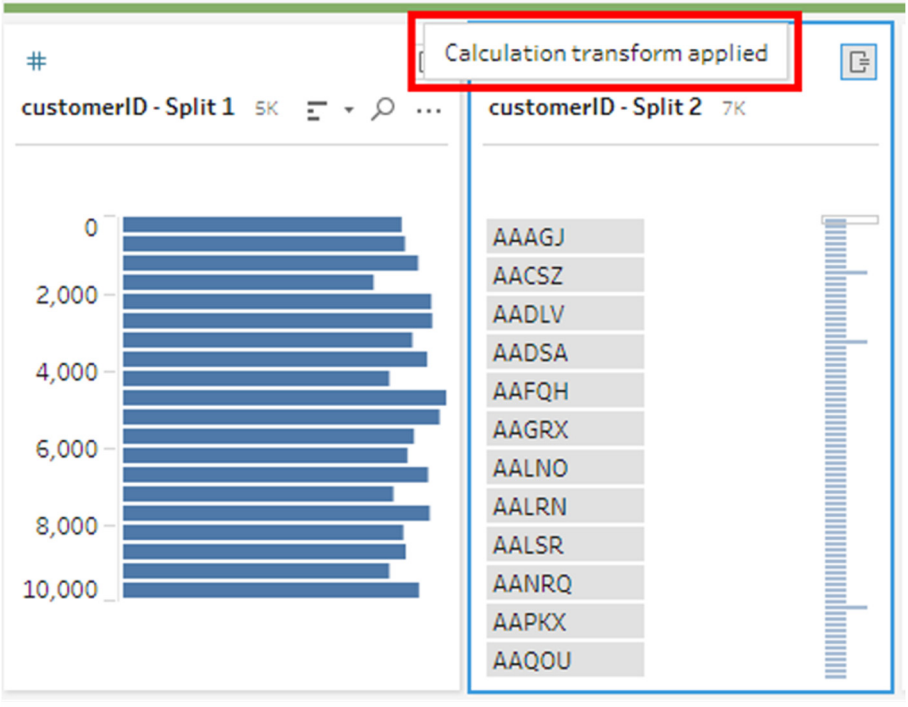


Figure 3.6 – Results of Automatic Split

Since Split 1 is a number, it shown as a continuous field.

2. **Rename Fields**

In many situations, the naming convention of fields in the database might not be easy for users to understand with visualizations. So those fields need to be renamed into something meaningful. Some fields may also have typos or be named incorrectly. In those scenarios, right-click on the card for that respective column in the Profile pane and select the *Rename Field* option. Renaming can also be done in the Input step.

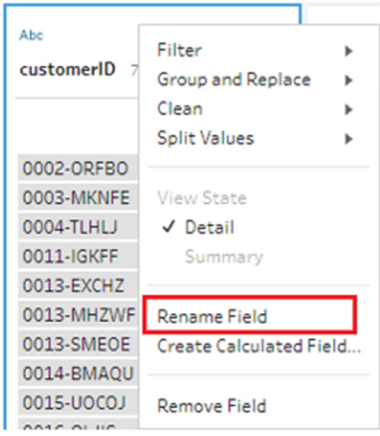


Figure 3.7 – Renaming Field

The resulting spilt fields from the *customerID* field need to be named *Customer Key* and *Customer Code*.

After renaming, a visual cue will appear at the top of the Rename Field, which acts as an identifier that this column has been renamed.

Figure 3.8 – Renamed Field

Another way to perform the same operation is to double-click on the column name and directly edit it there.

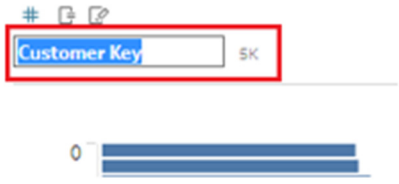


Figure 3.9 – Quick Renaming

We can also change field names by using the Contextual Toolbar. When we select a field's profile card, the *Rename Field* option opens up.

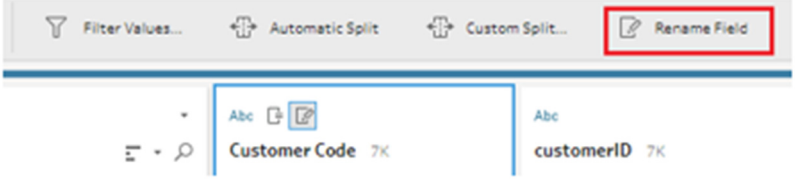


Figure 3.10 – Toolbar option for Renaming

3. **Removal of Fields**

Sometimes, the data set we are using may contain fields that are not being used at all. They occupy additional space in the memory, thus impacting performance. Removing them is one of the many simple steps towards optimization. There are two ways to do that in Tableau Prep Builder. One way is to use the

contextual toolbar. The other way is using the More options dropdown on the profile card.

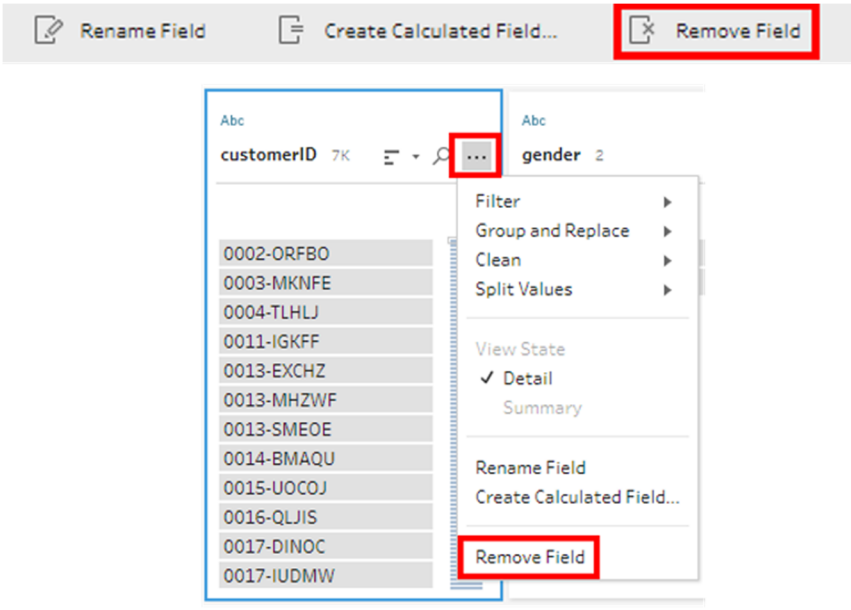


Figure 3.11a – Removing fields

Fields can also be removed using the recommendations option on the profile card.

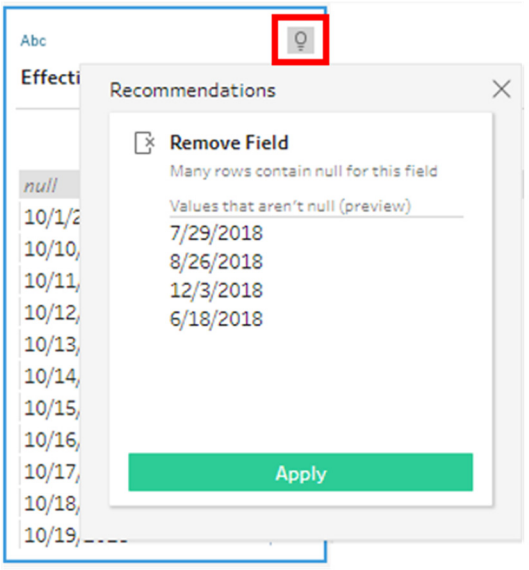


Figure 3.11b – Removing fields using Recommendations

4. **Changing Data Types**

When connected to a data source, all fields in a data source have a certain data type that reflects the information stored in that field and provides a preview of all the operations that can be applied in that field. Tableau Prep Builder supports the following five data types. Alng with that we can also specify data roles and Geographic roles if needed.

- Number (decimal)
- Number (whole)
- Date and time
- Date
- String

A field's data type is identified in the pane by one of the icons shown below:

Figure 3.12 – Data-type indicators

Most of the time, Tableau Prep Builder parses fields from the database with their default data types. Simply put, it picks *numeric* fields with a number data type, *date* fields with a date data type, and *alphanumeric* fields with a string data type. We get to see data for what they are. But sometimes, it picks a field's data type that doesn't suit it due to a misplaced punctuation or an extra character. In that case, we can manually set the field's data type just by clicking from the pre-allocated data type to a new pane from a number of options.

In this data set, it reads *Effective Date* field as a string.

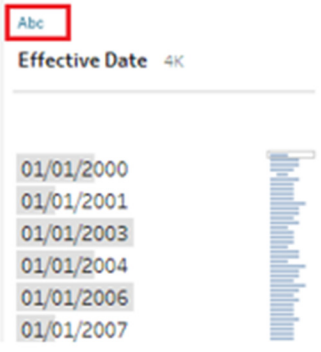


Figure 3.13 – Original Data-type of Effective Date

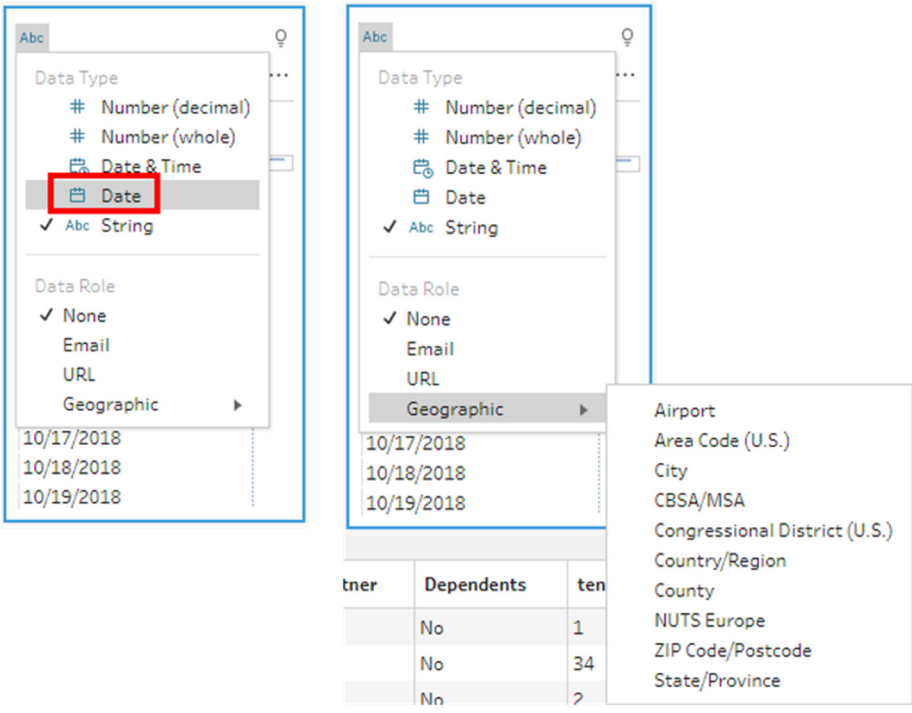


Figure 3.14 – Data Types and Roles

We need to set the data type to Date. We can also do that in the Input step.

5. **Filtering Data**

Filtering is an extremely important cleaning operation. It helps narrow down our data to relevant information only, removing all unnecessary information that can have an impact. The Input step also allows for filtering, which is discussed in Chapter 2

In the Telco Churn Dataset, the company's management wants to retain its customer base and target a particular customer group. The group includes customers from the last 24 months who have

a month-to-month contract and generate monthly revenues less than $80. The reason for selecting this group is that these customers are new and have a short-term contract with the company. Management wants those customers to opt for a long-term contract so their chance of switching companies is reduced.

The above conditions can be split into three sub-conditions:

- Contract type is Month-to-month – Dimension Filter
- MonthlyCharges (revenue) < $80 – Measure Filter
- Date range is the last 24 months – Date Filter

Dimension Filters are mostly discrete in nature, which means there are unique entries for filtering. They can have string, date (also treated as continuous when dealing with a range of dates), numeric (in case of IDs). A contract type is a dimension to which we can apply a filter.

Filters can be applied as Calculations. We can click on the *Add Filter* option  on the contextual toolbar, and the required condition will be entered in the Calculation editor window.

Figure 3.15 – Use of Calculation for discrete filtering

Filters can also be applied using the Profile card. There are two ways to use that method.

a. When the total number of values is small, we can directly select the values on which to do filtering operations. We can use the Ctrl key and select multiple values we want to use as a filter and then right-click to choose one filtering

option from Keep Only or Exclude. The Filtering operation we did in the first case can be done using this method as well.

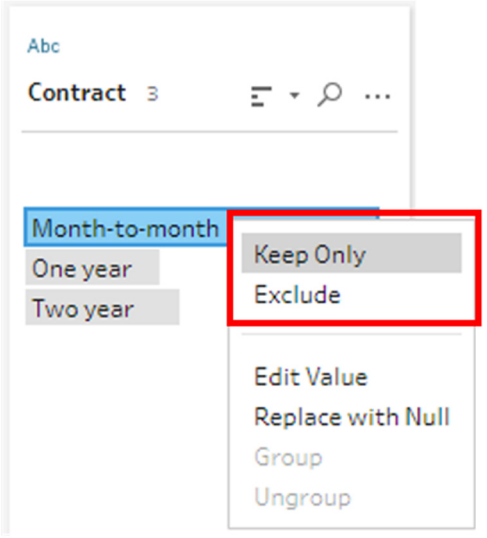


Figure 3.16 – Right-click filtering on Profile card

b. If the total number of values is large, we can go to the dropdown and select *filter*. In that case, we have to write a calculation similar to what we have discussed previously.

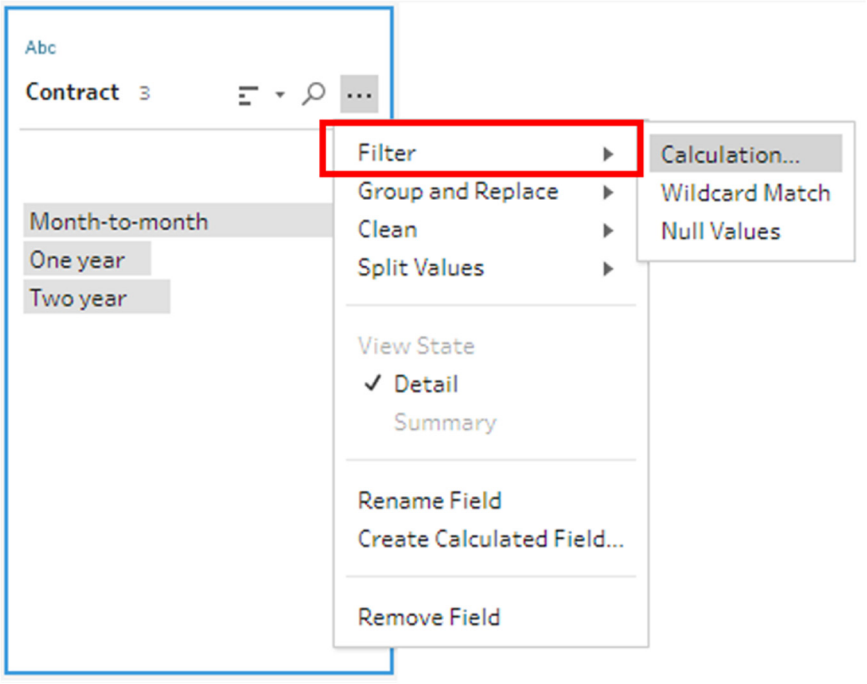


Figure 3.17 – Discrete field filtering options

Measure Filters are mainly aimed at numeric values that might be present in the data set. That function will help with quick filter operations using options such as Calculation, Range of Values, and NULL values.

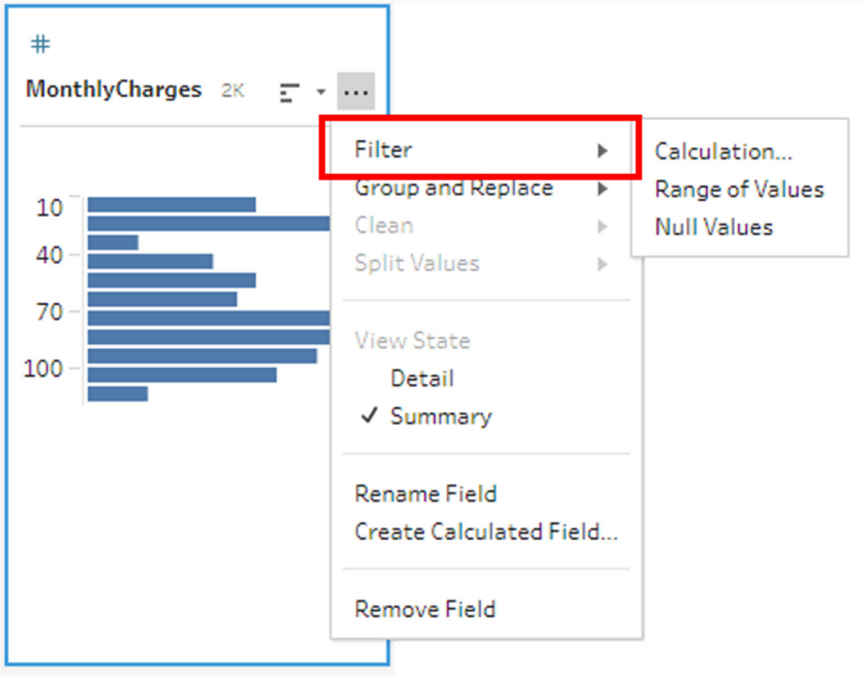


Figure 3.18 – Measure field filtering options

For this example, we need data filtered for *MonthlyCharges* below $80 since it makes more sense from a business perspective to identify low usage accounts. Using the Calculation option within Filters, we can set the condition precisely as shown below.

Figure 3.19 – Measure Filter using calculation

Using the *Range of Values* option, we can set the upper and lower limits. Using the example of the same field *MonthlyCharges*, the Range of values can be applied as a slider.

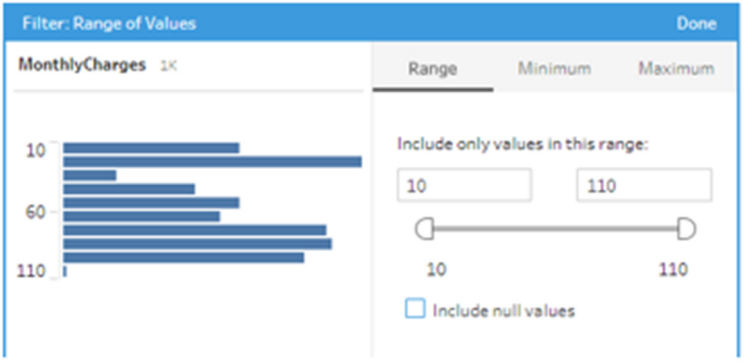


Figure 3.20 – Filter using Range

By default, Tableau Prep Builder displays the minimum and maximum values present for that measure field. The lower and upper limits can be changed as required.

The Minimum (greater than or equal to the set value) and Maximum (less than or equal to the set value) tabs in the same option can also be used to filter the data.

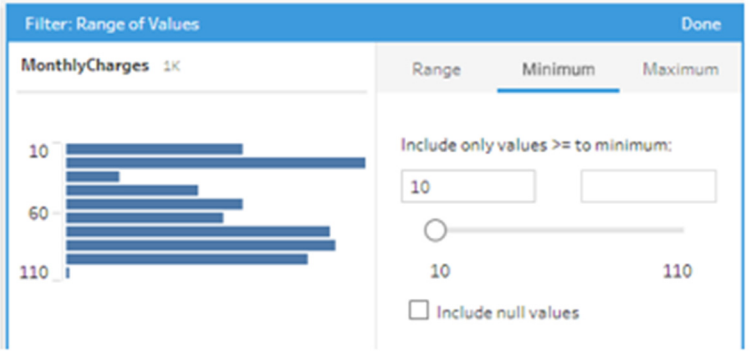


Figure 3.21 – Filter using 'Minimum'

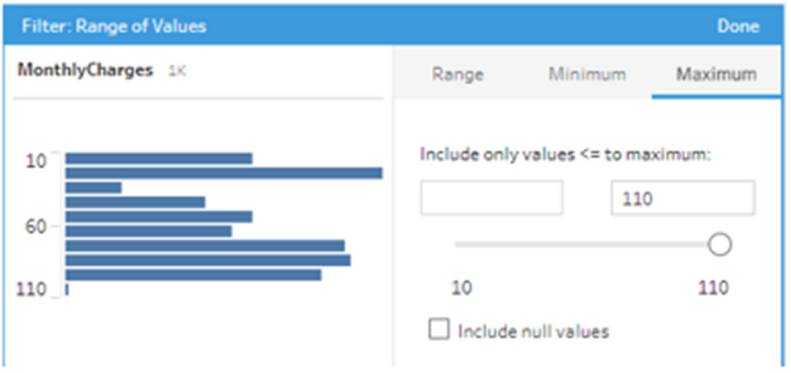


Figure 3.22 – Filter using 'Maximum'

This option also provides a checkbox a user can use to decide if they want to include the Null values or not.

Null values can also be excluded or included using the third option under the Filter function. Tableau Prep Builder has two options: *Null Values* and *Non-Null Values*. Depending on the requirement, the Tableau Prep Builder user can select either of the two options.

Dates can be filtered by three methods: Calculation, Range of Dates, and Relative Dates. We can find them by clicking on the dropdown and then going to Filter in any Date type field. In the current scenario, we must filter all the dates except those in the past 24-month period. To accomplish that, we have used all three methods to showcase how each one works.

Similar to the Dimension filters and Measure filters, a custom Calculation can be created based on whether we have a complex condition. In order to see all the customers who have an effective

date in the past 24 months, we can perform the following calculation:

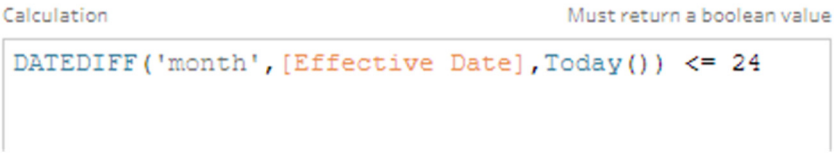


Figure 3.24 – Filter using Date Calculation

In the *Range of Dates* option, we can select both the upper and lower limits by changing the slider available. In this particular case, we have to limit the minimum date, so we will choose the minimum, click on the dropdown, and select a date that suits the condition.

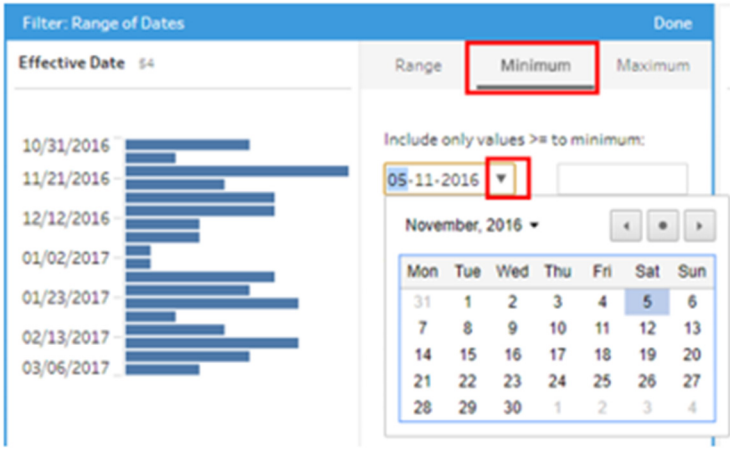


Figure 3.25 – Use of 'Range of Dates' Filter

In the *Relative Dates* option, we can select an anchor date that fixes one of the two required values to define a range. The other value is calculated based on which option a user selects.

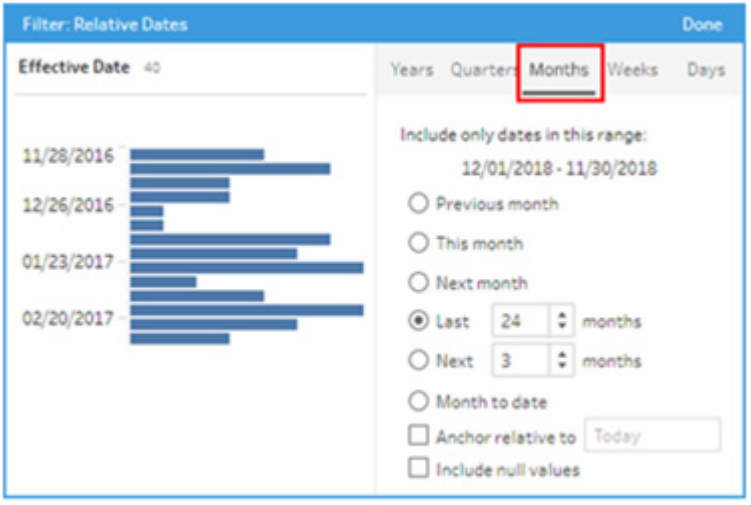


Figure 3.26 – Use of 'Relative Dates' Filter

3.3 Tracking Changes

Tableau Prep Builder provides an intuitive log of operations and manipulations implemented by a user. Selecting any of the changes will show the data as they were during the implementation of that step in the Profile pane, which comes in pretty handy. The log can be opened and expanded by clicking on *Changes* on the left side of the Profile pane.

A user can analyze the history of all the changes that have been implemented. For example, as part of data cleaning, we split the *customerID* field into two columns, which were later renamed *Customer Key* and *Customer Code*. We can see all the steps in sequence, which means any operation performed on data in Tableau Prep Builder is recorded in this log. Though all changes can be done in a single cleaning step, it is better to use multiple cleaning steps which makes it easy for debugging.

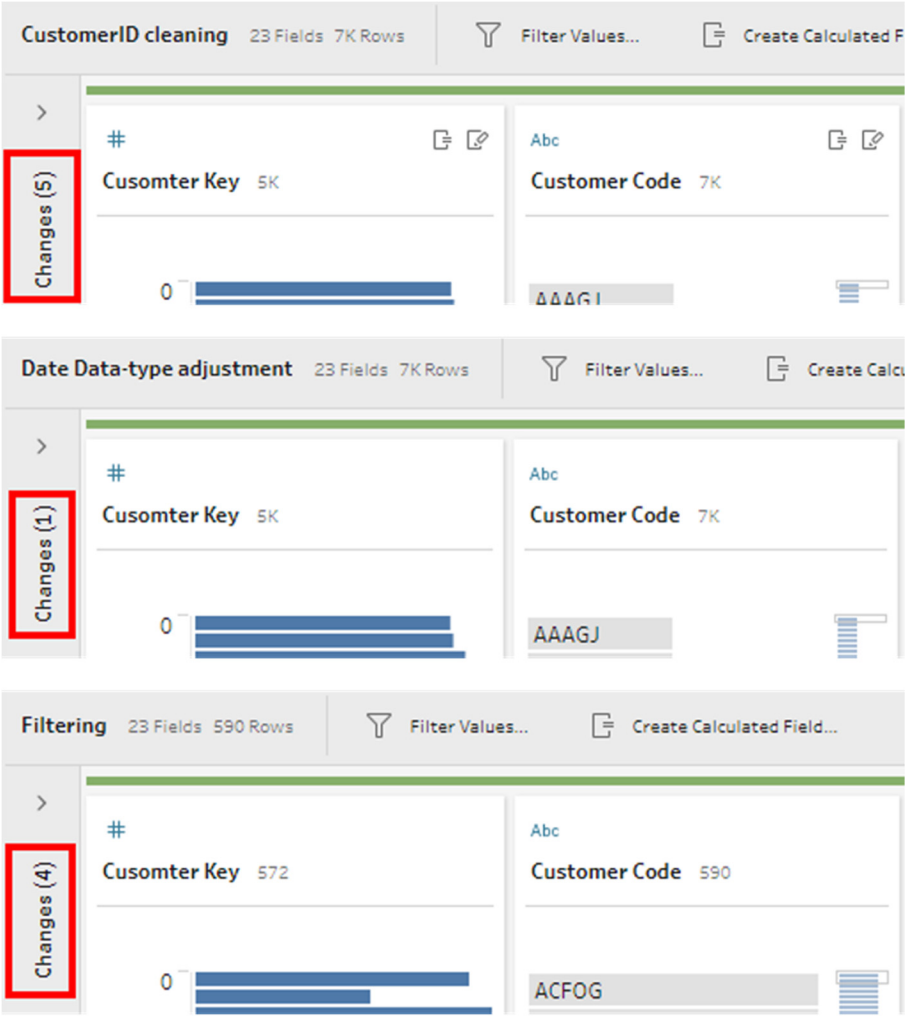


Figure 3.27 – Changes tab

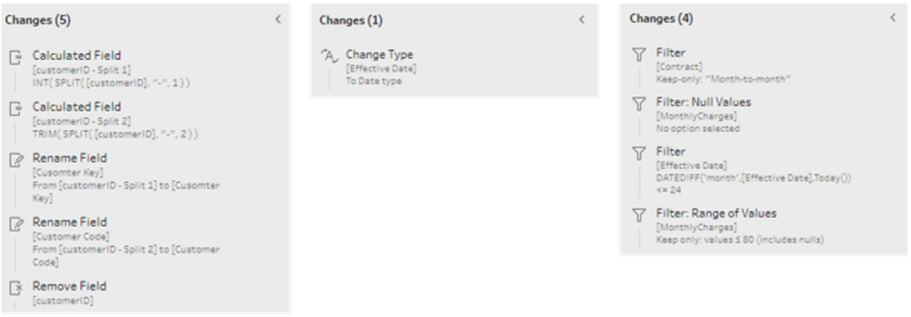


Figure 3.28 – Tracking all changes

By clicking on any of the steps, the data in the Profile pane updates and shows how the data looked when that specific operation was implemented.

> ***Note:** To see all the cleaning steps implemented, we can click on the last operation on the list.

If we want to edit a specific operation, we can use the highlighter or we can undo/delete it by clicking on the X next to that change, as highlighted in the image below. We will not lose anything that was done after that. However, if there are further calculations that are based on that field, those dependent calculations will be impacted.

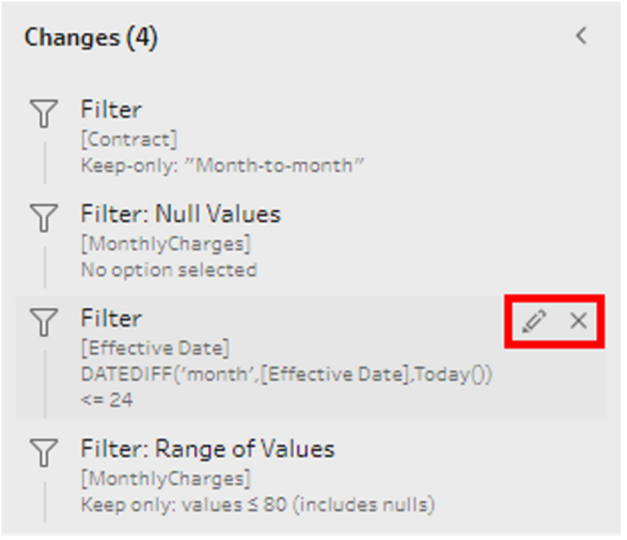


Figure 3.29 – Edit and Undo options

3.4 String Calculations

So far, we have seen examples of how we can use calculations to filter data. We used mostly Boolean calculations. But there are plenty of calculations that aren't Boolean such as manually type-casting certain fields that cannot be changed directly, as discussed in the Change data-type section, Creating Dates, or String Concatenation below.

a. **String Concatenation**

Let us suppose it will make more sense for an end user to see *MultipleLines* and *InternetService* field values next to each other since both columns contain related information. This can be achieved by creating a concatenation calculation. To create a calculated field, we can click on the field that is going to be used in the calculation and choose *Create Calculated Field*.

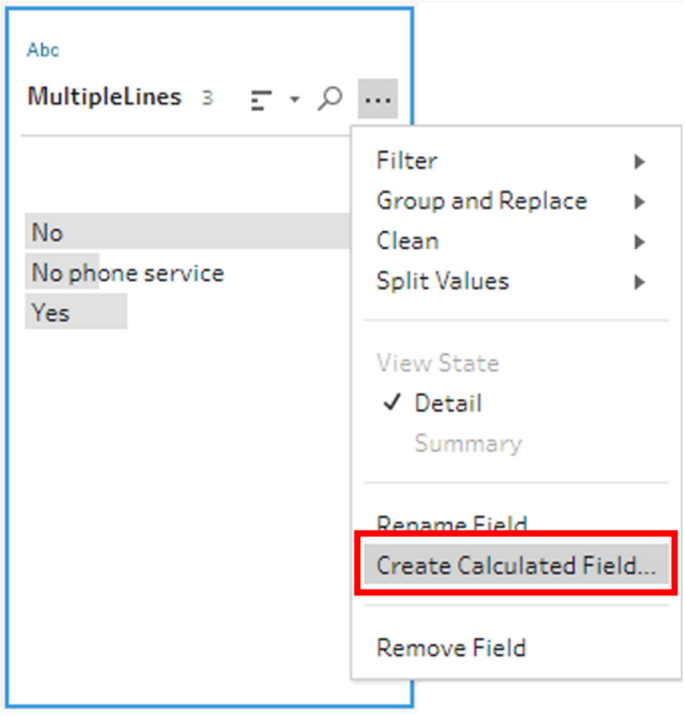


Figure 3.30 – Calculated Field

We need to concatenate two string fields here; the Tableau operator for this purpose is "+". We would also like to add a separator (-) between two values so the data can be easily distinguishable. Below is the calculation that will provide the desired outcome. Like Tableau Desktop, all special characters in Prep have to be placed in quotes.

Figure 3.31 – Create Calculated Field

We have named the newly added field *Multiple Lines – Service*. When a new field is created, it is usually the first column on the left side of the pane.

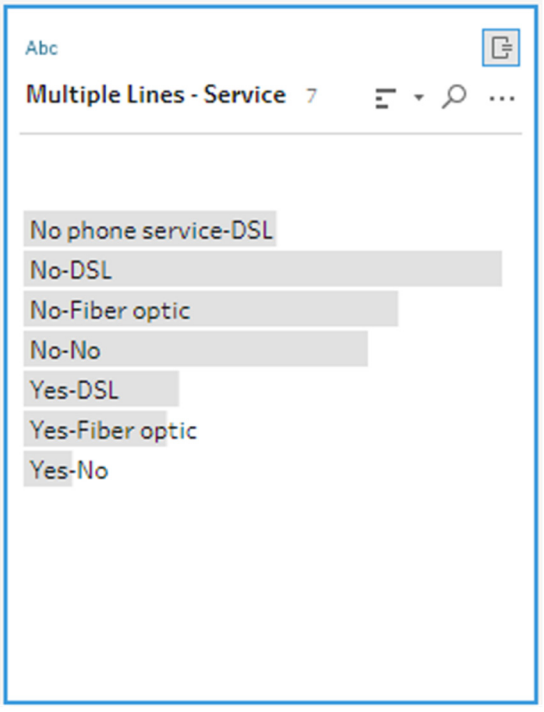


Figure 3.32 – Newly created field

b. **Split Calculation**

As already explained, the split feature is very powerful with string fields that have an identifier or delimiter that can be used to separate the string into segments. In our data set, the *customerID* constitutes an alphanumeric key. Each customer will have an associated numeric ID along with a string code.

Tableau has a built-in function called *SPLIT* that can be found on the functions section as shown below.

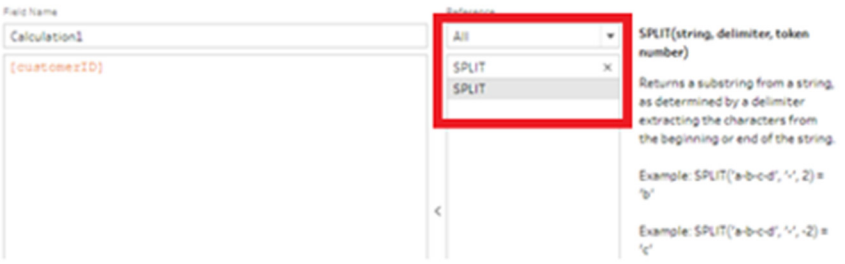


Figure 3.33 – SPLIT Function

All the functions have a brief description and an example to help us formulate the correct syntax.

Here, the separator is a hyphen (-). The separated numeric part has been named *CustomerID*.

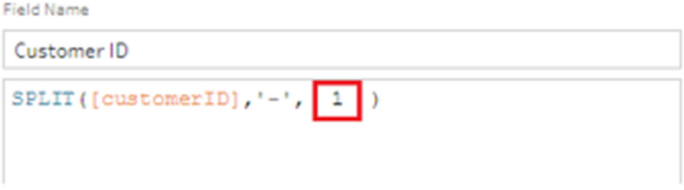


Figure 3.34 – Create Calculated Field 1

***Note:** The important aspect in the calculation is that to get the numeric part, the last argument in the above function is 1, which means it will fetch the first part of the *customerID*, which precedes the separator.

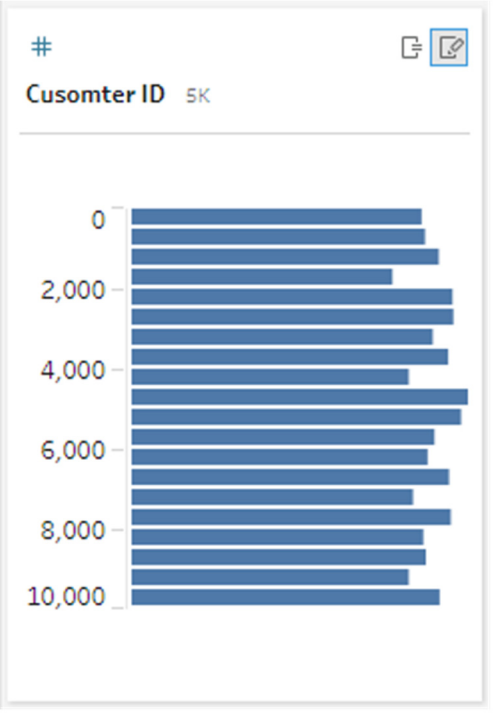


Figure 3.35 – Calculated field – Number data type

Since Customer ID field is a numeric field, we can see the histogram though it is a Discrete field.

If we change the last argument from 1 to 2, it will fetch the second half of the *customerID*. That will give us the string part of the column named *Customer Code.*

Figure 3.36 – Create Calculated Field 2

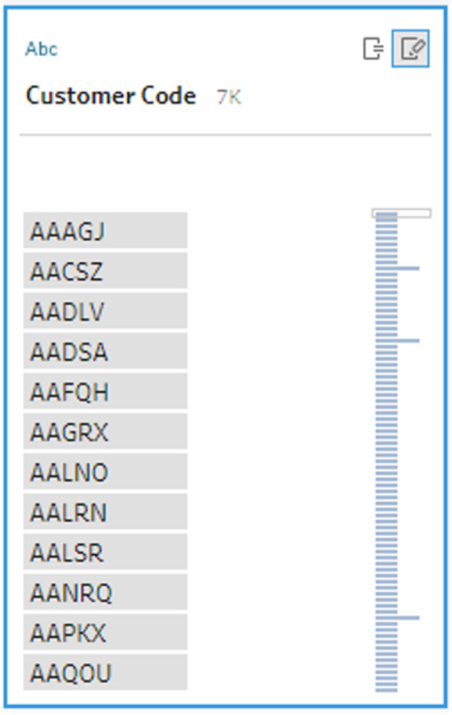


Figure 3.37 – Calculated field – String data type

After all the cleaning operations are performed, we will have the Tableau Prep Builder Flow, which has multiple cleaning steps.

Figure 3.38 – Tableau Prep Flow

We can see little annotations over each of the cleaning steps, which, when we click on them, help us easily identify the cleaning operation we performed in that particular step. To make the flow more informative, we can rename the cleaning step by

double-clicking on the label and entering an appropriate text, as discussed in Chapter 1.

Figure 3.39 – Flow with renamed steps

CHAPTER 4
GROUP and REPLACE

As discussed in the previous chapter, data cleaning can be one of the most important steps of the workflow. It can also be a labor-intensive job for any analyst. Human error will often lead to data field domains becoming needlessly large and field aggregations becoming entirely inaccurate.

Fortunately, thanks to both the automatic and manual Group and Replace functions in Tableau Prep Builder, a lot of this busywork is virtually eliminated.

4.1 Automatic Group and Replace Functions

There are three automatic Group and Replace functions in Tableau Prep Builder — *Spelling*, *Common Characters*, and *Pronunciation*. These three functions uniquely analyze the domain of a given field and automatically group similar fields together when they believe a mistake has been made. Although each of these functions are very powerful, using one of these functions is often not enough. Sometimes, we may need a combination of these.

Let us start by examining the *Spelling* and *Common Characters* functions. They are the most similar of the three algorithms. Consider this example: We have a data set with a field called *Month*. The domain consists of five members: *September, March, August, January,* and a new month named *Septembur*. Now clearly, this is a mistake that needs to be corrected.

By selecting Spelling or Common Characters, Tableau Prep Builder automatically groups these two values together and replaces every instance of *Septembur* in the dataset with *September*.

September (1)
March
August
January
Septembur (2)

Table 4.1 – Misspelled words

September
March
August
January

Table 4.2 – Grouping (1) and (2)

Once we have grouped and replaced within a field, the default value of grouped members will change to the most frequently repeating value within that group. When the group is created, a small paper clip appears next to the new group in the domain. We can edit the name of this group by right-clicking the group name in the left column and selecting *Edit Value*. Or if we believe there has been a mistake and one of the values cannot be grouped together, we can just right-click the incorrect group and select *Ungroup*.

Even though the *Spelling* and *Common Characters* functions can sometimes have similar outputs, they still have their own unique use cases and methodologies. The *Spelling* function, for instance, is best used in fields where the domain consists of actual words.

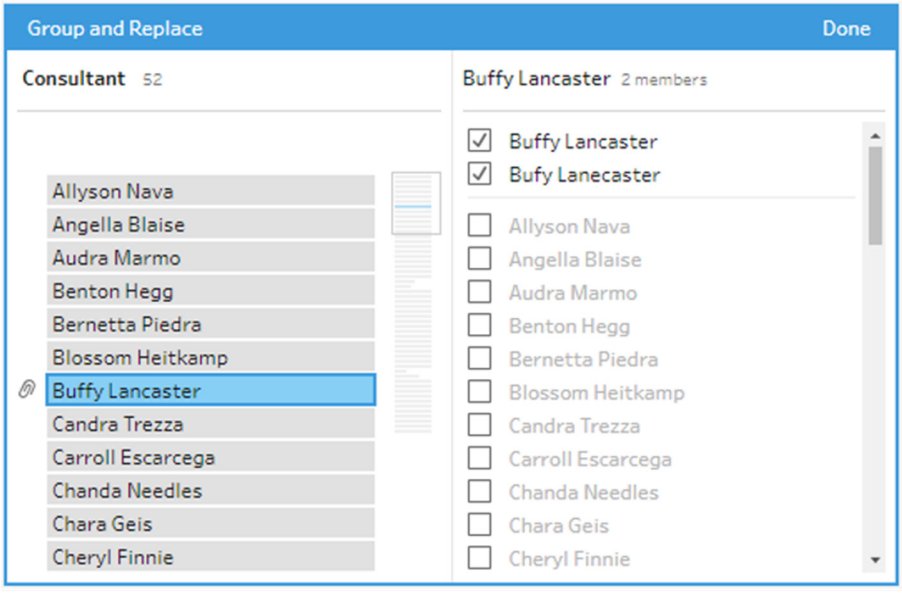


Figure 4.1 – Grouping Preview

Names, locations, categories, and other dimension fields are great examples of when to use this specific function.

Common Characters, on the other hand, is most suited for fields where the domain consists of strings that don't have a spelling but rather an order of specific characters. Its greatest utility lies in correcting ID fields, project codes, or other non-sequential character strings.

The third and final automatic grouping function is *Pronunciation*. Unlike the previous two functions, which look at the grammatical or sequential structure of domain members, this function analyzes how they are broken up phonetically. Like the Spelling

function, it is best used on fields that actually have a pronunciation. Again, names, categories, and other dimension fields are most suitable.

Now let us look at the underlying logic that Tableau Prep Builder uses to make these automatic groups. Understanding these processes will help us make better decisions on which function to use and when.

There can be situations where more than one of the above-described functions may be needed for the same field.

The *Spelling* function is based on the **Levenshtein distance** algorithm and works on all eight supported Tableau Prep Builder languages—English, French, Portuguese, Chinese, Korean, Japanese, Spanish, and German. What the algorithm does is determine which correctly spelled word has the shortest Levenshtein distance with the misspelled word. The Levenshtein distance is the number of individual characters that would either need to be inserted, substituted, or deleted from a misspelled word in order for it to match a correctly spelled word within the existing domain.

Using our September vs. Septembur example, these two words would have a Levenshtein distance of one. Since this is the lowest possible number, *Septembur* will be replaced with *September* and group the values together in the field.

The *Common Characters* function, on the other hand, uses **n-gram fingerprint** method in order to establish groups and patterns. This algorithm turns every member of a field's domain into a key of their unique characters after removing white spaces, punctuation, and so forth. Members with similar or identical keys are then grouped with one another, and the values are replaced.

The *Pronunciation* function employs the **Metaphone 3** algorithm produced and licensed by Lawrence Philips. The basic premise of the algorithm is that it breaks all words down into their phonetic groupings. It then compares the phonetic keys across all members of a domain, and similar ones are grouped together once they meet a certain threshold of commonality.

Using our Septemb*e*r vs. Septembur example, we can see how the two words have similar phonetic makeups and how the comparison between the two could be drawn.

When choosing which one of these automatic grouping functions to use, it is important to keep in mind their underlying methodologies. Which one makes the most sense in the context of a certain field? When examining the domain, which types of errors are present? In which order should the automatic functions be enacted? It may take a few tries to get the correct combination of grouping functions, but these are the types of questions we should ask ourselves in order to get on the right track.

4.2 Manual Grouping

Oftentimes, we'll find that the automatic grouping functions aren't enough to get our data set where we want it to be. They will do a great job of getting 99% of the desired groups but will require a manual touch in order to get the data ready to be analyzed. That is where Tableau Prep Builder's *Manual Grouping* feature comes in handy.

The need for manual grouping typically stems from a couple of different scenarios:

- There are too many typos to the point where none of the algorithms can establish a pattern. This is a rare occurrence but does happen.
- A pattern is established between two domain members that are meant to remain separate.

For example, A *Name* field contains *Freddy Vargas* and *Freddie Vargus* in its domain. They are two different people but could be accidentally grouped together with the Pronunciation function.

- There is a desire to combine two domain members that share little to no commonality. That can often result from formatting differences.

For example, a *Client* field contains *Dana Sherman* and *Marla Feld* in its domain. They both work for the same company called *Sutton Systems*, and we want to have their data linked together. We can then manually group these and rename them both *Sutton Systems*.

As another example, a field is supposed to be binary, but due to formatting inconsistency, there are three values instead of two. The domain includes these values: *True*, *False*, and *Yes*. There must have been a miscommunication regarding formatting, so we must now group the *True* and *Yes* together in order to rectify this. The new domain values are now *True* and *False*.

There are many reasons why someone may need to use manual grouping, but these seem to be the most common. Automatic grouping functions should normally be where we start when trying to combine domain members, but they should always be double-checked.

4.3 Example

Now we are ready to try an exercise. The *Unorganized & Associates.xlsx* file required for this example is in Chapter 4. This data set is for a fictional consulting company called *Unorganized & Associates*. As introduced in Chapter 2, the data set is about a professional services firm and its consultants which has information about Resource Utilization.

Requirement: We need to help this consulting firm clean up their data using automatic and manual Group and Replace functions.

Our first step is to connect to the data source. We need to select *Microsoft Excel* from the connections pane and navigate to the data source *Unorganized and Associates* in the Chapter 4 folder.

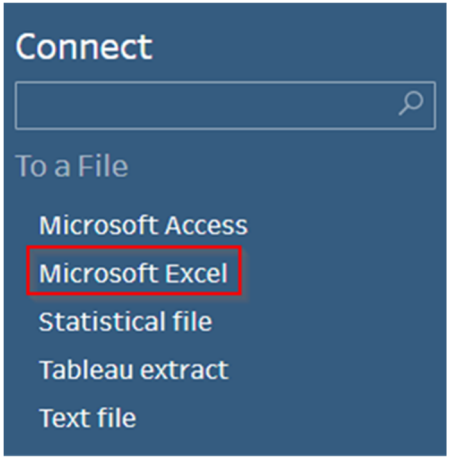


Figure 4.2 – Connections Pane

Next, we need to drag *Number of Hours* into the workflow pane from the Tables pane.

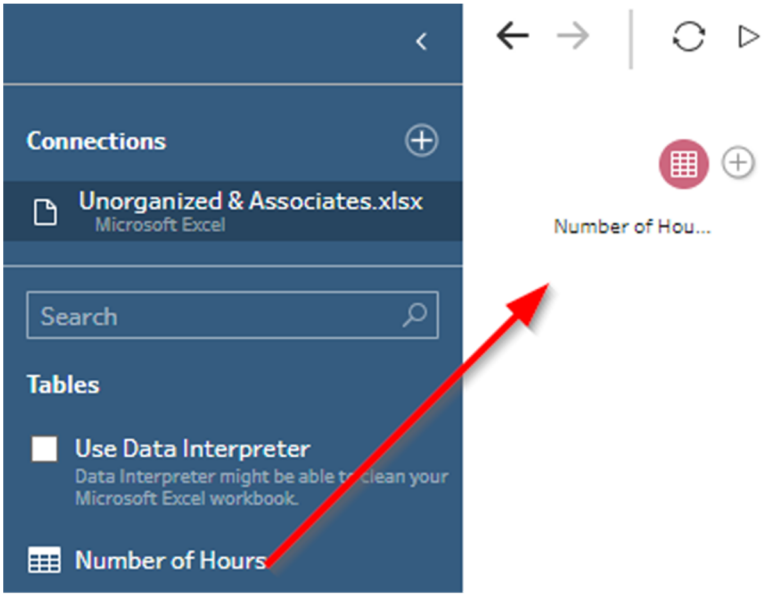


Figure 4.3 – Table selection

Then we need to add a cleaning step so we can both examine and repair the domains of our fields.

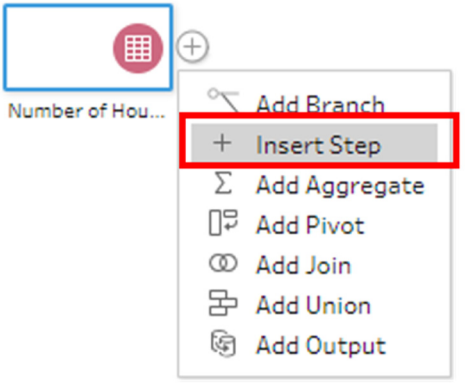


Figure 4.4 – Adding a Cleaning step

Now let us look at the domains for our *Consultant* and *Client* fields. The firm should only have 51 employees and 74 clients, so there has clearly been some inconsistency in terms of reporting. Let us see if we can get those numbers down. Let us start by trying to group the *Consultant* field with Spelling. We can click on the ellipses or More options dropdown.

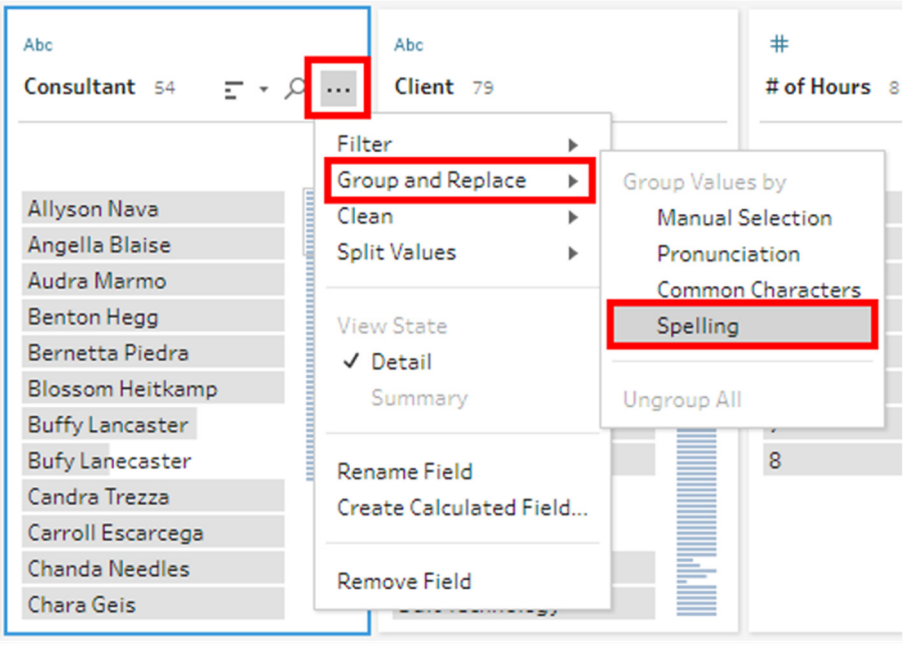


Figure 4.5 – Group on Spelling

After that is applied, the *Consultant* domain should now shrink to 52 members. We can see that Tableau Prep Builder has created two groups: *Buffy Lancaster* and *Marylin Stoddard*. As soon as the group is created, we can see the *paperclip* indicator. We will use other tools later on to trim it down further.

*__Note:__ The Changes pane as seen in Chapter 3 where all changes performed in this step are logged. As we use more Group and Replace functions, more changes will be displayed here.

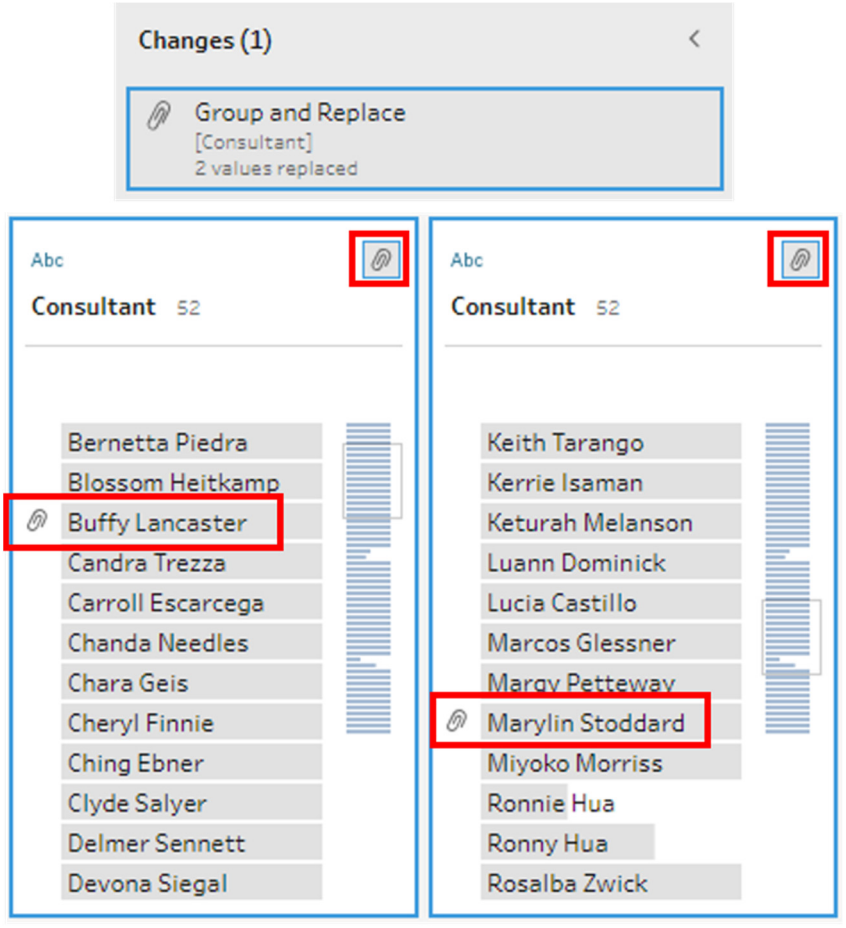


Figure 4.6 – Consultant field – grouped changes

Now let us look at the *Client* field. Again, the domain has too many members. There should only be 74 clients, but there are currently 79.

Figure 4.7 – Client field

Let us start by selecting the *Common Characters* function from the options menu and see how much the domain is reduced.

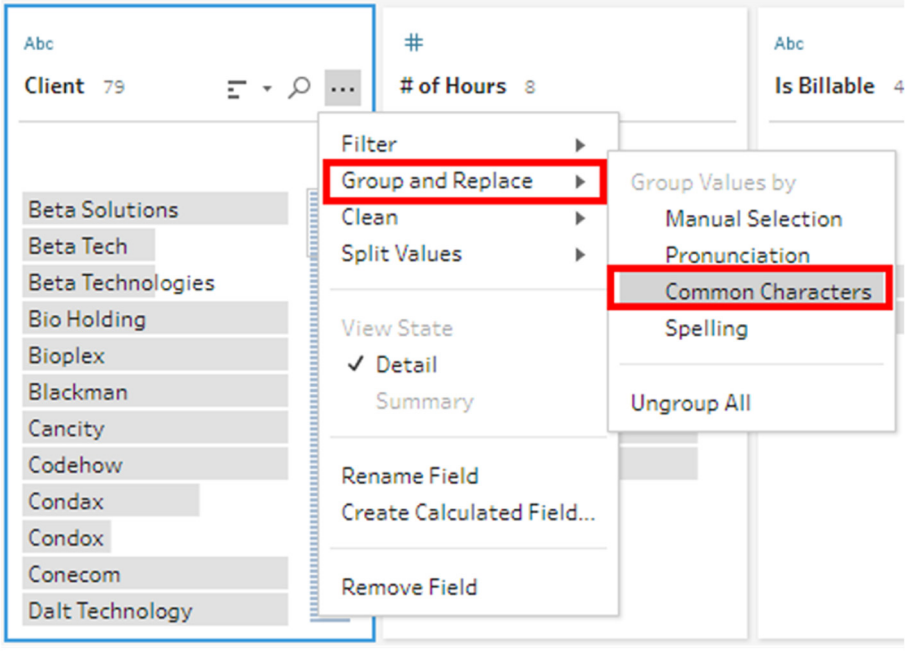


Figure 4.8 – Group on Common Characters

The domain for Clients has been reduced from 79 to 78, and a new item has been added to the Change pane highlighted below.

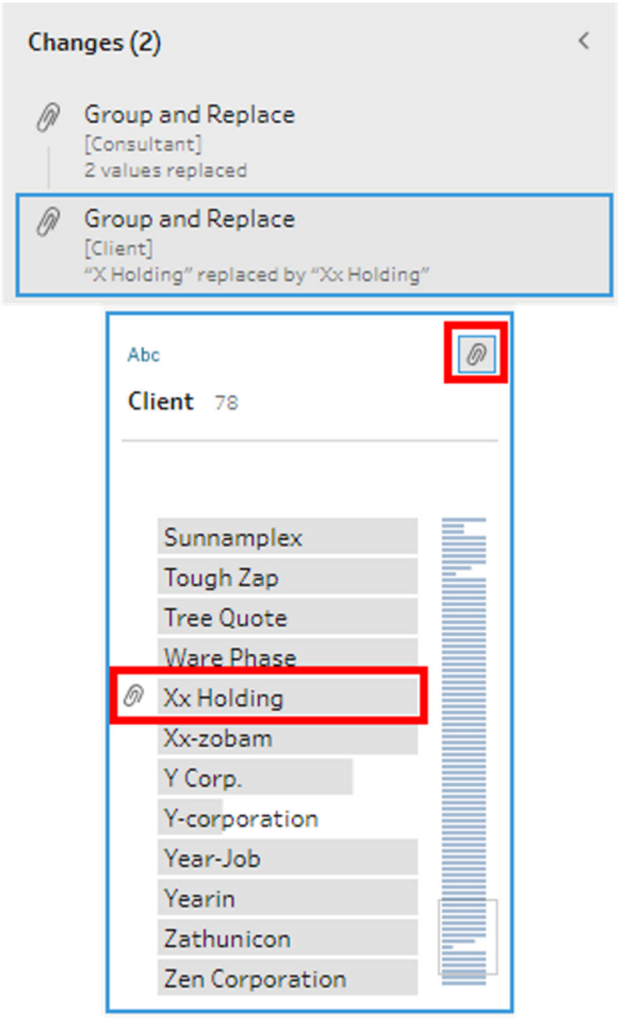


Figure 4.9 – Client field – grouped changes

If we try grouping with Spelling as well, we will reduce the domain of the field from 78 to 74 and create some new groups like Condax, Doncon etc.

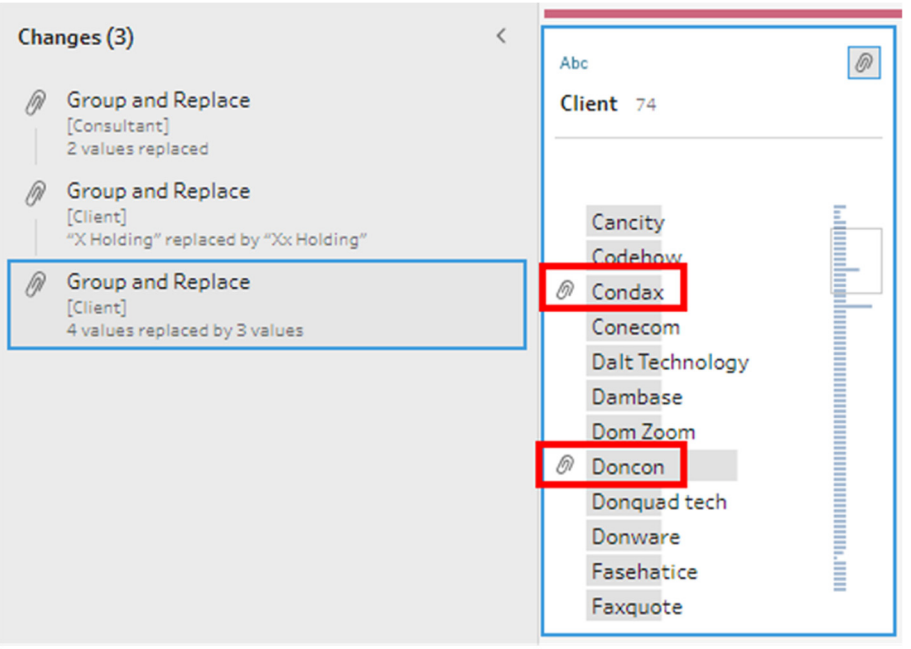


Figure 4.10 – Client field – Group on spelling

Now let us look at our next Group and Replace function.

Until now, we've only been using the Spelling and Common Characters functions to clean our data. Now let us use Pronunciation to further reduce the size of the domains. Select Pronunciation from the list of Group and Replace functions under the *Consultant* field.

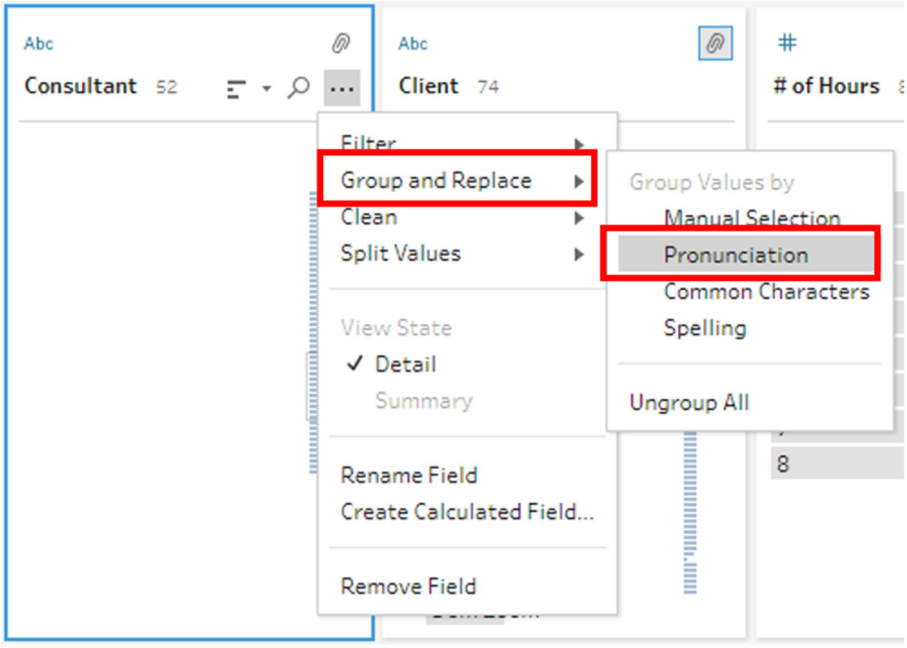


Figure 4.11 – Group on Pronunciation

After performing this action, our new domain for the *Consultants* field has been reduced from 52 members to 50. One of the groups created is *Frederick Ferrari III*. During data entry, someone wrote *Frederick Ferrari the Third* instead of *Frederick Ferrari III*. The other group, *Ronny Hua*, was created from a spelling error.

Figure 4.12 – Consultant field - Group on Pronunciation

The Pronunciation function accidentally over-reduced the size of the *Consultants* domain. Let us ungroup *Ronny Hua* in order to fix that mistake. We can right-click on the member and select the *Ungroup* option.

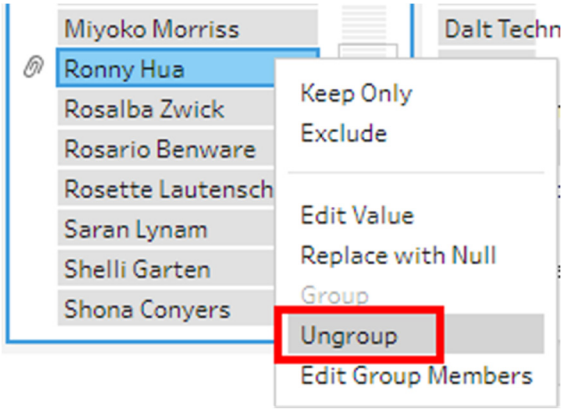


Figure 4.13 – Ungrouping

The *Consultant* field has now been fully cleaned. Let us move on to fixing the *Client* field's domain. We can perform the same Pronunciation function on this field as well.

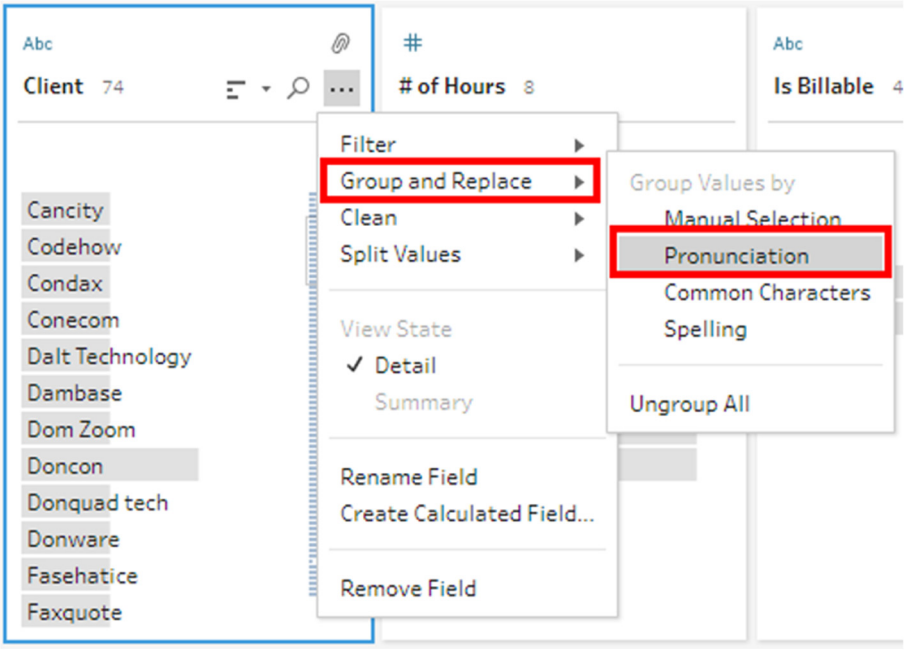


Figure 4.14 – Group on Pronunciation

After grouping we have 72 members in the *Client* domain. Two new groups have been created under the names *Y Corp.* and *Beta Technologies*, which have rectified multiple formatting issues. The *Beta Technologies* group combined *Beta Technologies* and *Beta Tech.*, while the *Y Corp.* group combined *Y Corp.* and *Y Corporation*. This is an example of how the Pronunciation function can recognize and interpret abbreviations in a data set.

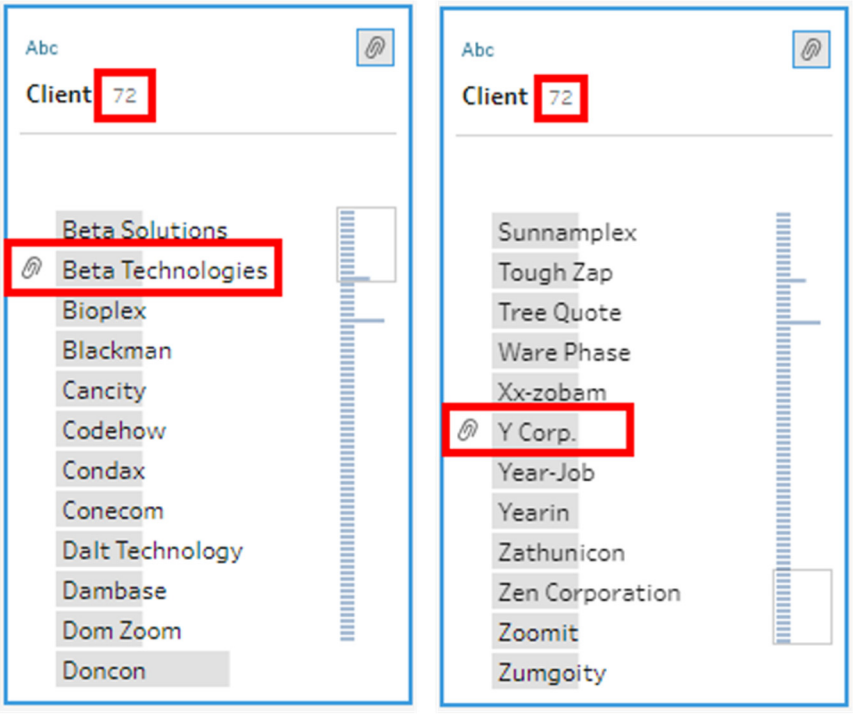


Figure 4.15 – Client field - Group on Pronunciation

This field is almost completely normalized now. We have just one more correction to make. It appears that we have two different divisions of the same company listed on the client sheet. We need to combine *Beta Solutions* and *Beta Technologies* into *Beta Corporation*. We can click the *Manual Selection* option under *Group and Replace*.

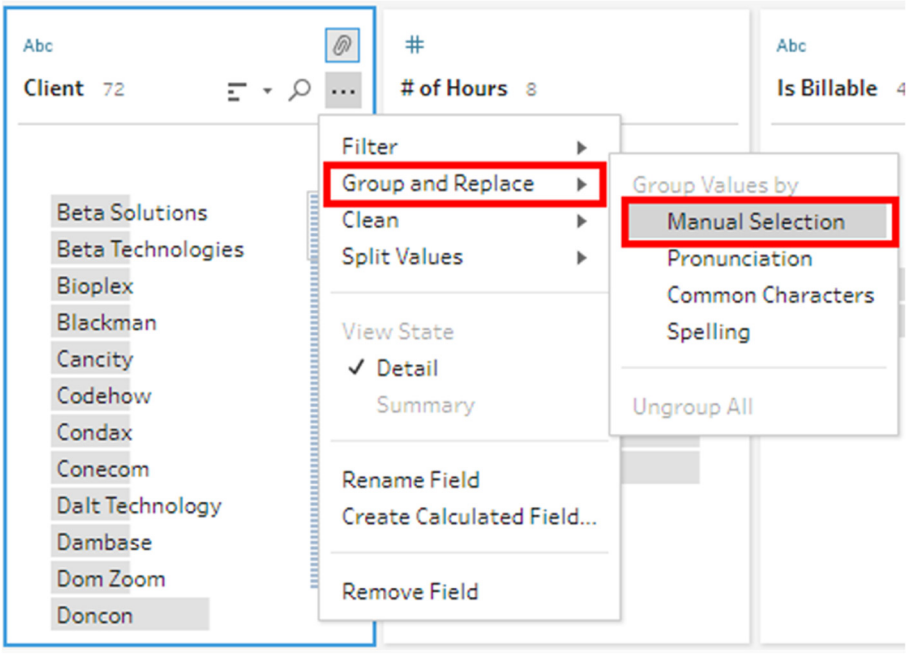


Figure 4.16 – Group on Manual Selection

Now we can select *Beta Solutions* and check the box next to *Beta Technologies* in the pane to the right. Then click *Done* in the top right corner.

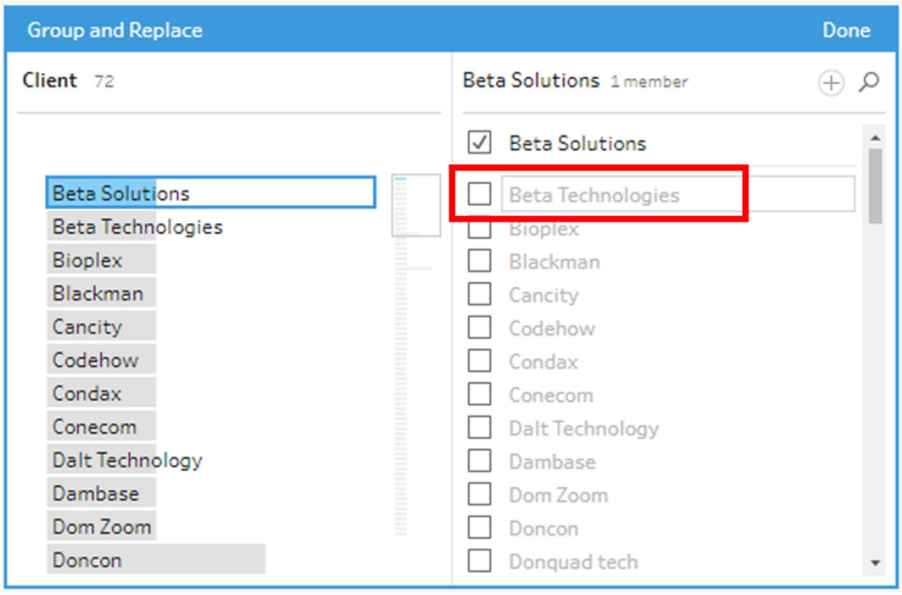


Figure 4.17 – Beta Corporation – Manual Selection

The name now needs to be changed. We can right-click on the new group we just created and select the *Edit Value* option. Type in *Beta Corporation* and press Enter.

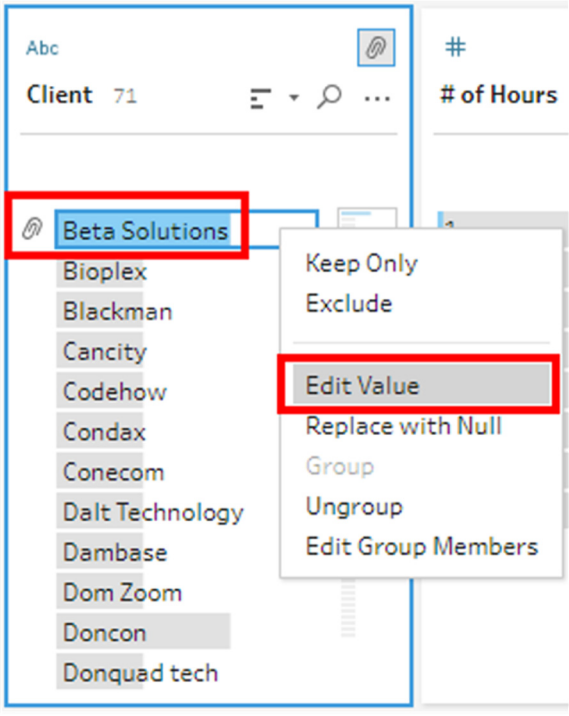


Figure 4.18 – Edit Value – Creating Alias

If we examine this change in the *settings* pane, there is now a red dot next to *Beta Corporation*. That's because this value is not present in the original data source, and Tableau Prep Builder is making us aware of that. This way any new member will get added into this group automatically when the data updates.

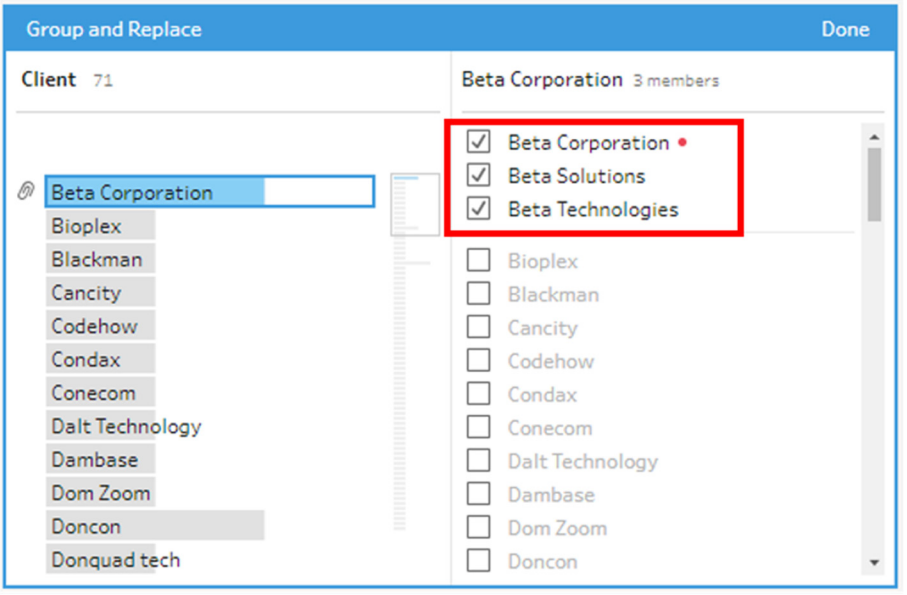


Figure 4.19 – New Value placeholder

Now the *Client* field has been completely repaired. The only field left to clean is the *Is Billable* field. That is supposed to be a binary field with the only two values in the domain being *True* and *False*. Due to a data entry error, the values *Yes* and *No* are also included in the domain. We again have to use *Manual Selection* in order to correct this mistake. Select the *Manual Selection* option from the dropdown menu.

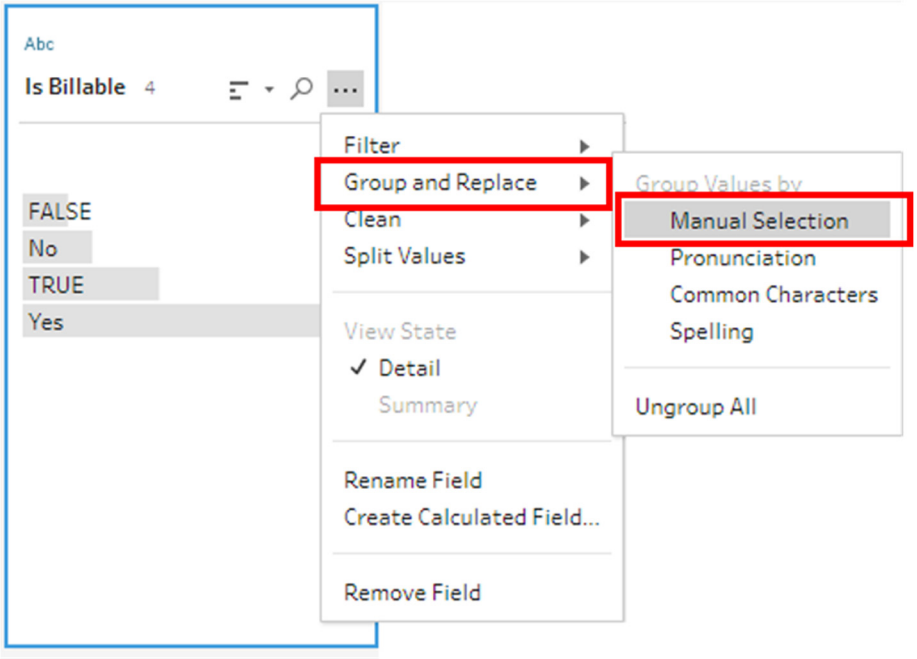


Figure 4.20 – Group on Manual Selection

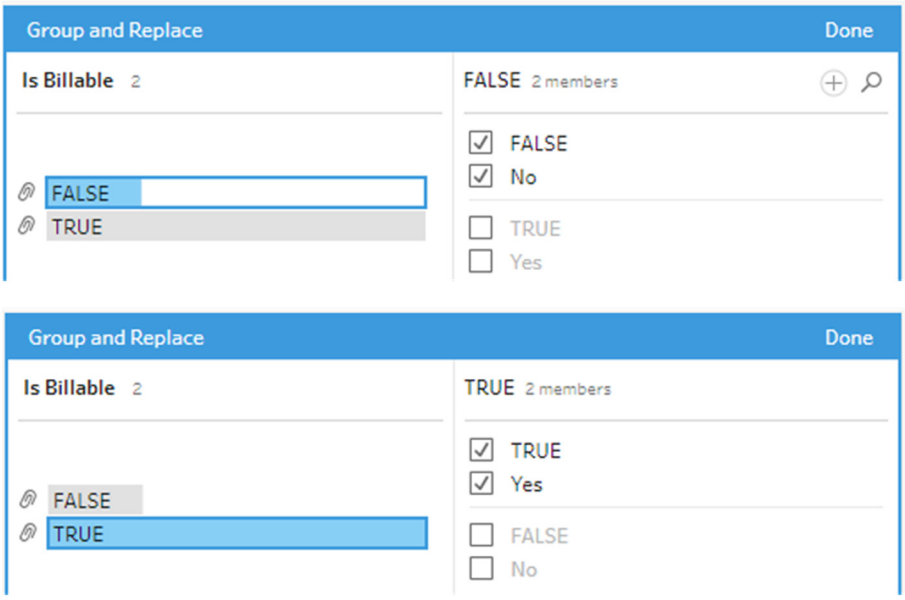


Figure 4.21 – Is Billable field - Group on Manual Selection

We can perform the same manual grouping as in the previous step and then add an Output step to the workflow. Set the Output type to Comma Separated Value (CSV) and select the destination. Now our data is clean, exported, and ready to be analyzed in Tableau. At any point in the process, we can visit the changes to track all the operations.

Figure 4.22 – All Changes

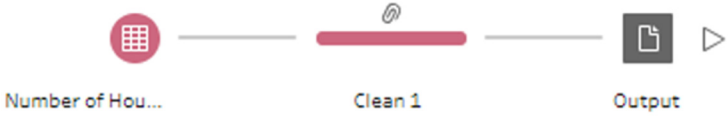


Figure 4.23 – Final workflow

CHAPTER 5
AGGREGATION and PIVOT

The previous two chapters focused primarily on data cleaning. Although the removal of fields improves performance, we need to think about the number of rows and the granularity of the data. So, we need to consider the following questions: *When the data is getting prepared for Tableau visualizations, do we need row-level transactional data? How much of drill-down capability does the end user need?*

That is where aggregations come in. In most cases, we don't need every record. Aggregations allow us to take large amounts of data, which may not be as useful at the lowest level of detail, and roll them up and create buckets of groups of data with summarized numbers. The data is thus most helpful to represent key metrics with simple slice-and-dice and filter features on the Tableau dashboard used by senior management.

At the same time, the shape of the data causes major issues when creating visualizations. So, pivoting plays an important role in the data structure for visual analytics.

In this chapter, we will discuss Aggregations and Pivoting. Although both can be treated as individual chapters, they were put together in a single chapter for an easier reading and learning experience.

5.1 Aggregations

There are more than 8.5 million people living in New York City. About 55% of them use public transportation. Imagine the size of the data set the MTA (Metropolitan Transportation Authority)

uses to track each transaction. Without aggregations, it would be a huge task to analyze that data set.

Aggregations are used to understand data at a high level. By definition, they are used to summarize data. We use aggregations to create buckets of data in order to gain insight based on specific rules.

This concept of aggregations can be applied using Tableau Prep Builder. But first, let us talk about the different ways aggregations can be applied.

5.2 Aggregation Functions

There are 12 types of aggregation functions we can use in Tableau Prep Builder.

If we are familiar with the different aggregation functions, we can skip over this section.

1. **Sum** – This is the most common aggregation and will be chosen as default. It is the total value of two or more numbers.

Name	Hours Worked
John Smith	8
John Smith	8
John Smith	8
Sum	**24**

Table 5.1 – Sum example

We see that John Smith has three rows of data that list his hours worked. If we were to aggregate the data by using the Sum aggregation, we would be left with one row of data that shows he worked a total of 24 hours.

2. **Average** – This is also a common aggregation. It is calculated by taking the sum of the values in a table and dividing that sum by the total count of rows. It is also termed *mean* value.

Name	Hours Worked
Tyler Morgan	8
Tyler Morgan	3
Tyler Morgan	10
Average	**7**

Table 5.2 – Average example

The average in this case is calculated as (8 + 3 + 10)/3, which equals 7.

3. **Median** – This aggregation is looking for the midpoint in a group of integers when sorted in either low-to-high or high-to-low order.

For example, in a data set containing 30, 57, 22, 93, and 71, the median, or midpoint, would be 57. When the numbers are arranged, the central value is selected: 22, 30, **57**, 71, 93.

If the data set contains an even number of entries, the average of the middle two numbers will be the median. For example, in a data set containing 30, 57, 22, 93, 71, and 66, the median is 61.5.

When the numbers are arranged 22, 30, **57**, **66**, 71, 93, the median will be (57 + 66)/2.

4. **Count –** A count is simply the number of times a particular entry can be found.

Name	Hours Worked
John Smith	8
John Smith	3
John Smith	12
John Smith	9
Count (Name)	**4**

Table 5.3 – Count example

We see that there are four rows of data. If we were to take the count of Name, the result is a count of 4 since it occurs four times.

5. **Count Distinct –** Similar to a count, count distinct counts only unique or distinct entries. In the previous example, John Smith is repeated four times. While the count of rows is 4, the count distinct is only 1.
6. **Minimum –** This aggregation is looking for the smallest value in a data set. In the previous example, the minimum, or smallest, is 3.
7. **Maximum –** The opposite of minimum is maximum. The maximum is looking for the largest or the highest value. In the previous example, the maximum hours worked is 12.
8. **Std. Dev –** The standard deviation represents how spread out the numbers are (how much they vary) based on the mean value. It is used to identify if a number is normal.

Name	Hours Worked
Tyler Morgan	8
Tyler Morgan	3
Tyler Morgan	12
Tyler Morgan	9
Standard Deviation	**3.24**

Table 5.4 – Standard Deviation example

Although there are a lot of steps involved in calculating standard deviation and calculating each step takes some time, we will quickly look at the steps here.

To calculate Std. Dev, we must first find the *mean* or average, which is 8. Next, we must find the *difference* from mean for each row—0, -5, 4, 1. Next, we need to calculate the *variance.* We need to square the difference—0, 25, 16, 1—and then find the average, which results in 10.5. The standard deviation is the square root of the variance, which is 3.24.

9. **Std. Dev Pop.** – The population standard deviation is used when there is access to an entire data set versus just a sample.
10. **Variance** – The variance is an indication of how much a given set of numbers vary from the mean. It is calculated by taking the square of standard deviation. That was illustrated when calculating the Std. Dev.
11. **Variance Pop.** - The population variance is used when there is access to an entire data set versus just a sample.

12. **Percentile** – This is the measure indicating the value below which a given percentage of observations fall. It is also called relative standing.

5.3 Group By

The *Group By* function is used when deciding the level of detail of the data set. As the data become more aggregated, granularity is lost. So, we need to be careful in deciding the level of data aggregation.

Aggregations can be grouped by one or more fields.

Name	Department	Hours Worked
John Smith	Analytics	8
Christopher Wood	HR	8
John Smith	Finance	8
John Smith	Analytics	8
Christopher Wood	HR	4

Table 5.5 – Sample data

In Table 5.5, there are two fields we can group by with three combinations. The table can either be grouped by Name field or Department field individually or Name and Department together. Each one will give different results, illustrated in the figures below using the Sum as a default aggregation. Any other type of aggregation can be applied.

Name	Hours Worked
John Smith	24
Christopher Wood	12

Table 5.6 – Group By - Name

Department	Hours Worked
Analytics	16
HR	12
Finance	8

Table 5.7 – Group By - Department

Name	Department	Hours Worked
John Smith	Analytics	16
Christopher Wood	HR	12
John Smith	Finance	8

Table 5.8 – Group By – Name and Department

Caution must be taken when aggregating, as duplicates can cause issues. Table 5.6 is aggregating by the Name field. If there are multiple employees in an organization named John Smith who belong to different departments, the data should not be aggregated by Name. Name and Department should be the fields that must be used for Grouping, as in Table 5.8.

5.4 Example 1

For this example, we will be using the *Reycling_Diversion_and_Capture_Rates*.csv file. This data source is available to the public and is based on recycling rates in New York City. It is meant to tell us how much waste is disposed of by the Department of Sanitation by recycling and how much paper, metal, glass, and plastic are recycled.

The goal in this flow is to understand how much each borough is recycling per fiscal year. With a simple six-step process, we can group and aggregate the data to get this information.

Data can be aggregated in both Tableau Desktop and Tableau Prep Builder. As mentioned before, performance must always be considered when connecting to large data sets, and the focus should be on the end user's needs and the availability of finer granular data. If the information is not needed in raw form at the transactional level, it may be better to aggregate the data prior to connecting to it in Tableau Desktop.

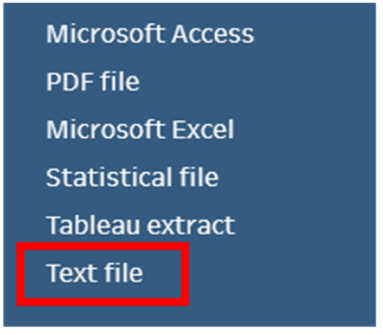
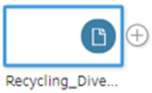


Figure 5.1 – Connect to data file

The first step is to connect to the data source, which is a text file found in Chapter 5 folder.



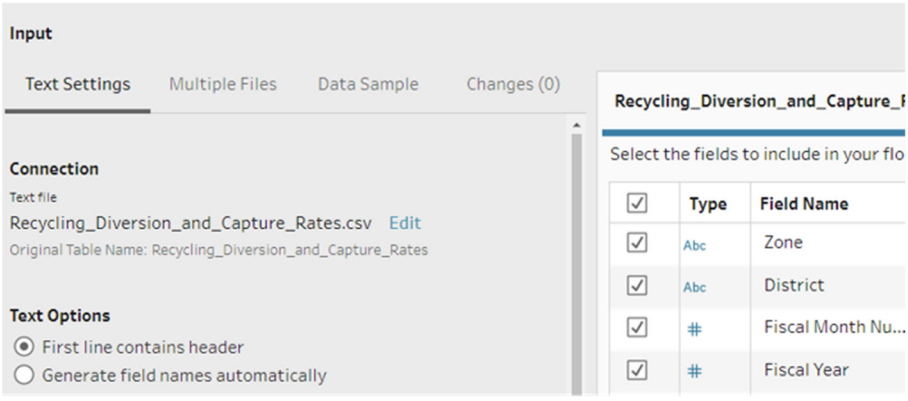


Figure 5.2 – Data input – Default configuration

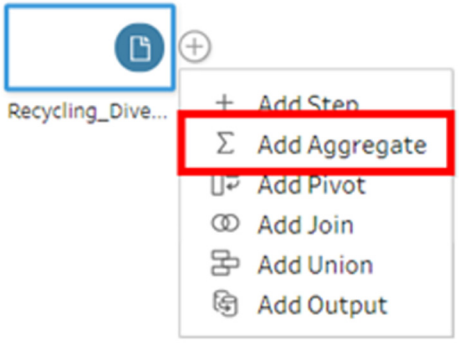


Figure 5.3 – Adding an Aggregate Step

Next, we must use the Add Aggregate step and choose the Dimensions needed for grouping and the Measures needed for aggregation. Any extra fields will increase the granularity of the data set. The requirement here is to aggregate the diversion rate for each fiscal year and zone (borough).

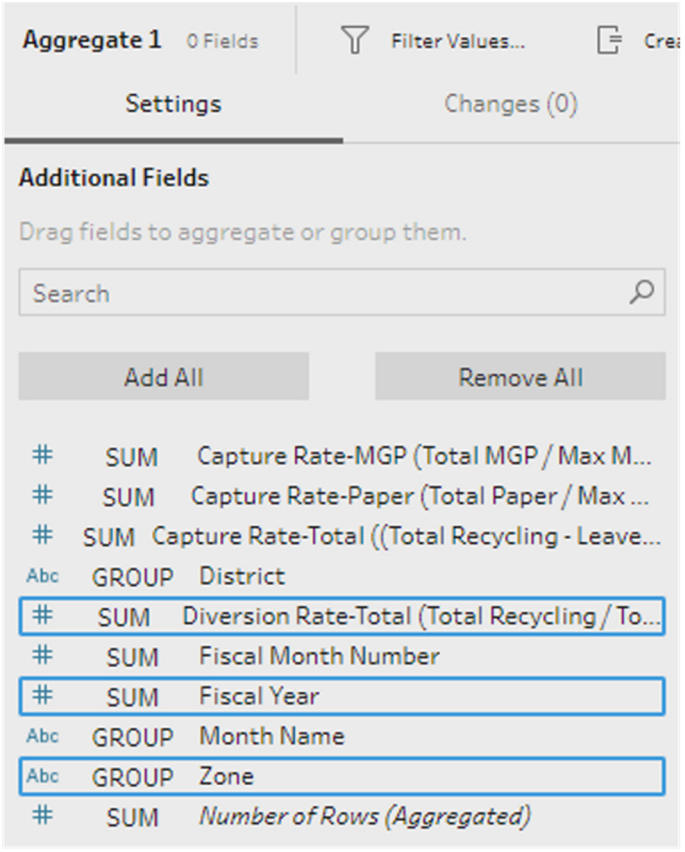


Figure 5.4 – Fields for Grouping and Aggregation

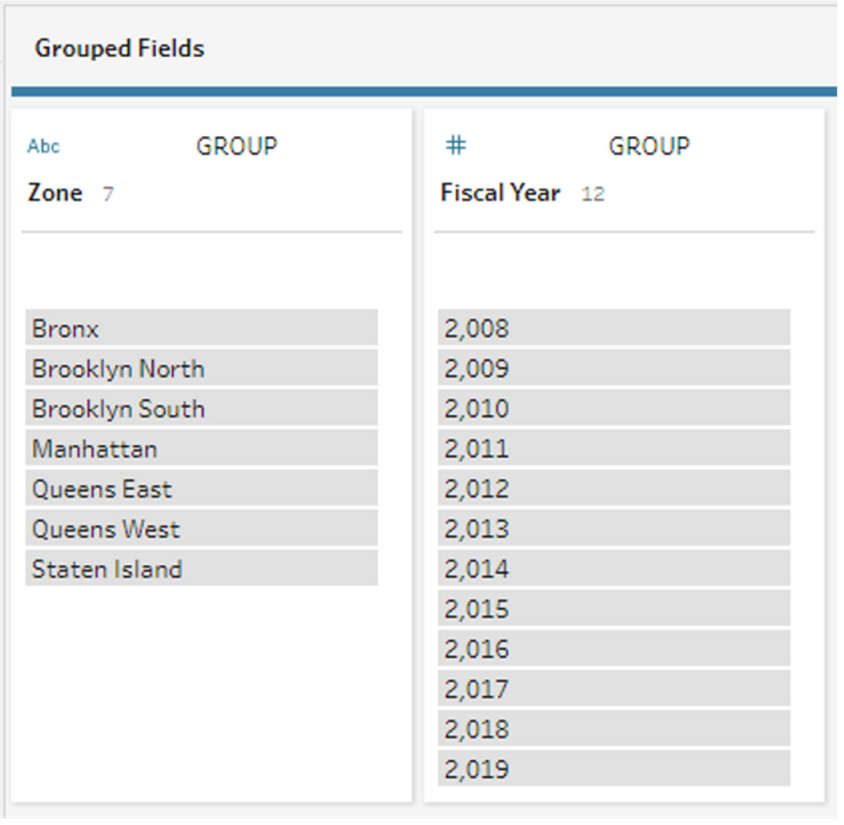


Figure 5.5a – Fields for Grouping

In the *Grouped Fields* section, we need to drag the fields to be grouped—Zone and Fiscal Year. Then we can drag the Diversion Rate-Total field to the *Aggregated Fields* section. By default, SUM is selected as the aggregation function. We can right-click and select other aggregations

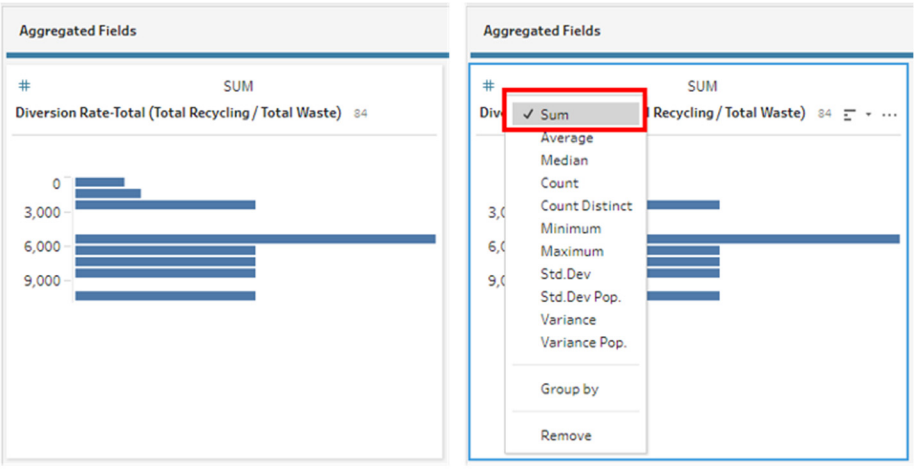


Figure 5.5b – Field for Aggregation

***Note:** The Fiscal Year field is an integer, but it should not be aggregated.

We can click on the headers to change the group or aggregation type.

Figure 5.6 – Final workflow

Now the data is ready for output. We can add a cleaning step to see the preview or if any further cleaning is needed. A new data set will be created to show the Diversion Rate-Total per borough in each fiscal year.

Save to Output.hyper

Zone	Fiscal Year	Diversion Rate-Total (Total Recycling / Total Waste)
Queens East	2,015	5,781.597029357153
Queens West	2,010	5,887.464573596908
Queens East	2,011	5,772.259535512095
Bronx	2,019	1,409.4072395386452
Queens West	2,012	5,929.542458728833
Queens East	2,017	5,777.841654089907
Brooklyn North	2,017	6,448.592823569967
Queens East	2,013	5,728.757045317197
Manhattan	2,008	10,176.44754164557
Queens West	2,008	5,879.795894190939

Figure 5.7 – Preview of Output

5.5 Pivot

Tableau Prep Builder provides a very important and interesting feature in Pivot. Although this feature is available in Tableau Desktop, it is not possible for all types of data. But Prep can do it for any type of data.

When we look at some data sets or reports generated, they look like Pivot tables where the Dimensions are on both X-axis (horizontal) and Y-axis (vertical) and values filled for each of the cells. That is referred to as a Crosstab. If all the columns are converted into rows, then the feature is called Transpose. This feature in Tableau Prep Builder is called Pivoting.

To technically understand this, the columns and rows are considered two arms, and the point at which they converge is the fulcrum. If the horizontal arm is rotated to meet the vertical arm using the fulcrum, it gives a pivoted data set. In very simple words, a wide-flat data set is converted into a thin-long-columnar data set.

Let us look at some very simple data that is wide. It is a very small retail data set that has Sales and Number of Units. We need Unit Price from this data set. So, there is a need for an additional column, which will be Sales divided by Number of Units.

	Sales			Number of Units		
Category →	PC	Tablet	Phone	PC	Tablet	Phone
Customer ↓						
Adam	1500	2500	6000	2	6	10
Gemma	1200	4000	1500	1	12	3
Lisa	650	1000	8500	1	4	15
Michael	5000	1500	5500	7	4	5

Table 5.9 – Wide data set

With this kind of a data structure, it will be difficult to create calculated fields since it adds three more columns for each calculation and the data file increases in width, becoming very cumbersome and making it difficult to read and extremely complicated for visual analytics.

So, we need to use the top left corner as the fulcrum and convert all the measure columns into rows.

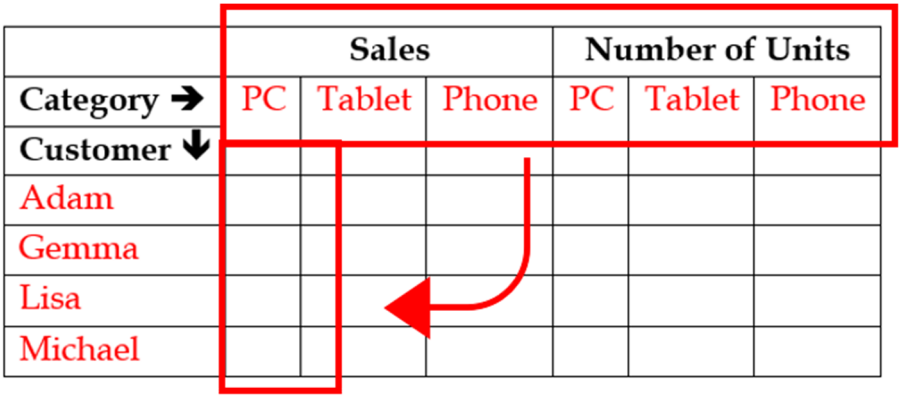


Figure 5.8 – Pivot Activity

After the pivot action, the table will be in a more readable row-level format, which is easy for calculations.

Customer	Category	Sales	Number of Units	Unit Price = Sales/Units
Adam	PC	1500	2	750
Adam	Tablet	2500	6	416.67
Adam	Phone	6000	10	600
Gemma	PC	1200	1	1200
Gemma	Tablet	4000	12	333.33
Gemma	Phone	1500	3	500
Lisa	PC	650	1	650
Lisa	Tablet	1000	4	250
Lisa	Phone	8500	15	566.66
Michael	PC	5000	7	714.28
Michael	Tablet	1500	4	375
Michael	Phone	5500	5	1100

Table 5.10 – Pivoted data set result

We will use this action in Tableau Prep Builder. One very important point to remember is that the number of rows for each ID will increase by the number of columns pivoted. For example, we have a data set of 1,000 rows with 1,000 unique keys, and 10 columns need to be pivoted. After pivoting, the resulting data set will have 1,000 * 10 = 10,000 rows.

5.6 Example 2

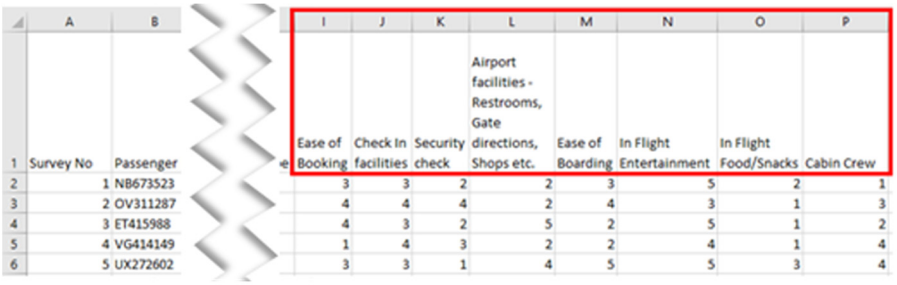
The file required for this example is in Chapter 5 folder.

> *Air Passenger Survey Data.xlsx* is an Excel file that has a sample passenger satisfaction survey from various airports across the United States. As introduced in Chapter 2, this Excel file has a tab that shows all the questions that have been answered. Each question will have a rating from 1 to 5, with 1 being very dissatisfied and 5 being very satisfied.

Requirement: Need to build a dashboard that shows the percentage of satisfied customers for each question. Users would also need the ability to slice and dice or filter by airports, age group, type of travel, and so on.

The reason to select a sample survey data set is that data pivoting is most widely used for survey analytics, a vast area for Tableau users and visual analytics in general.

First, let us investigate the shape of the data set. We can see that columns I through P are the survey questions. Creating percentages for each question and assembling them on the dashboard is a little tedious. Instead, we will pivot (or transpose) these eight columns.



	A	B	I	J	K	L	M	N	O	P
1	Survey No	Passenger	Ease of Booking	Check In facilities	Security check	Airport facilities - Restrooms, Gate directions, Shops etc.	Ease of Boarding	In Flight Entertainment	In Flight Food/Snacks	Cabin Crew
2	1	NB673523	3	3	2	2	3	5	2	1
3	2	OV311287	4	4	4	2	4	3	1	3
4	3	ET415988	4	3	2	5	2	5	1	2
5	4	VG414149	1	4	3	2	2	4	1	4
6	5	UX272602	3	3	1	4	5	5	3	4

Figure 5.9 – Fields for Pivot

The first step is to input the data as an Excel file, as demonstrated in Chapter 2. We will create a new workflow and start with *Air Passenger Survey Data.xlsx*.

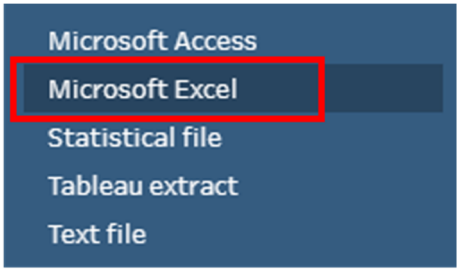


Figure 5.10 – Connection to xlsx

We will use the Air Passenger Survey table with default configuration.

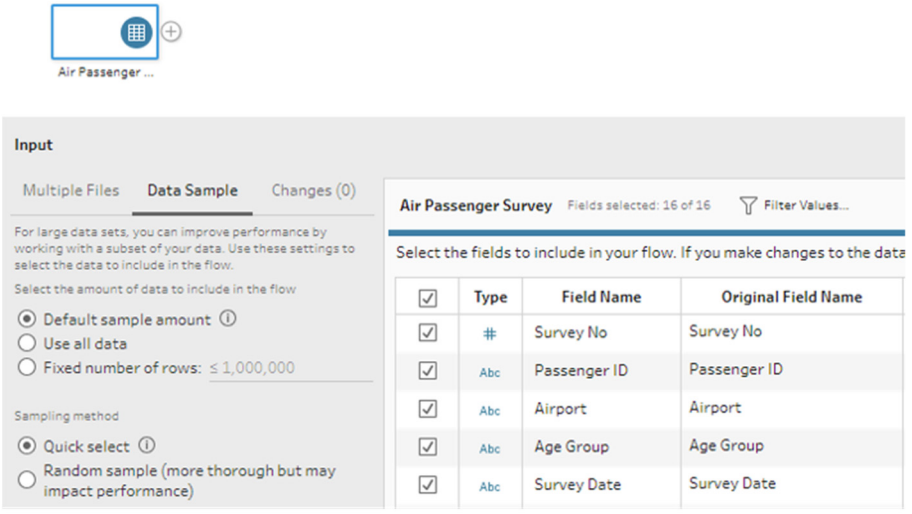


Figure 5.11 – Input data configuration

Next, we will use the Pivot step. Here, we have to be very careful with the fields that need to be pivoted. The columns that are selected for the pivot function will have a header called Pivot1 Names and the value entries will be in a field called Pivot1 Values.

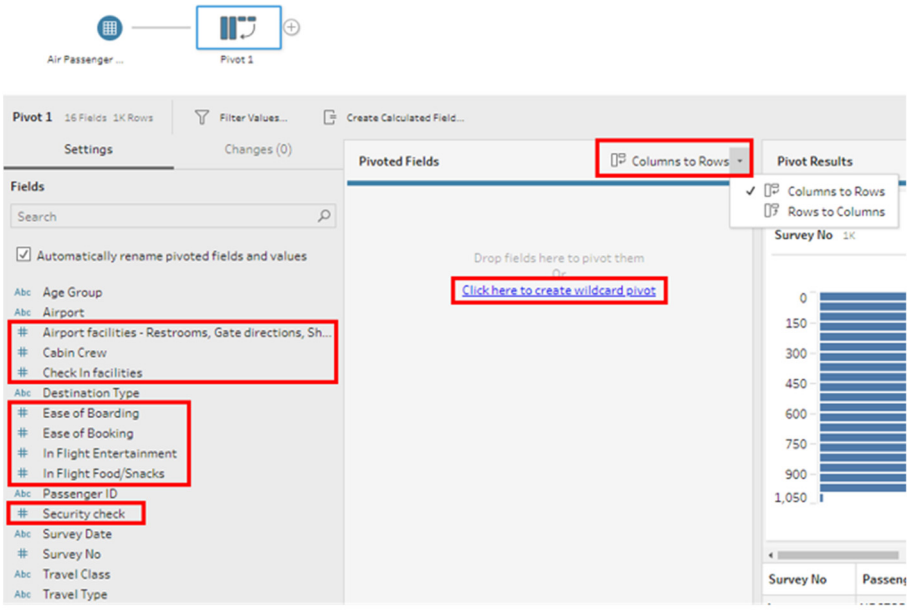


Figure 5.12 – Pivot configuration and properties

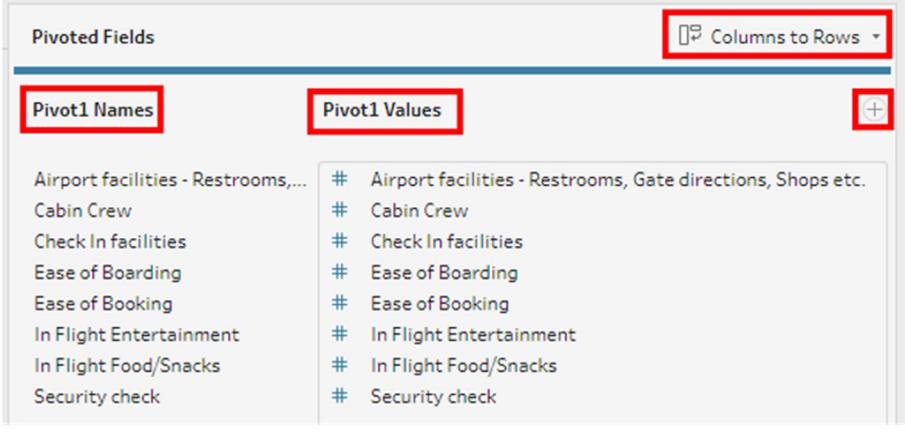


Figure 5.13 – Fields for pivoting

We can use the + symbol in the right corner to add more fields that need to be pivoted. This is the requirement shown in the demo when multiple fields have to be brought at a row level.

We can also use a Wildcard Pivot by searching for commonly names fields. In most cases Pivoting or Transpose activity would convert, Columns to Rows. But we also have the option to Unpivot where we can convert Rows to Columns.

The fields on the left panel are the dimensions used for slicing and dicing, which stay as is.

Let us save the workflow as *Pivot Workflow*.

After this step, we are ready for output. We need to rename the fields in the Pivot step or use the Cleaning step.

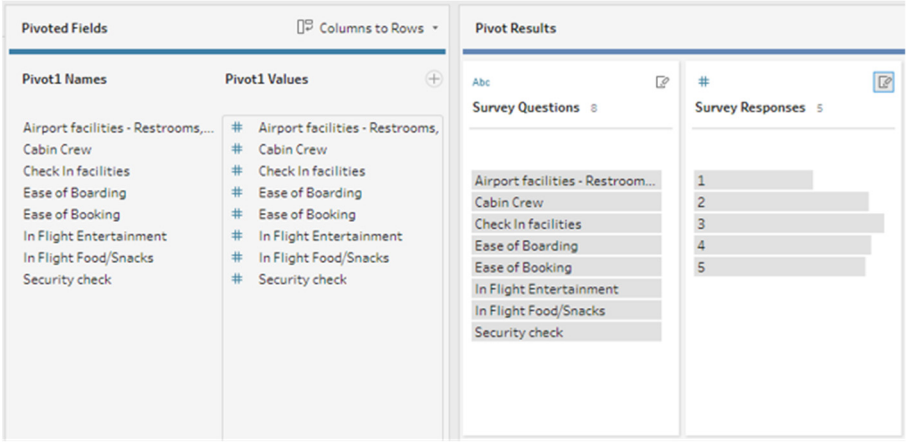


Figure 5.14 – Pivot Results and renaming pivoted fields

Now, the output will have the questions repeated at a row level for each Passenger ID. The final workflow looks like this:

Figure 5.15 – Final Workflow

CHAPTER 6
JOINS and UNIONS

Combining data sets to use fields across multiple files in the same visualization has been very challenging for Tableau Desktop users. As mentioned in Chapter 1, Blends and Cross database Joins have been very powerful, but they have their own drawbacks. This chapter deals with the usage of multiple files in the same workflow to be combined into a composite data set. We will discuss Joins and Unions on Tableau Prep Builder.

6.1 Join

A Join is an operation used to add fields to an existing table from two or more tables based on some matching criteria. The most common usage in relational databases exhibiting a star schema is when the Fact table is joined with multiple Dimension tables (e.g., Date Dimension, Employee Dimension, Counterparty Dimension, Location Dimension, etc.) based on the required Key fields. A Join creates wide tables by adding columns from all the tables used. We also have the option of removing unwanted or redundant columns.

The same concept can be applied using the Tableau Prep Builder interface. Before we go into an actual example, we need to understand the various Join types available in Tableau Prep Builder.

To understand the various Joins, we will use two simple tables—Name and Age, and Name and Course.

NAME	AGE
Adam	25
Emily	20
John	21
Kathy	26
Michael	24
Stella	27

Table 6.1 – Age Table

NAME	COURSE
Adam	Math
Emily	Engineering
John	Psychology
Kevin	Media
Olivia	Fine Arts
Samuel	Biology

Table 6.2 – Course Table

For a Join, we need three basic things:

1. A common field. In this case, the Name field.
2. The common field must match the data type. In this case, both are strings. In many cases, we use an ID or Index field or Key Identifier. If the data type does not match, we can change the data type in the workflow.
3. Unique and Not Null values for the Key field to avoid duplicates.

***Note:** The Join clauses (=, >, <, >=, <=, <>, or !=) also need to be considered in appropriate cases. The most commonly used is the equal (=) match.

6.2 Types of Joins

1. **Inner Join –** This is one of the most commonly used Join types. It is used to identify only the common entries between the tables. It works like an intersection in the Venn Diagram. This is the default Join.

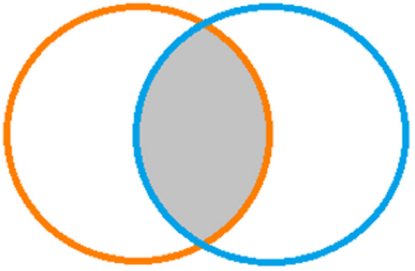


Figure 6.1 – Inner Join

NAME	AGE	COURSE
Adam	25	Math
Emily	20	Engineering
John	21	Psychology

Table 6.3 – Inner Join result

2. **Left Join** – This is also one of the most commonly used Join types. It provides all the values from the left table along with the common entries between the tables.

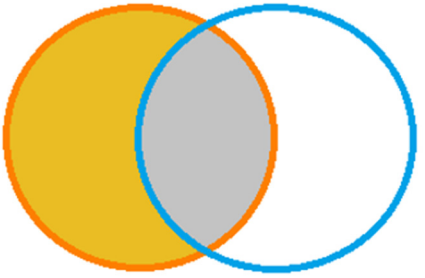


Figure 6.2 – Left Join

NAME	AGE	COURSE
Adam	25	Math
Emily	20	Engineering
John	21	Psychology
Kathy	26	*Null*
Michael	24	*Null*
Stella	27	*Null*

Table 6.4 – Left Join result

3. **Right Join** – This is the opposite of the Left Join where importance is given to the right table. It provides all the values from the Right table along with the common entries between the tables.

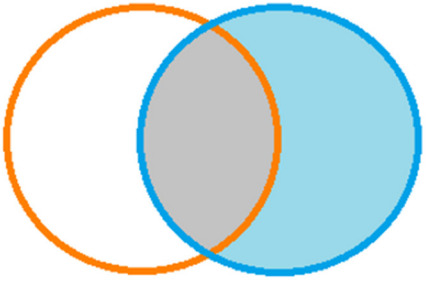


Figure 6.3 – Right Join

NAME	COURSE	AGE
Adam	Math	25
Emily	Engineering	20
John	Psychology	21
Kevin	Media	*Null*
Olivia	Fine Arts	*Null*
Samuel	Biology	*Null*

Table 6.5 – Right Join result

4. **Full Outer Join** – This can be simply referred to as Outer Join. It will provide a master data set with all members from all data sets, including the common members.

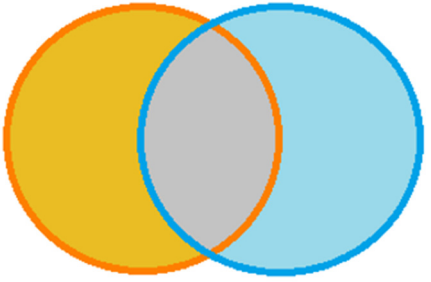


Figure 6.4 – Full Outer Join

NAME	AGE	COURSE
Adam	25	Math
Emily	20	Engineering
John	21	Psychology
Kathy	26	*Null*
Michael	24	*Null*
Stella	27	*Null*
Kevin	*Null*	Media
Olivia	*Null*	Fine Arts
Samuel	*Null*	Biology

Table 6.6 – Full Outer Join result

5. **Left Join without Intersection** – As the name suggests, this will show the members in the left table without the matching (intersection) entries. In Prep, it is called *Left unmatched only*.

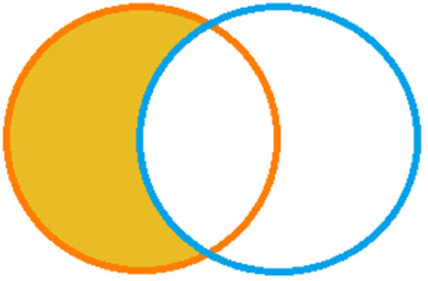


Figure 6.5 – Left Join without Intersection

NAME	AGE	COURSE
Kathy	26	*Null*
Michael	24	*Null*
Stella	27	*Null*

Table 6.7 – Left Join without Intersection result

6. **Right Join without Intersection –** As the name suggests, it will show the members in the right table without the matching (intersection) entries. In Prep, it is called *Right unmatched only*.

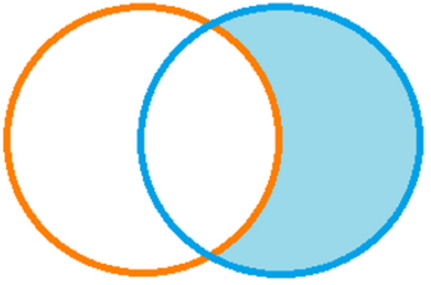


Figure 6.6 – Right Join without Intersection

NAME	COURSE	AGE
Kevin	Media	*Null*
Olivia	Fine Arts	*Null*
Samuel	Biology	*Null*

Table 6.8 – Right Join without Intersection result

7. **Full Outer Join without Intersection –** This Join will show all members of all tables without the common (intersection) entries. In Prep, it is called *Unmatched only*. If we look at it further, it is actually the union of the Left Join without Intersection and Right Join without Intersection.

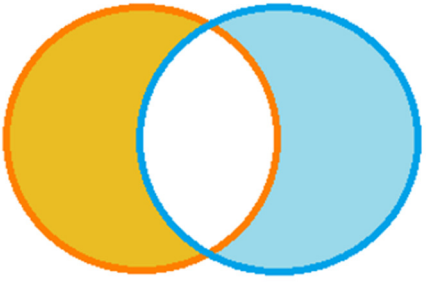


Figure 6.7 – Full Outer Join without Intersection

NAME	AGE	COURSE
Kathy	26	*Null*
Michael	24	*Null*
Stella	27	*Null*
Kevin	*Null*	Media
Olivia	*Null*	Fine Arts
Samuel	*Null*	Biology

Table 6.9 – Full Outer Join without Intersection result

6.3 Example 1

All the files required for this example are in Chapter 6.

> ***Note:** We need to be a little careful about this folder since there are multiple variations of the same data. Different files are used for different examples. Although they are named the same, they have different file extensions.

1. Trade transaction information for different types of stock. This is a sample *Cancels and Corrections data.tde* file with some random values for four years.
2. State names in full for the United States and their regional groups in *State – 2 Letter Abbreviations.xlsx* file.
3. Trade status—whether a particular trade underwent a correction or cancellation in *Status – Cancels and Corrections.txt* file. This is a tab-delimited file.

Requirement: Need an output that provides only trades that have been either cancelled or corrected along with region information with full names for all states.

The idea behind providing different types of data sets is to show that Tableau Prep Builder can blend different types of files and databases in the same flow. We will use Example 1 folder.

The fields are not of the right data type because of the .tde file, but we are not focusing on data cleaning per se, as most of it is

covered in Chapter 2 and Chapter 3. Here, we will only focus on combining files.

The first step would be to bring the data sources as discussed in Chapter 2. We can bring all data sources at once or bring them when required.

We will create a new workflow and start with *Cancels and Corrections.tde*. We will use the default configuration settings. There are no changes needed in the input step.

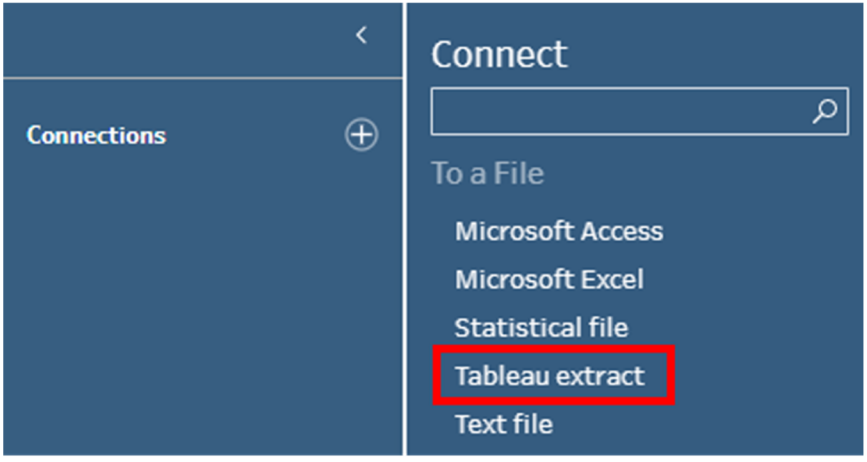


Figure 6.8 – Connection to .tde

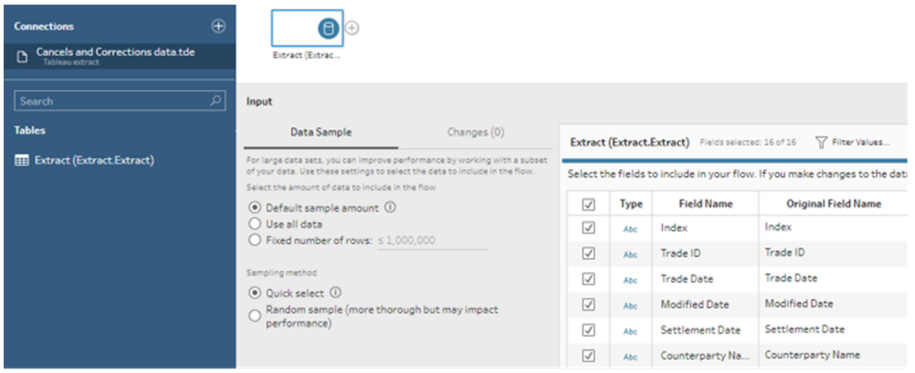


Figure 6.9 – Configuration settings and field settings

Next, we will add the Cleaning step to preview the data. This is optional. But as mentioned before, using this step will help us understand the shape and format of the underlying data after each of the transformation steps through the flow, which is discussed in Chapter 3.

> ***Note:** Notice that the data connection looks like a *cylinder* because of the extract. Database connections look the same.

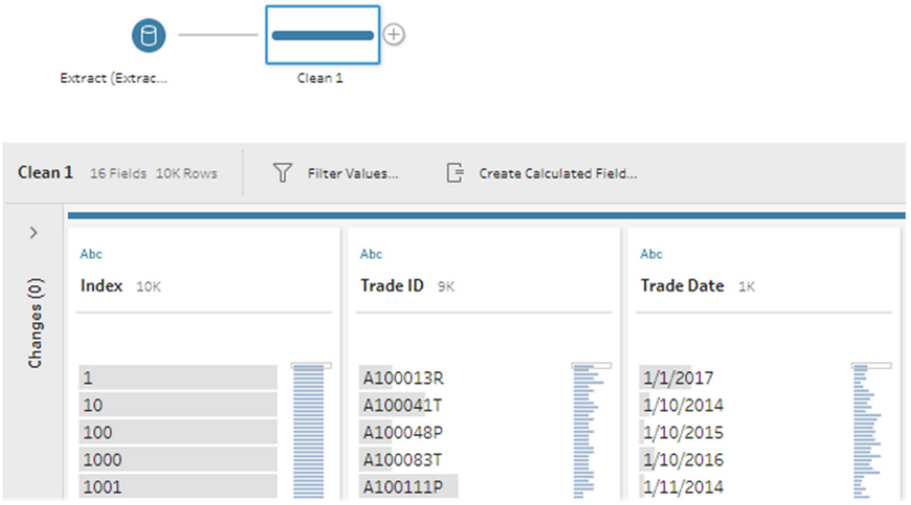


Figure 6.10 – Data Preview

Next, we will bring the *State – 2 Letter Abbreviations.xlsx* to use the State name and Region columns required for the Tableau dashboard.

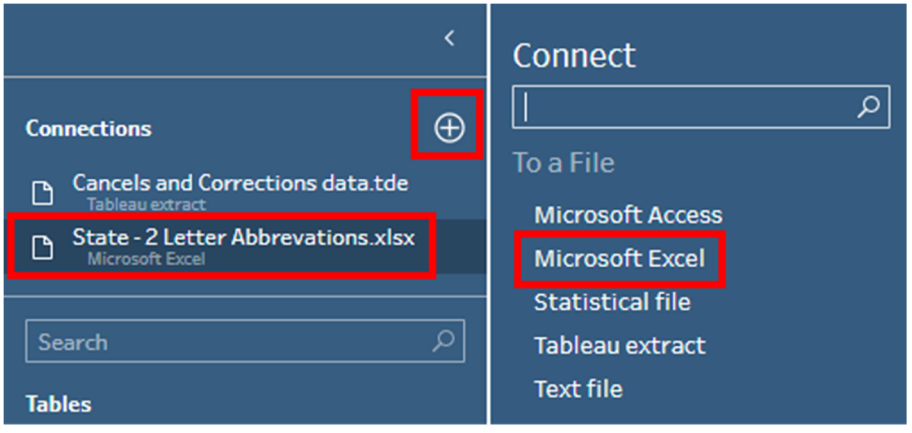


Figure 6.11 – Connection to .xlsx – Adding a new source

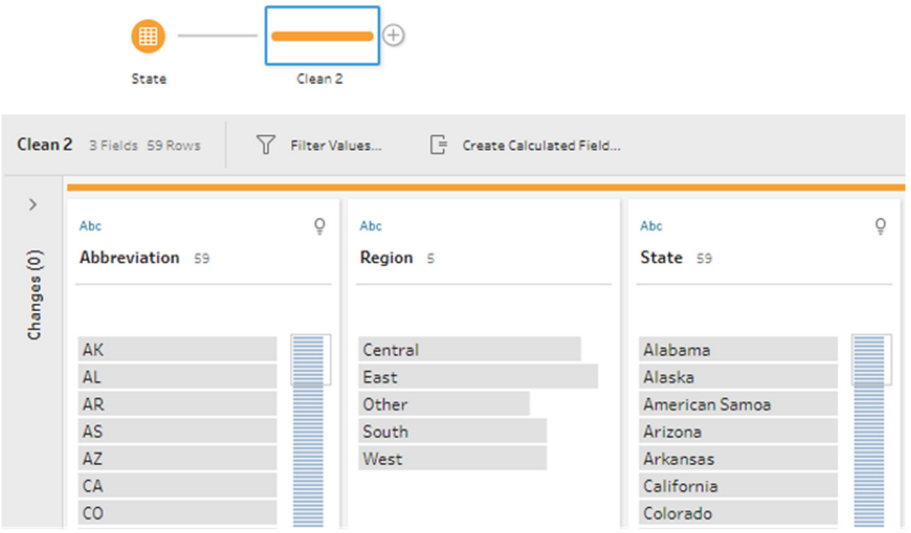


Figure 6.12a – Data Preview

***Note:** Notice that the data connection looks like a *spreadsheet* or a *grid* because of the local Excel file.

As seen in other chapters, since recommendations is switched on, State is suggested as a Geographic role. If needed for mapping, we can use this feature.

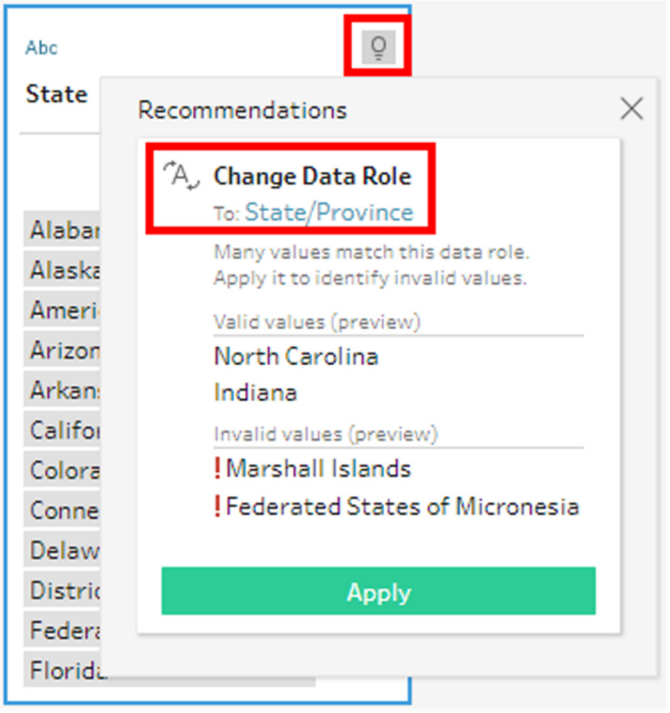


Figure 6.12b – Geographic role suggestion

Next, we will bring the *Status – Cancels and Corrections.txt* to use trade status data. We are keeping all the files ready for the next steps.

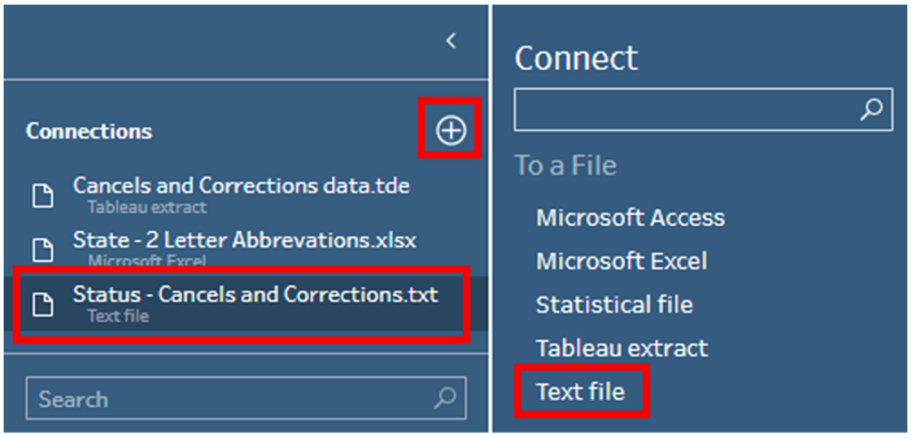


Figure 6.13 – Connection to .txt – Adding a new source

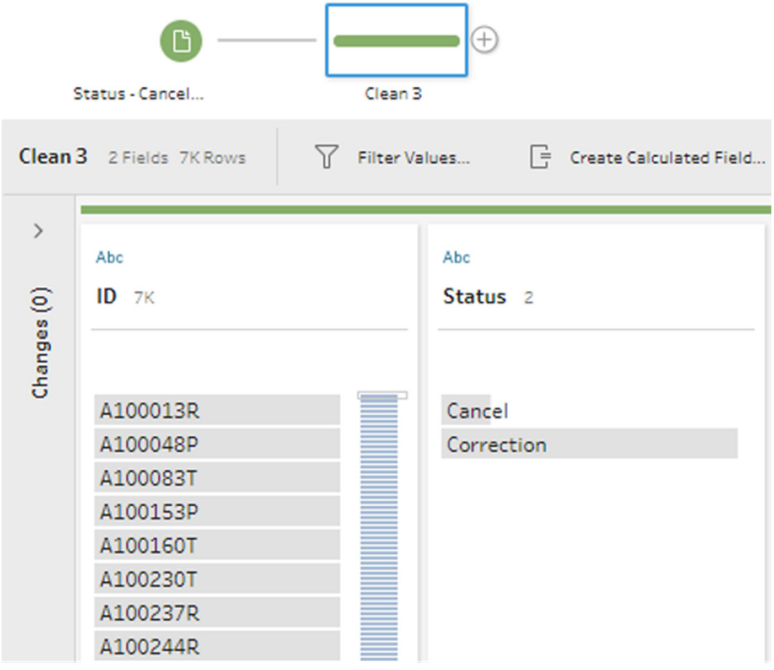


Figure 6.14 – Data Preview

***Note:** Notice that the data connection looks like a *note* because of the local text or *.csv* file.

When this data source is used, Tableau Prep Builder is smart enough to identify that *tab* was the delimiter.

***Note:** Also notice that Tableau Prep Builder has clearly defined each stream in the workflow with a different color (blue, orange, and green) to make identification easy.

Now that all data sources are ready, we can combine them. At any point in the workflow, we can only join two data sources (or two streams, in this case). At this point we can rename the steps as discussed in Chapter 1 for easy identification.

We can create a Join in two ways. We can click on the + icon and add a Join step. After adding a step, we need to manually configure the join step by step by selecting the second data set.

We can just drag and drop an existing stream on the other. This is the easier method as the required data sources are automatically selected and we can continue with the fields and the required clause.

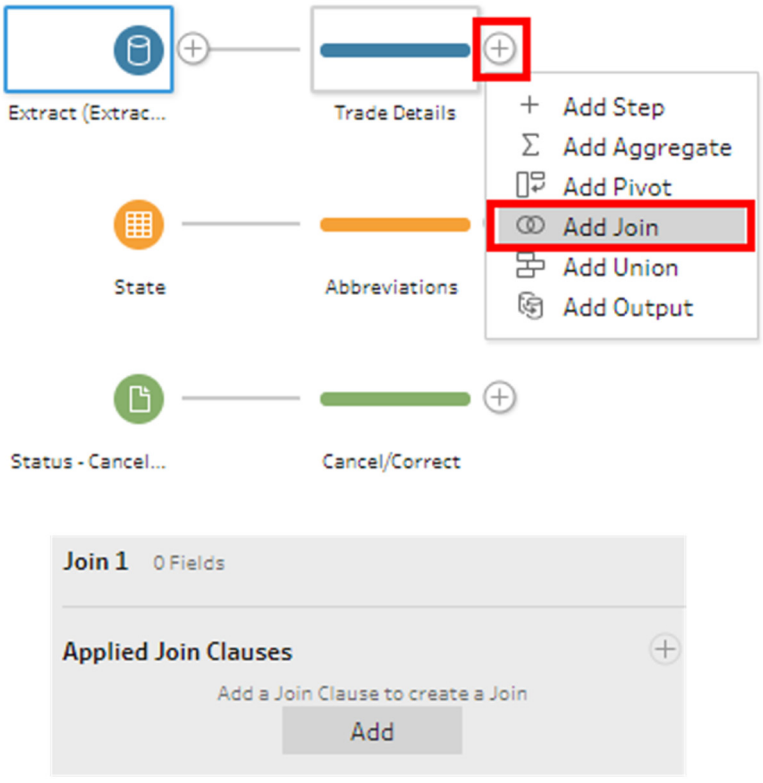


Figure 6.15 – Adding a Join step

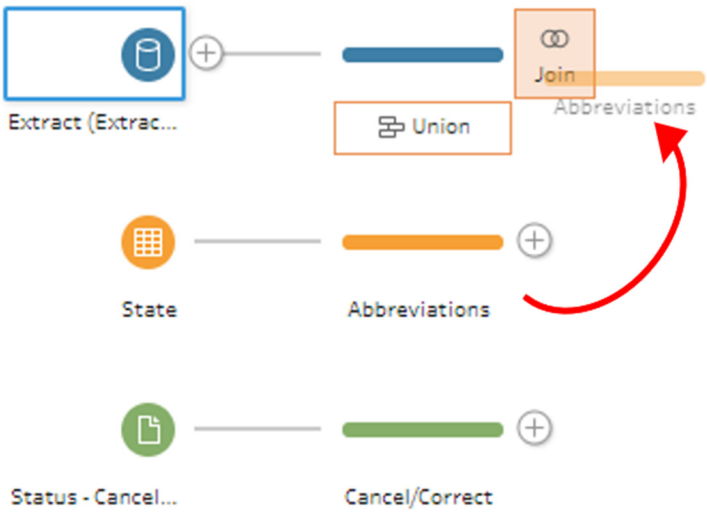


Figure 6.16 – Dragging and Dropping for a Join step

For the first Join, we will use the trade transactions and abbreviations.

We are focusing on the State field, so we need to match the State field and the Abbreviation field.

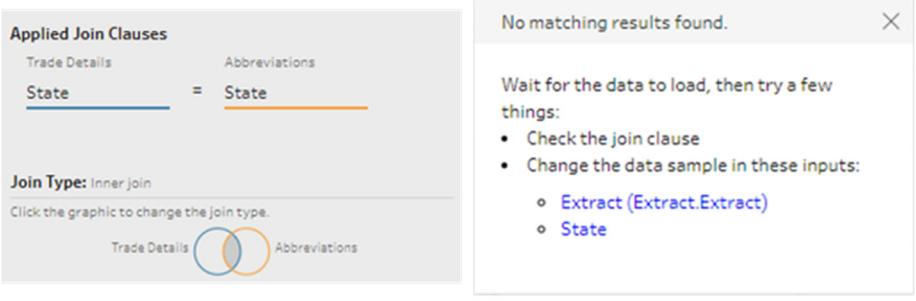


Figure 6.17 – Warning for Join step

We can see that, by default, the tool has picked a field for the Join criterion because of the spelling of the fields. Since this is not the field, there are no matching members, and a warning pop-up is displayed.

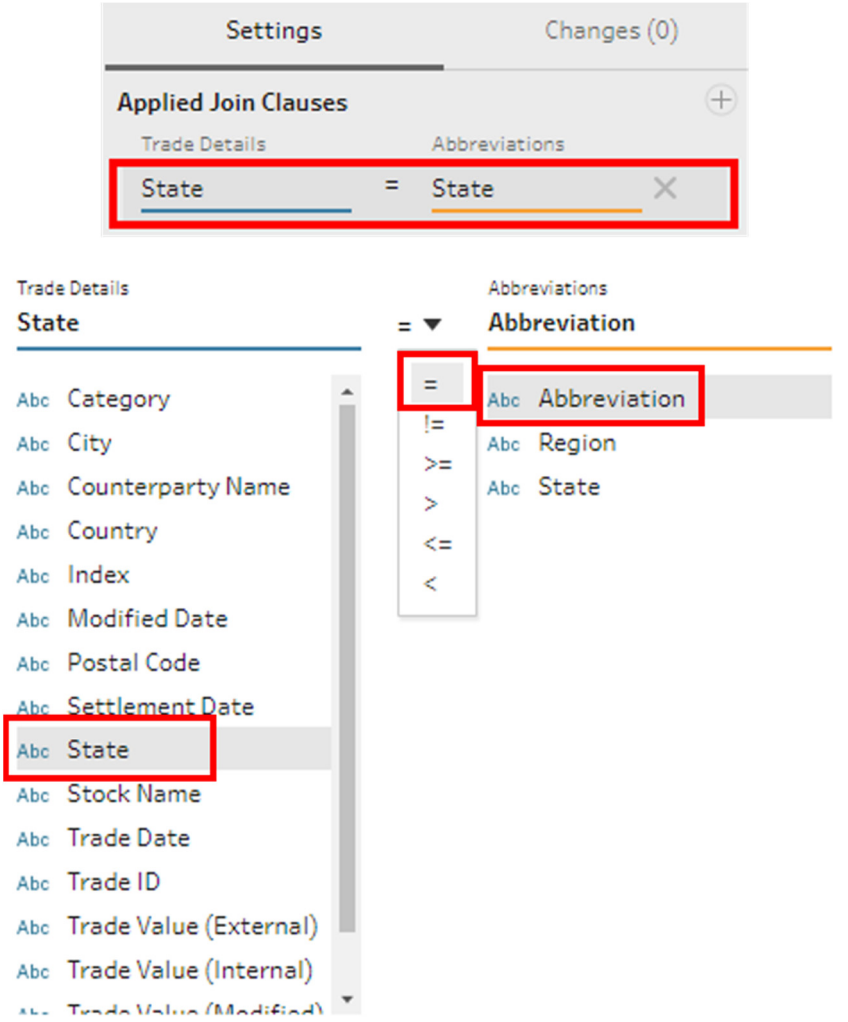


Figure 6.18 – Defining Join field and clause

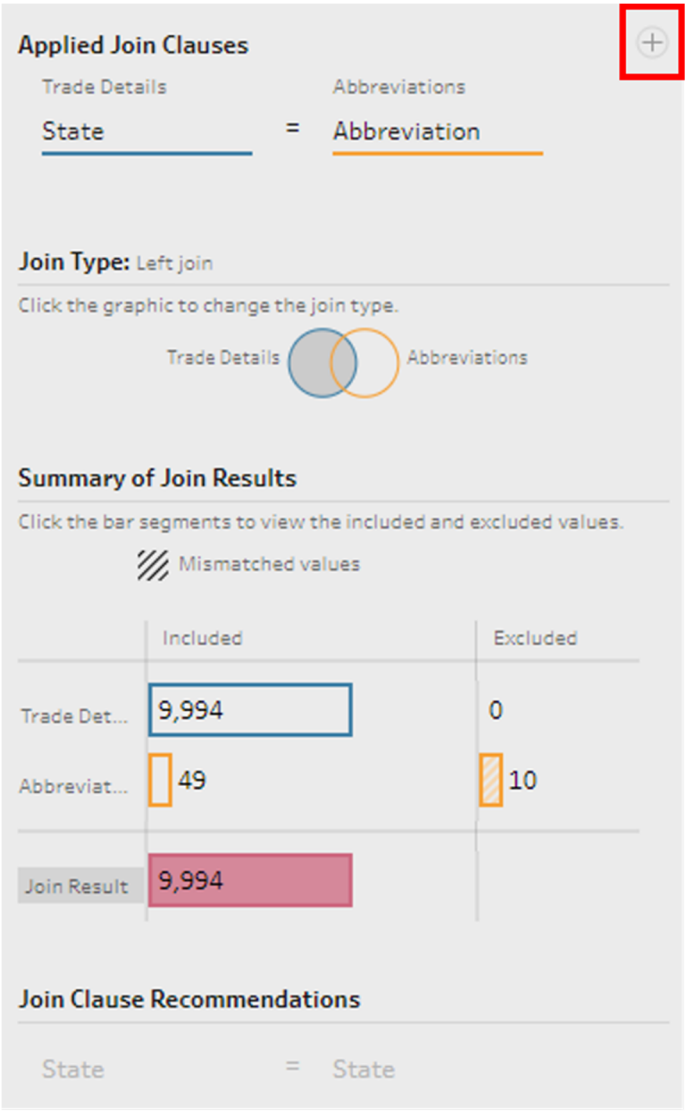


Figure 6.19 – Join configuration window

Next, we need to specify the correct fields and, if required, select the clause for the Join from the dropdown. In this case, we can use an Inner Join or Left Join. Both types yield the same results.

The Join configuration window is fairly intuitive and has three main features.

- **Applied Join clauses** – We select the required matching fields. If needed, we can use the + on the right corner to add additional clauses. Blue is the color for left table (or first stream), and orange is the color for the right table (or second stream).
- **Join type** – As explained before, we can select the appropriate resulting data. For this example, we can use the Inner Join or Left Join. When we select part of the Venn Diagram, it shows different Join types.

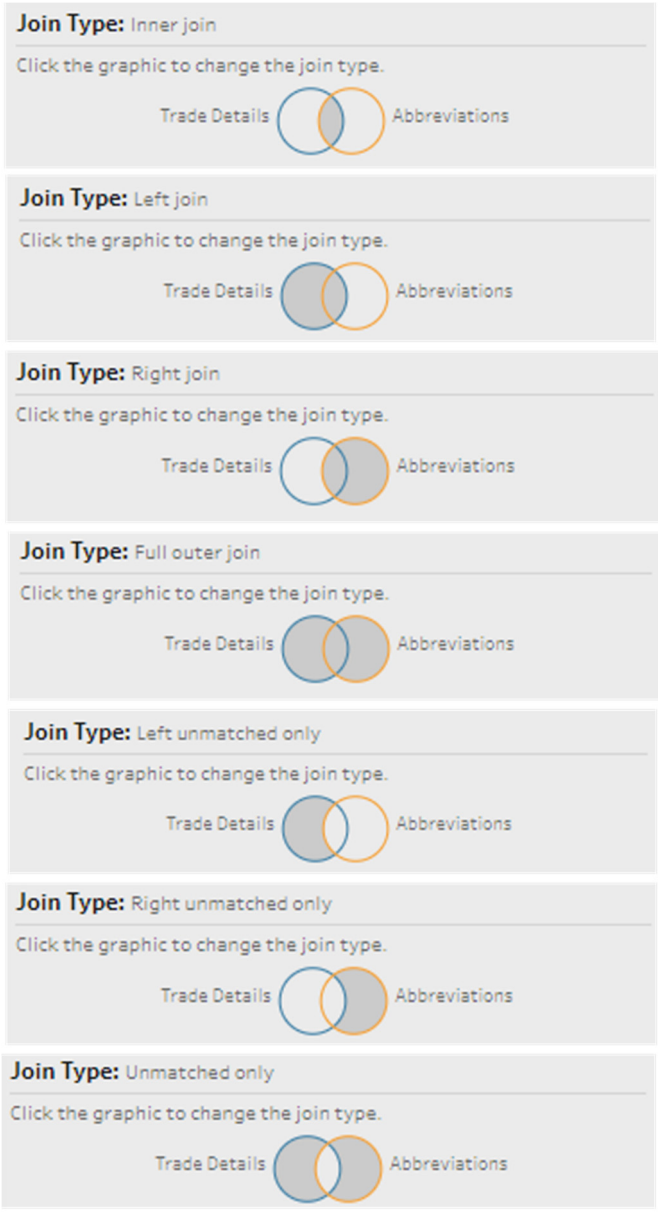


Figure 6.20 – All Join Types

- **Summary of Join Results –** Shows the number of records that are being read and the number of matches and mismatches (cross-hatch lines). The resulting data is shown in dark pink.

***Note:** The Join recommendation is also displayed, but it might not be of much use in this example.

On the right side, we have more data discovery options. It shows the mismatched entries in red. We also have the data Profile pane and Data grid to see the underlying joined data.

We can also use the checkbox to show only the mismatched values from either of the sources. This helps in adding some cleaning steps before the Join step to repair some of the spelling errors or remove some unwanted characters.

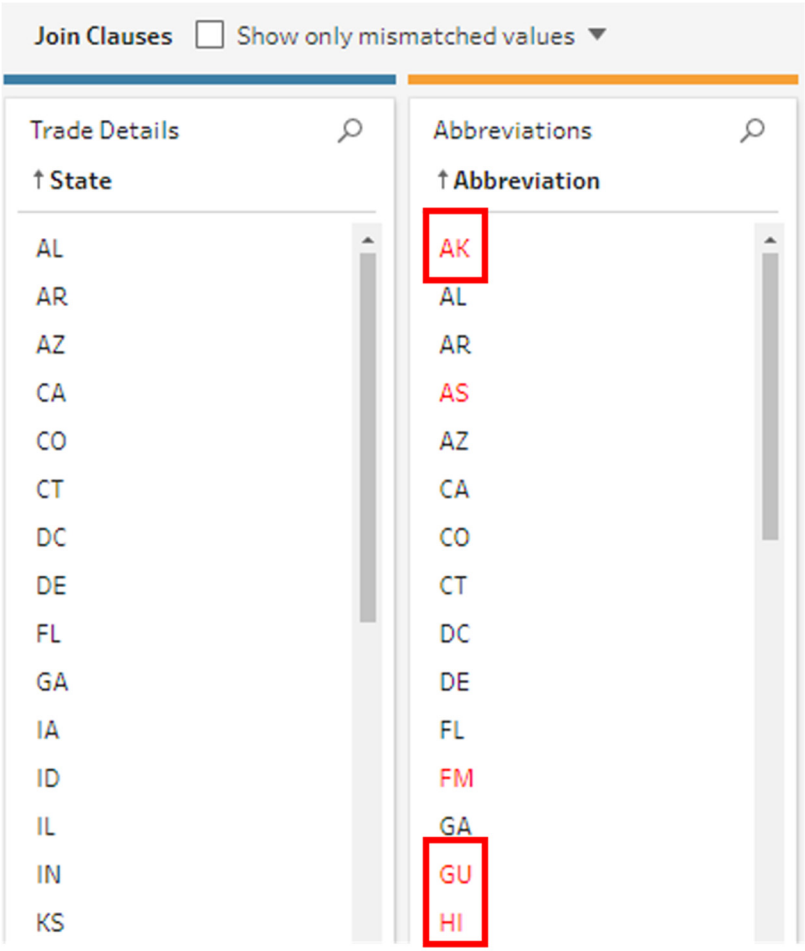


Figure 6.21 – Mismatched entries

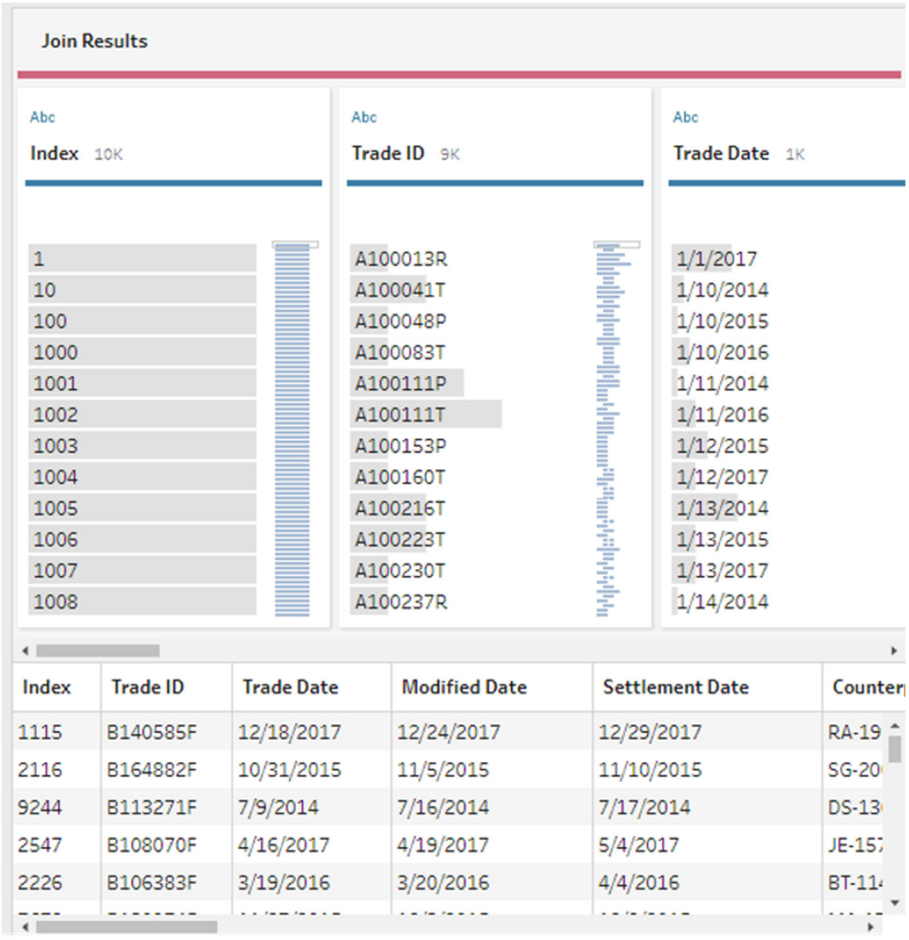


Figure 6.22 – Profile pane and data grid

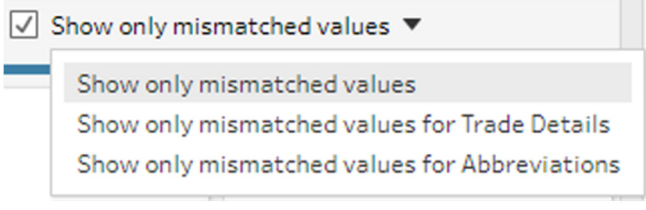


Figure 6.23 – Displaying mismatched entries

Figure 6.24 – Workflow after the first Join

We can save the workflow as *Example 1 Workflow*. As mentioned before, it is good practice to save the workflow after every step.

Now, it is time to use the last data file to identify only the required trades based on the status.

When we create this Join, we see the same error as some random field that matches (State field with Status field). But we need to match on the Trade ID and ID fields.

This time, we will go with an Inner Join because we need only trades that have a status. If we use a Left Join, we can filter out all the Null values in the next cleaning step.

Now the resulting data set will have 6,650 records.

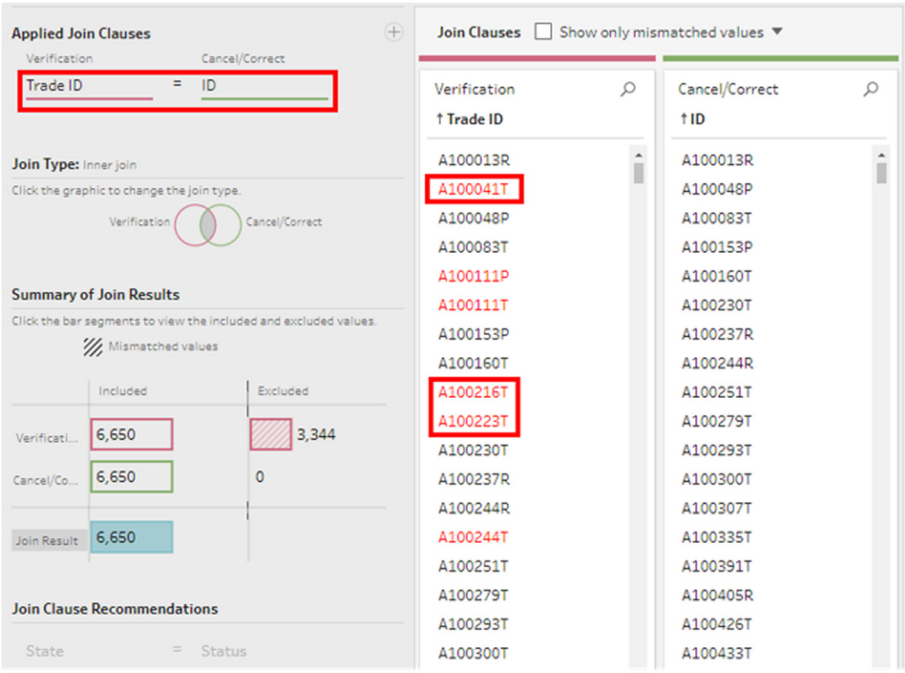


Figure 6.25 – Join configuration and mismatched entries

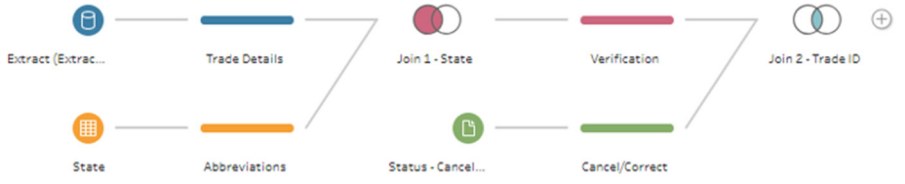


Figure 6.26 – Workflow after the second Join

Next, we need a cleaning step to remove some of the duplicate fields and rename some of the new fields. After that, the data is ready to be output into a file.

We are removing the State field from the Tableau Extract, the Abbreviation field from the Excel file, and the ID field from the Text file. Then we can rename the appropriate fields.

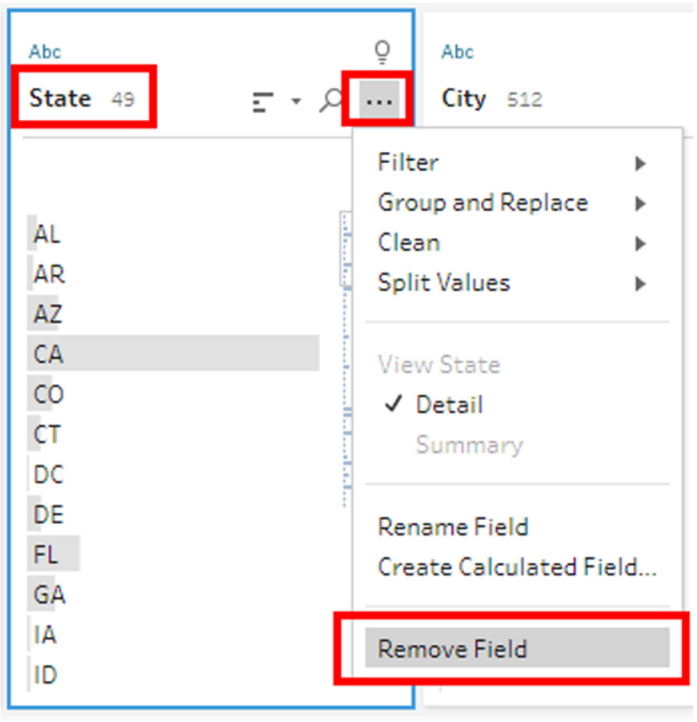


Figure 6.27a – Removing duplicate fields

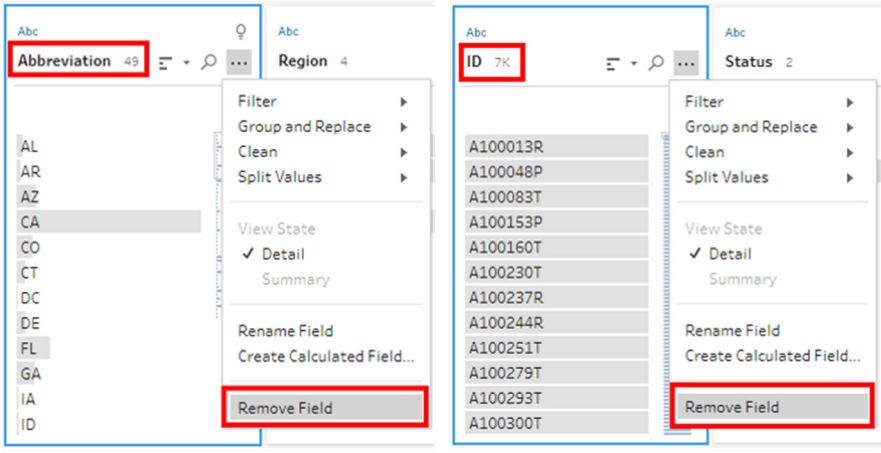


Figure 6.27b – Removing duplicate fields

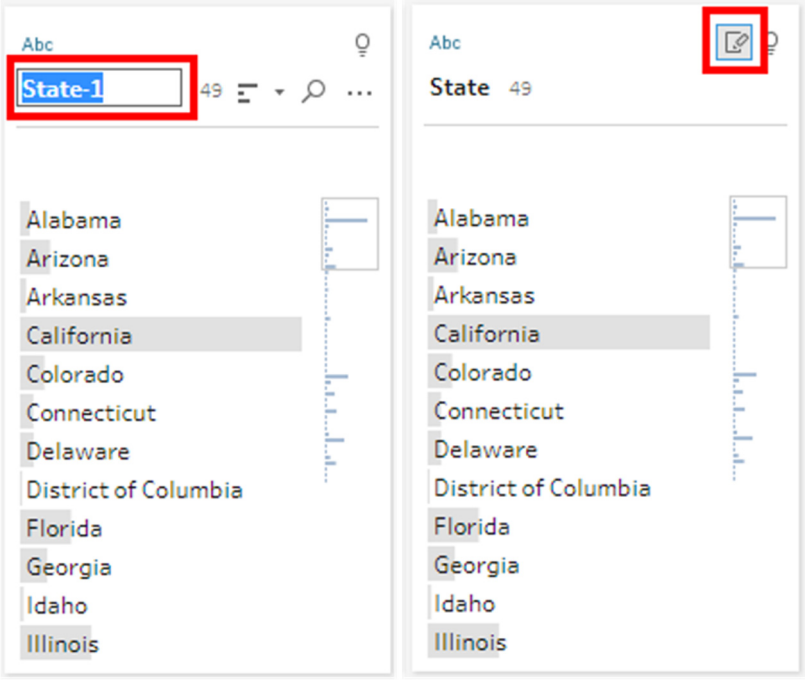


Figure 6.28 – Renaming the field

Finally, we will write the data into a .hyper format.

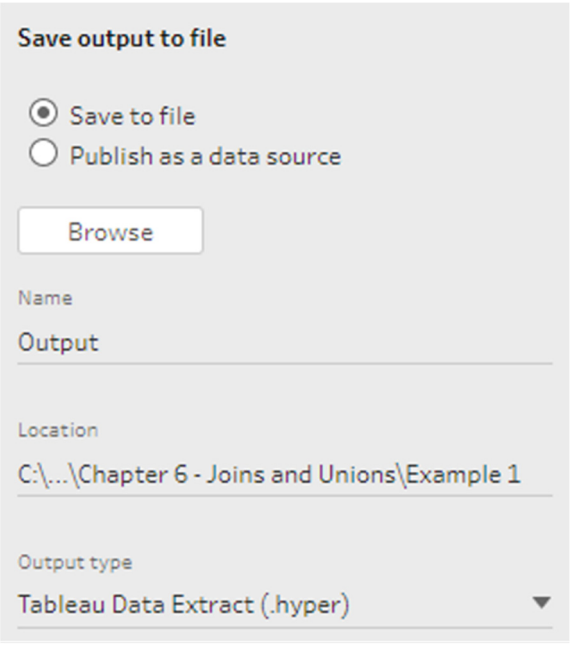


Figure 6.29 – Creating Output

Figure 6.30 – Final workflow

6.4 Union

A Union is an operation that appends multiple tables one below another to collect all information from various tables that might have similar information. Unlike a Join that creates wide tables, a Union creates long tables by adding rows.

When creating a Union in a database using SQL, the table schemas (i.e., fields) from the various tables must match, and the position fields need to be same. If the structure and naming convention is not maintained, the SQL throws an error. So, in many cases, we rename the field and create the Select statement in the required order of fields.

But the advantage in Tableau Prep Builder (or Desktop) is that we have a flexibility of renaming fields and merging all the mismatched columns. Also, by default, Tableau Prep Builder will create a *Union All*, which keeps all duplicates. If needed, the duplicates can be put together in the Aggregate step.

To understand the mechanism of Unions in Prep, let us consider three simple tables. All the tables will have census population for US states, segmented by Regions.

State	Popln.	Region
California	37.25 M	West
Washington	6.72 M	West
Oregon	3.83 M	West

Table 6.10 – West region

State Name	Popln.	Region
Texas	25.14 M	Central
Illinois	12.83 M	Central
Nebraska	1.82 M	Central
Minnesota	5.3 M	Central

Table 6.11 – Central region

State	Tot. Popln.	Region
New York	19.37 M	East
New Jersey	8.79 M	East
Florida	18.8 M	East

Table 6.12 – East region

As we can see though the fields have similar features, the headers are different. When we create a Union, the headers that do not match will have their own columns

State	Popln.	Region	State Name	Tot. Popln.
California	37.25 M	West	*Null*	*Null*
Washington	6.72 M	West	*Null*	*Null*
Oregon	3.83 M	West	*Null*	*Null*
Null	25.14 M	Central	Texas	*Null*
Null	12.83 M	Central	Illinois	*Null*
Null	1.82 M	Central	Nebraska	*Null*
Null	5.3 M	Central	Minnesota	
New York	*Null*	East	*Null*	19.37 M
New Jersey	*Null*	East	*Null*	8.79 M
Florida	*Null*	East	*Null*	18.8 M

Table 6.13 – Union result

With Tableau Prep Builder, we can merge all the mismatched fields and rename them to suit the output. (The same feature is available in Tableau Desktop.) The advantage of Tableau Prep Builder is that we can Union different types of data sources.

State	Census Population	Region
California	37.25 M	West
Washington	6.72 M	West
Oregon	3.83 M	West
Texas	25.14 M	Central
Illinois	12.83 M	Central
Nebraska	1.82 M	Central
Minnesota	5.3 M	Central
New York	19.37 M	East
New Jersey	8.79 M	East
Florida	18.8 M	East

Table 6.14 – Union result after cleanup

6.5 Example 2

All the files required for this example are in Chapter 6 folder.

1. Trade transaction information for four years (2014–2017) is present in different tabs in the Excel file. This is a sample *Cancels and Corrections data.xlsx* file with some random values for four years.
2. US state names in full and their regional groups are also present in a tab in the same Excel file.
3. Trade status – Whether a particular trade underwent a correction or cancellation is also present in another tab in the same Excel file.

Requirement: Need an output that provides only trades that have been either cancelled or corrected along with the Region information and full names for all the states.

We can see that the requirement is the same. But we need to combine the four years of data as a Union and then apply the Join, as seen in the previous example. We will use the Example 2 folder.

For this example, we will use a simple wildcard—Union. For the next example, we will try a Union using different data files.

We will create a new workflow and start with the *Cancels and Corrections data.xlsx*.

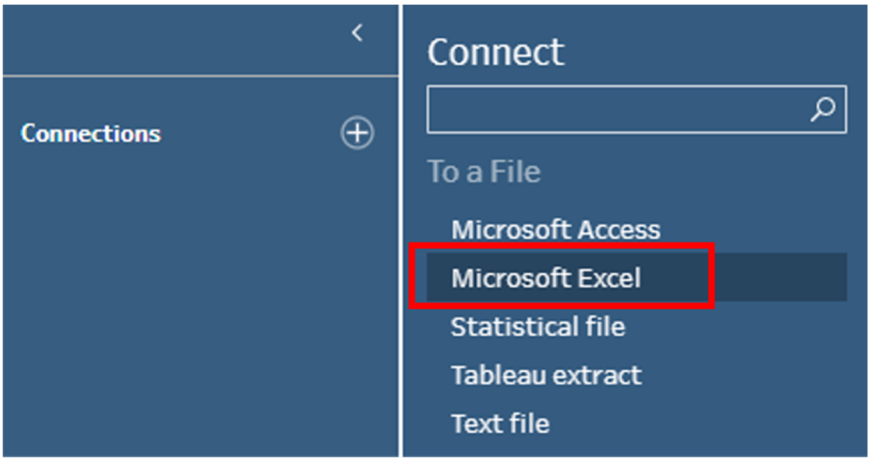


Figure 6.31 – Connection to .xlsx

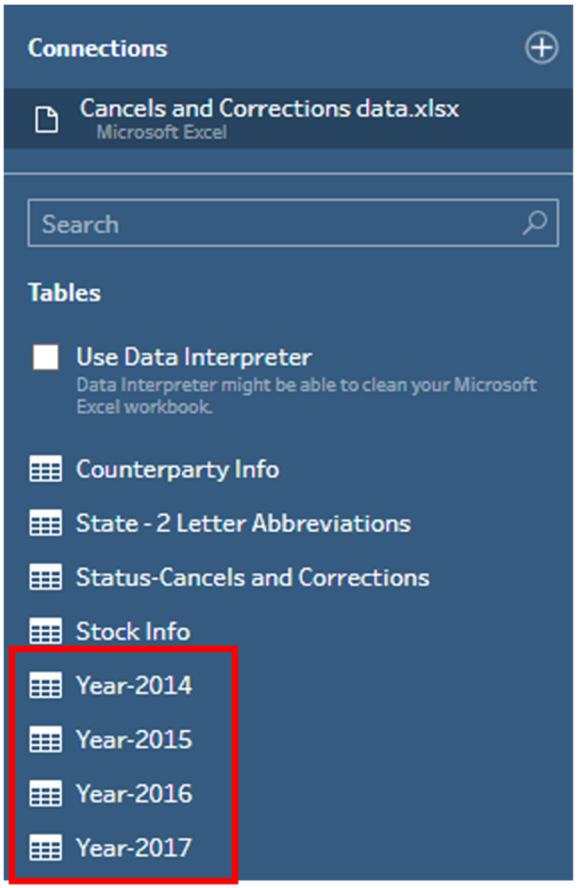


Figure 6.32 – Tabs in the Excel file

Here, we can see all eight tabs along with a Named Range table (Excel list). For the Union, we are interested in the four years.

We will start with one tab and use the wildcard option from the configuration window.

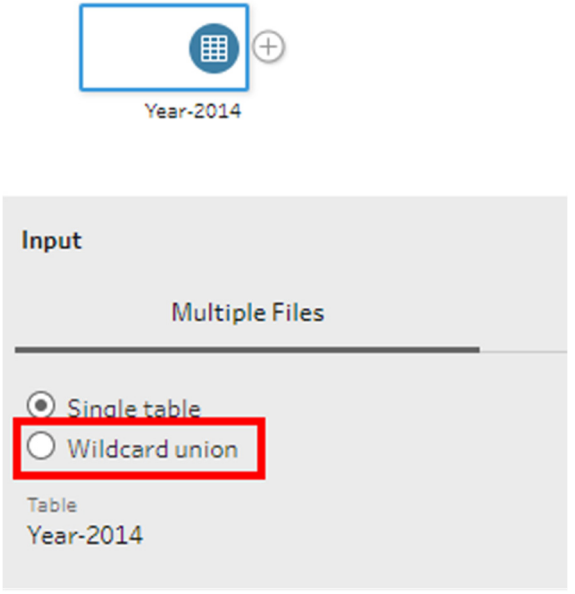


Figure 6.33 – Single tab

The configuration window has options to select the main folder. In this case, it is Example 2. We also have the option to select subdirectories (subfolders) where some more files can be stored.

> ***Note:** We have to remember that the main folders and subfolders must have the same type of file (i.e., we cannot select Excel files and text files in a wildcard Union).

We also have an option to either include or exclude certain types of files based on the search pattern.

The wildcard Union can also be used for individual files in the same folder with the same pattern.

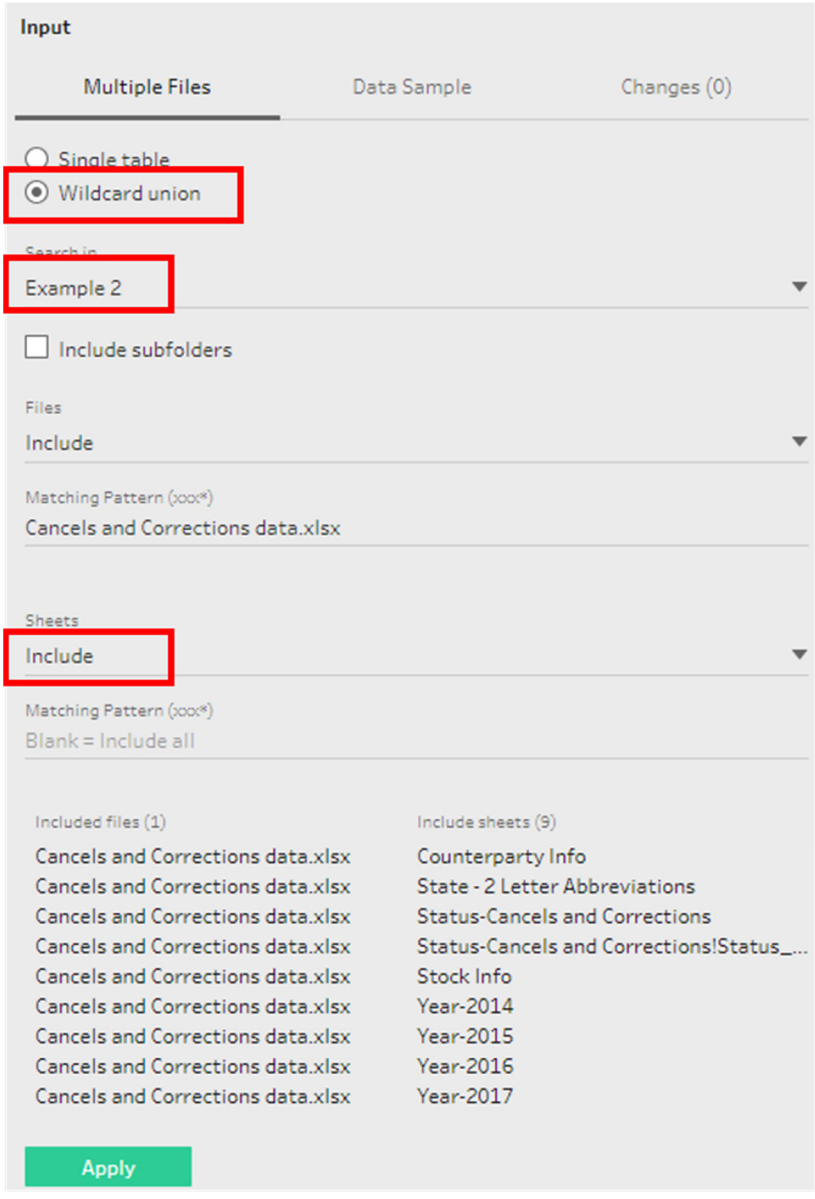


Figure 6.34 – Wildcard union configuration

Right below the configuration, we can see all the available sheets in this Excel file.

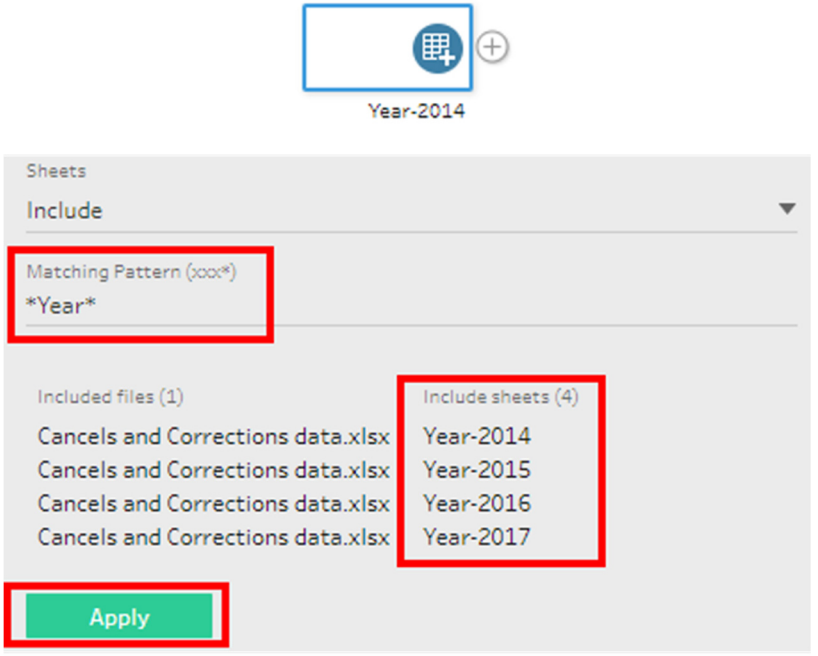


Figure 6.35 – Matching pattern search

We are searching for *Year*. So, anything that matches these characters will be displayed in the list. We can click *Apply* to use the required sheets. All these files will be put into a Union. When the Union is created, we will notice a small + in the data connection step to indicate that there is more than one table.

We can save the workflow as *Example 2 Workflow*. We can add a cleaning step to see the fields and the underlying data. We can see the last two fields showing the path and the files being used. We can remove them if not needed in the output.

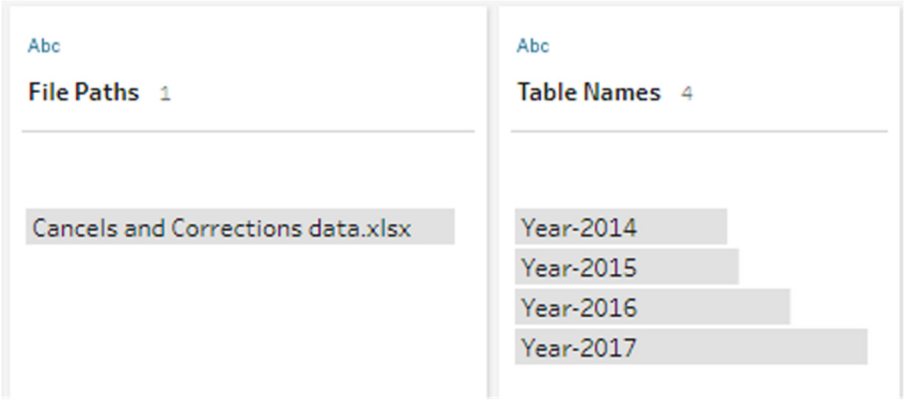


Figure 6.36 – File Paths and Table Names

Here we have to be a little cautious. As mentioned before, the wildcard Union works very well if the schema matches in all tables (i.e., the fields are in the same position and named the same). Whenever there is a mismatch, we encounter Null and duplicate fields. Repairing such issues will be explained in the next example.

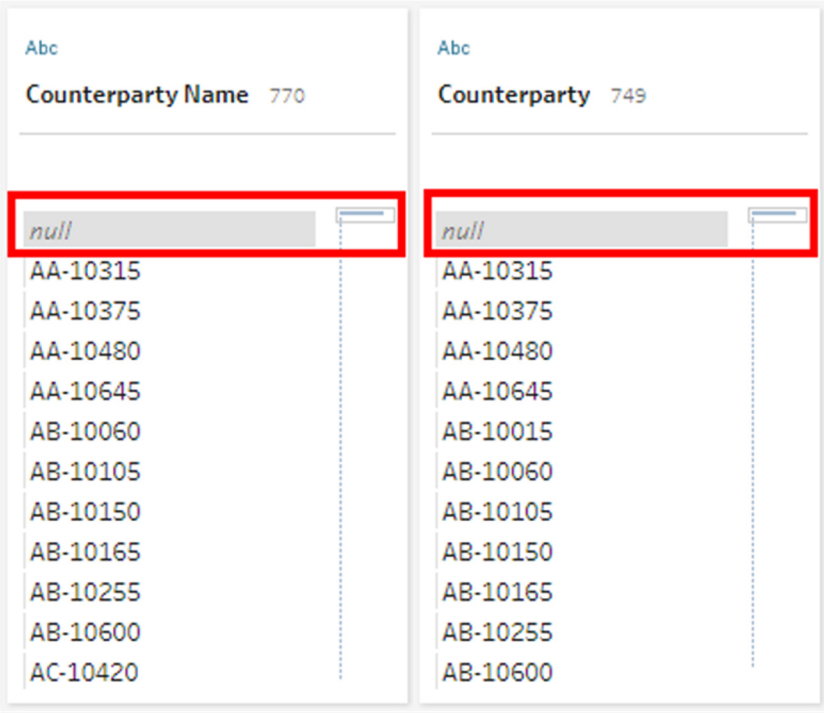


Figure 6.37a – Mismatches and Nulls

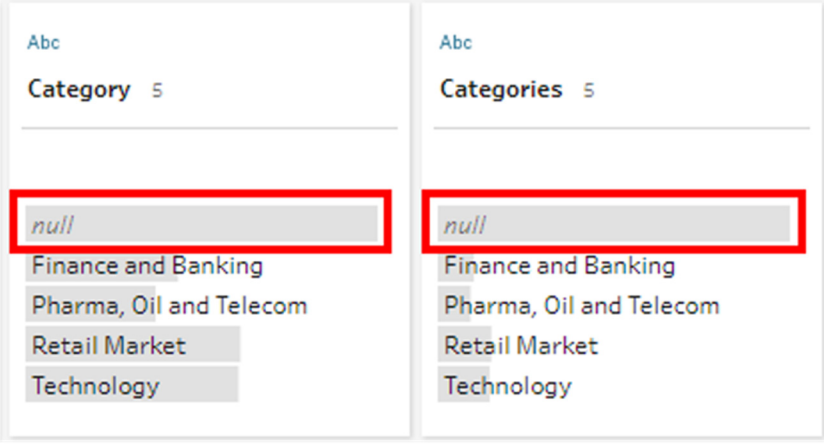


Figure 6.37b – Mismatches and Nulls

Now that the Union is ready, we are left with two Joins. As in the previous example, one is to replace the State Abbreviation, and the other is to get the Status of Trade.

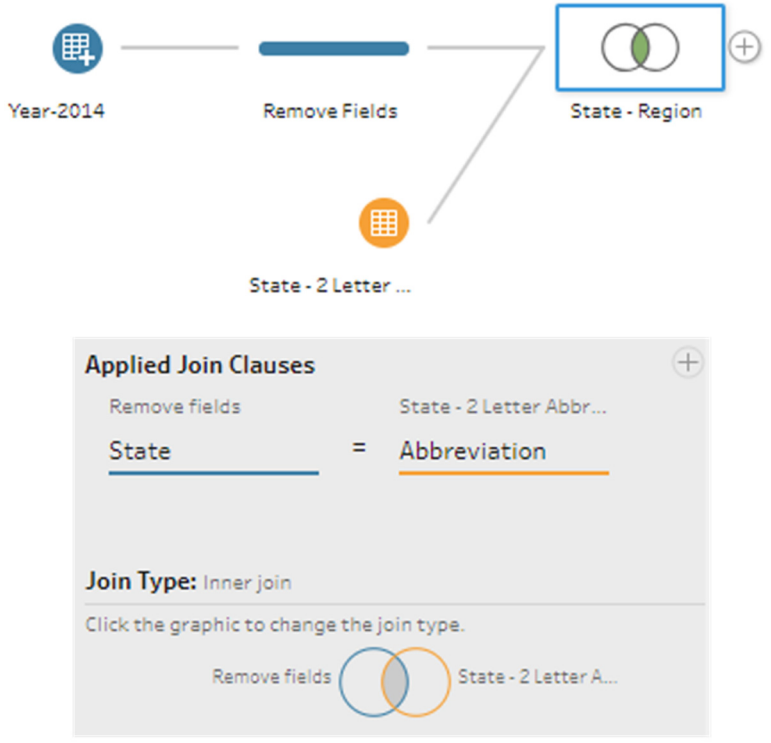


Figure 6.38 – Join for State info

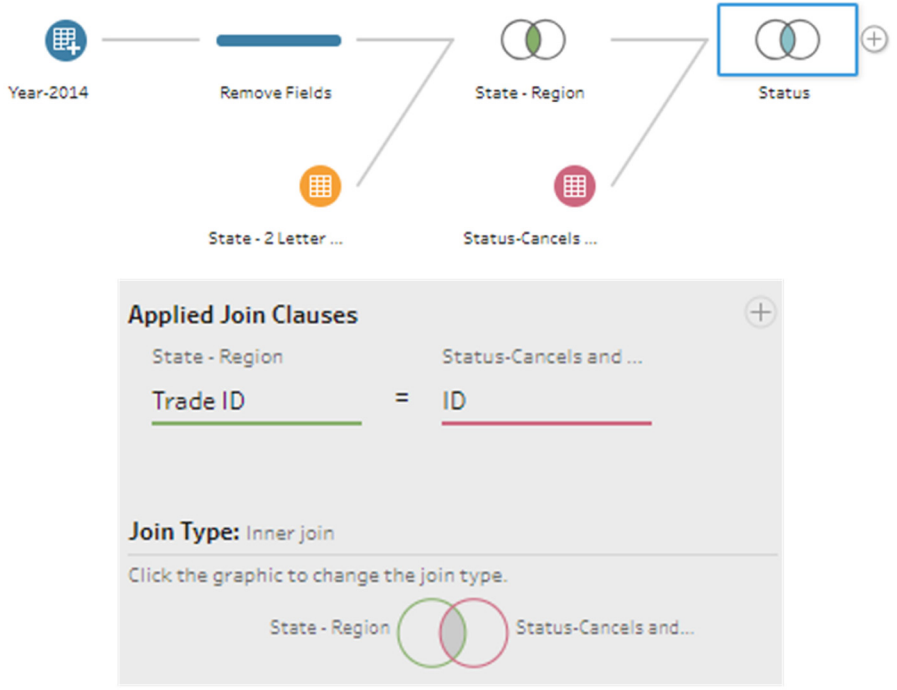


Figure 6.39 – Join for Status info

If needed, we can add another cleaning step to remove redundant fields and rename some of the new ones. Then we can write the file to the file or the Tableau Server.

Figure 6.40 – Final workflow

6.6 Example 3

All the files required for this example are in Chapter 6 folder.

1. Trade transaction information for four years (2014–2017) is present in different files.
2. We have sample cancels and corrections data in two .csv files for 2014 and 2015 and two Excel files for 2016 and 2017.
3. State names in full for the United States and their regional groups are in the *State – 2 Letter Abbreviations.xlsx* file.
4. Trade status – Whether a particular trade underwent a correction or cancellation is in the *Status – Cancels and Corrections.txt* file. This is a tab-delimited file.

Requirement: Need an output that provides only trades that have been either Cancelled or Corrected, along with Region information with full names for all the States.

This is the same requirement, but the approach will be a little different. The .csv files and Excel files have the same fields. But the field names are different. The fields are Index [named Key], Postal Code [named Zip Code], and Trade Volume (Units) [named Volume]. So, this example will help us identify and repair the fields that do not match. We will use the Example 3 folder.

For this example, we will use four input steps, one for each file.

Let us create a new workflow and start with the *Cancels and Corrections - 2014.csv* and *Cancels and Corrections - 2015.csv.* Instead of bringing the files one by one, we will use the CTRL key

and select. That way, multiple individual files can be brought into the workflow.

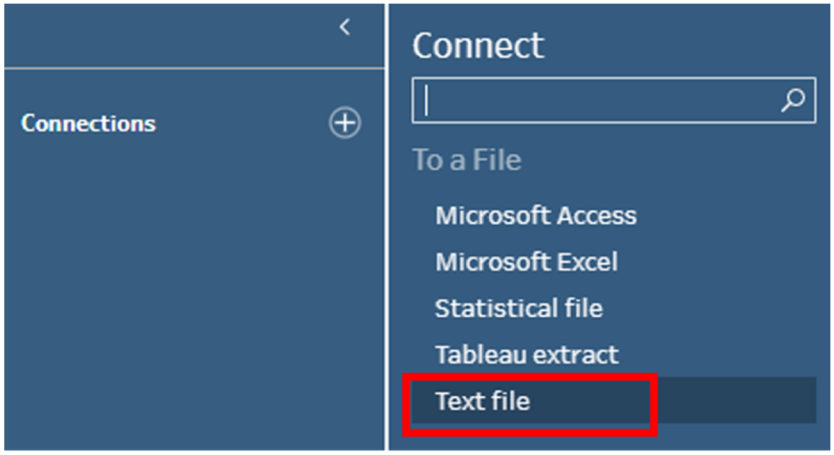


Figure 6.41 – Connection to .text (.csv)

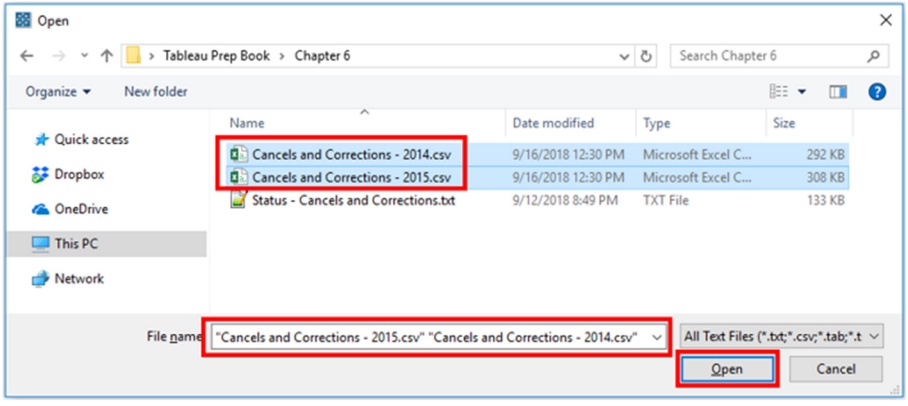


Figure 6.42 – Using CTRL key – Multiple file selection

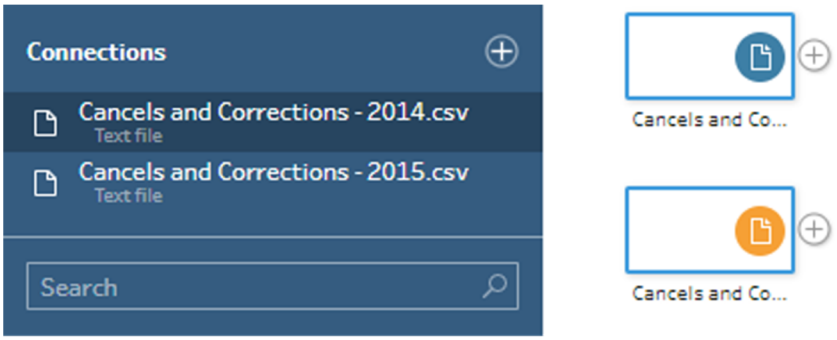


Figure 6.43 – Two Input steps – Auto created

We should also notice that when these files were brought in, Tableau Prep Builder identified *comma* as a delimiter.

In the same manner, we will bring *Cancels and Corrections - 2016.xlsx* and *Cancels and Corrections - 2017.xlsx*.

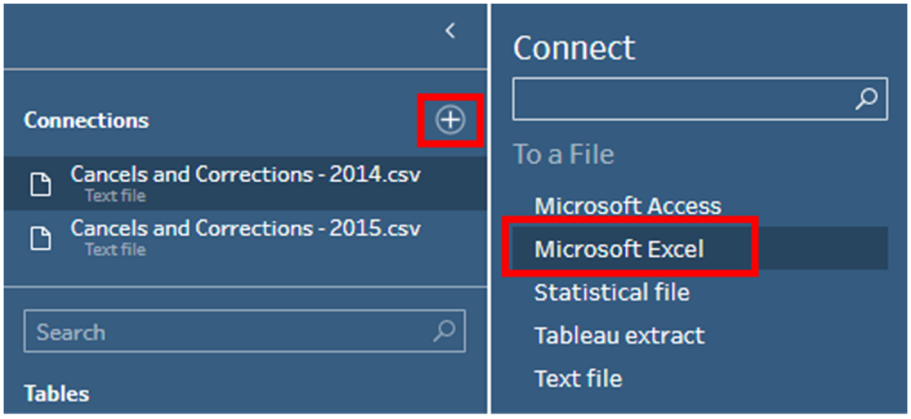


Figure 6.44 – Connection to .xlsx – Adding a new source

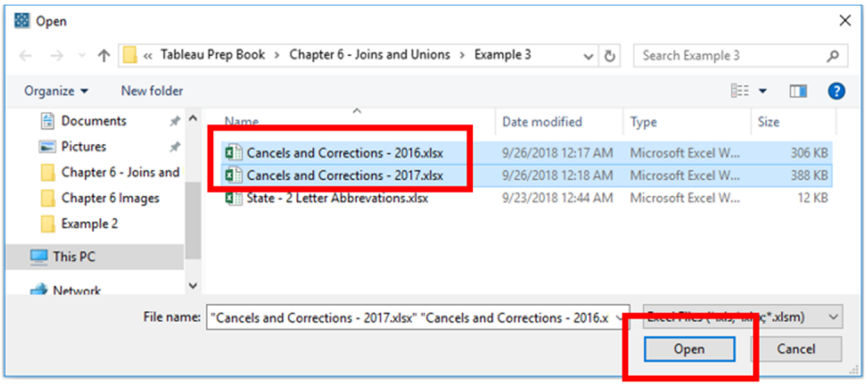


Figure 6.45 – CTRL key – Multi file selection

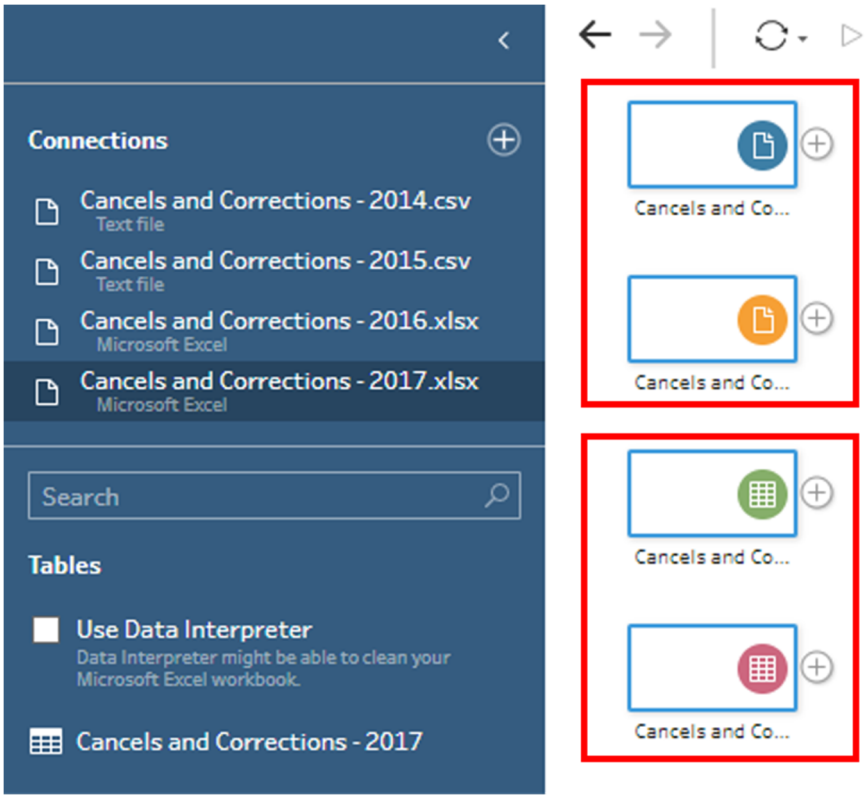


Figure 6.46 – All Inputs in the same workflow

The two .csv files have the same structure, and the two Excel files have the same structure. So, we will go for two Unions and bring them together into a final Union.

Like the Join explained before, we can create a Union by Adding a step or dragging and dropping tables over each other. Unlike Joins, we can Union multiple files at once.

Figure 6.47 – Union 1 for CSV files

The Union shows that all the fields are matched and color-coded for easy field identifications.

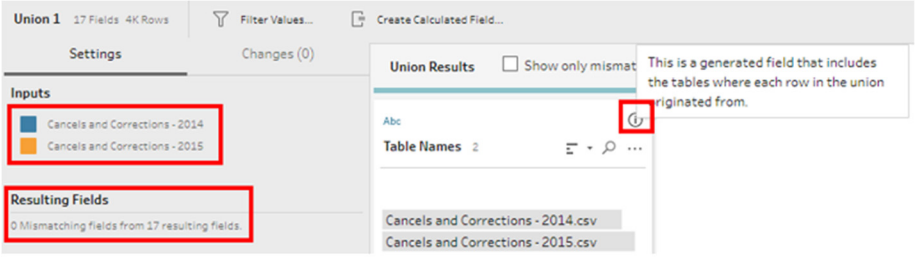


Figure 6.48 – Union 1 results

We can see a small cue (the information symbol) that says that the *Table Names* field is auto-generated.

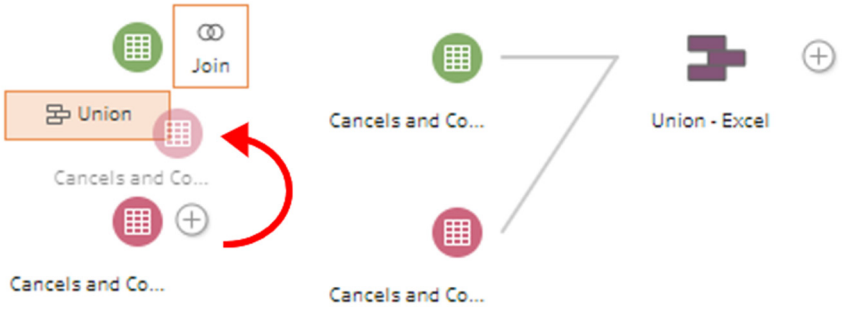


Figure 6.49 – Union 2 for Excel files

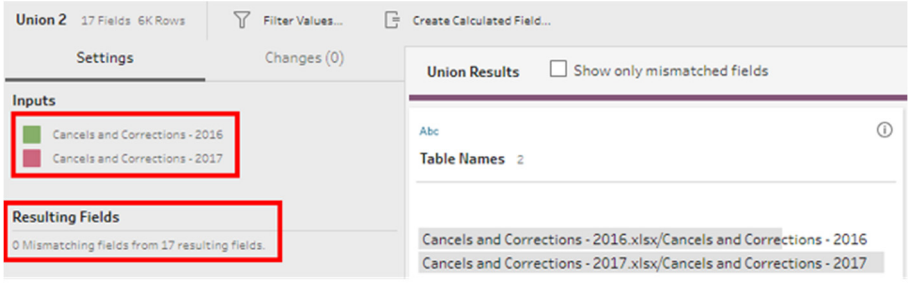


Figure 6.50 – Union 2 results

This Union also shows that all the fields have matched.

Before we proceed to the next Union step, we will name the workflow *Example 3 Workflow*.

For the final Union, we will append all values from Union 1 and Union 2.

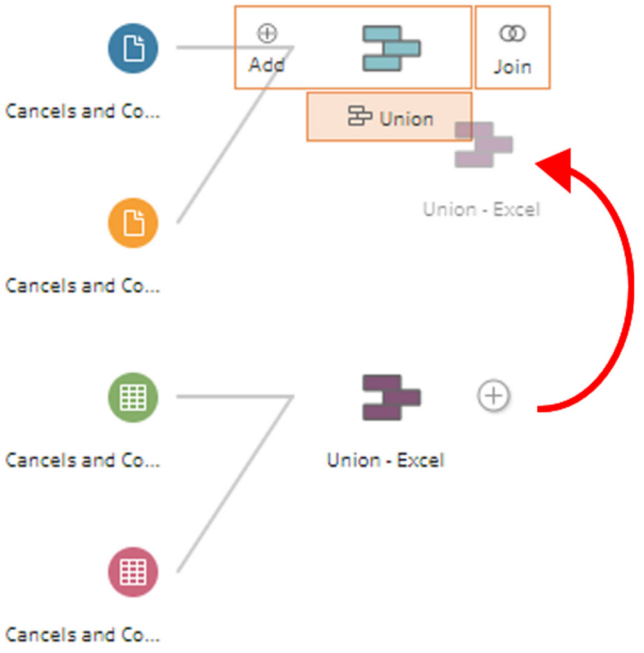


Figure 6.51 – Union of both results

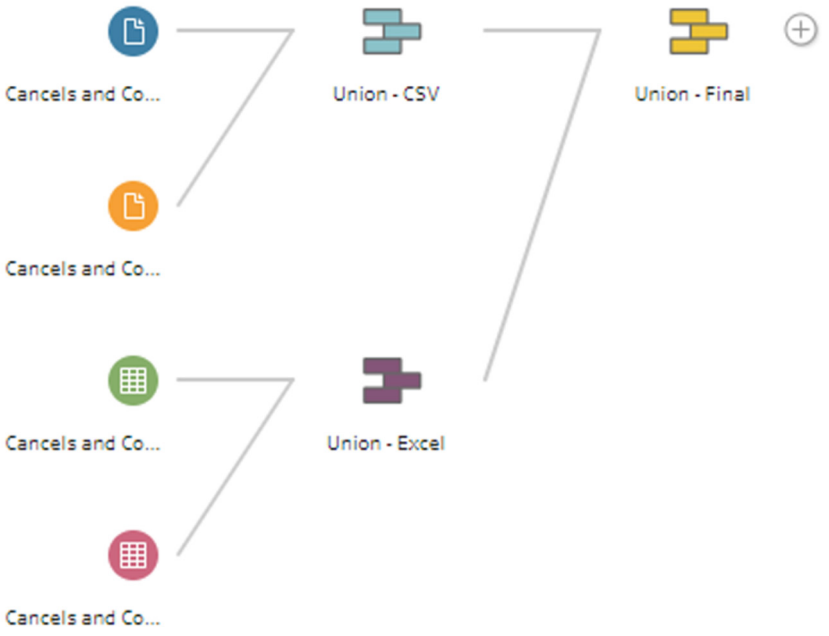


Figure 6.52 – Workflow after the final Union

As soon as the Union is created, Prep was smart enough to identify the mismatched fields in the configuration window. That can mean one of the two things—either the field is not named the same, or the field is missing from the data.

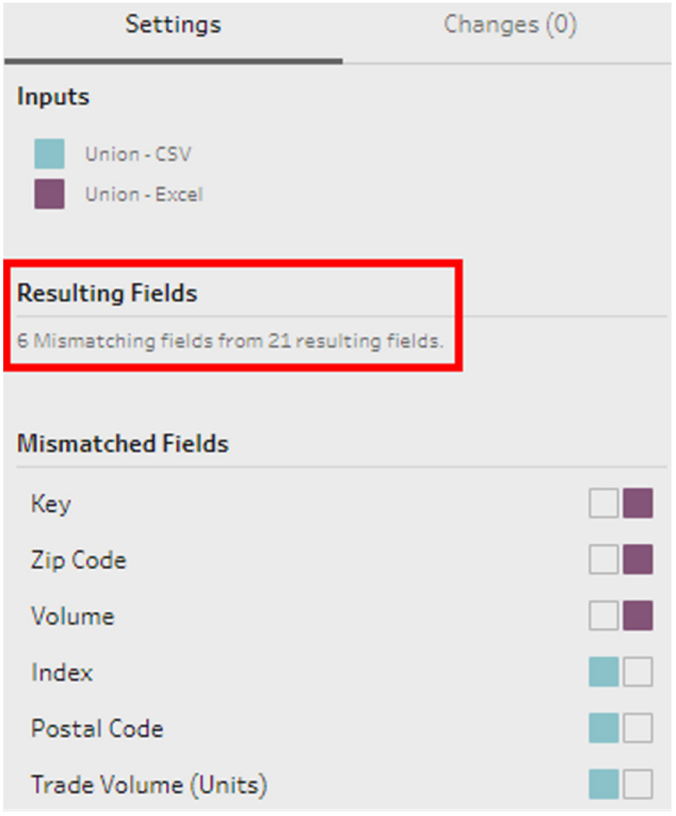


Figure 6.53 – Mismatched fields

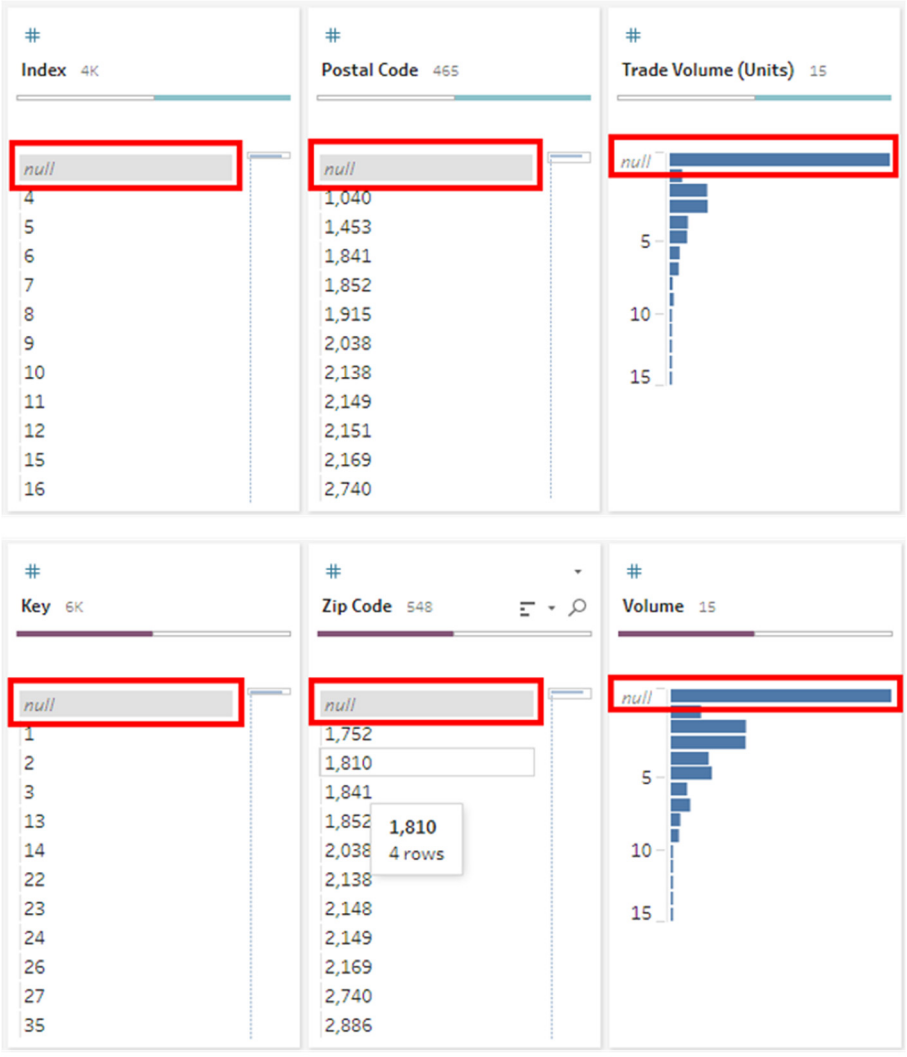


Figure 6.54 – Mismatched fields with Null entries

Now it is time to repair. We can repair it in two ways. We can make changes from the Profile pane of the Union step, or we can

add a Cleaning step and make changes. This step is similar to the one used in Tableau Desktop.

For this example, we will use the settings offered by the Union step.

First, we need to focus only on the mismatched fields. Since we have many fields, scrolling left and right is a little difficult.

So, we will use the checkbox to show only the required fields.

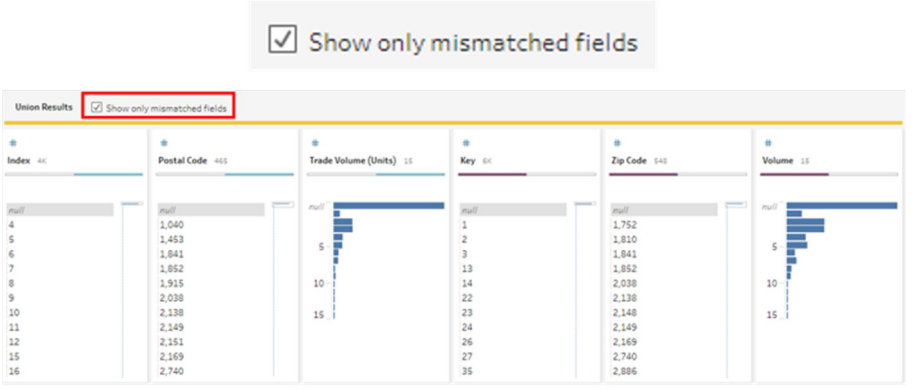


Figure 6.55 – Displaying only the mismatched fields

We can merge the mismatched fields by selecting multiple fields using the CTRL key. Then we can right-click on one of the fields or click on the dropdown to merge fields.

Figure 6.56 – CTRL key – Multi field selection to merge

We can use the easy drag-and-drop method where one field is dragged on top of another matching field.

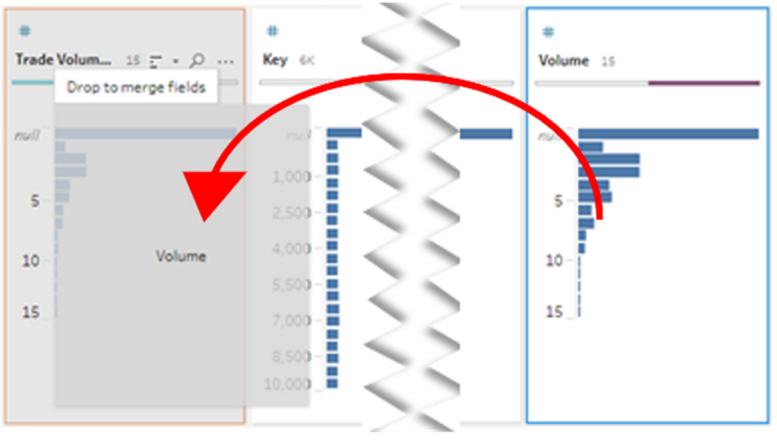


Figure 6.57 – Drag and drop fields to merge

After the fields are merged, we can rename the fields. When we use the CTRL key selection method, the new field name will be taken from the second field. When we drag and drop, the new field name will be taken from the first field.

Now that all the fields are ready, we need to join the State information and also join the trade status information. Then we can add a cleaning step. All steps are similar to the previous examples.

After that, we can write the data into an output file (or publish to Tableau Server).

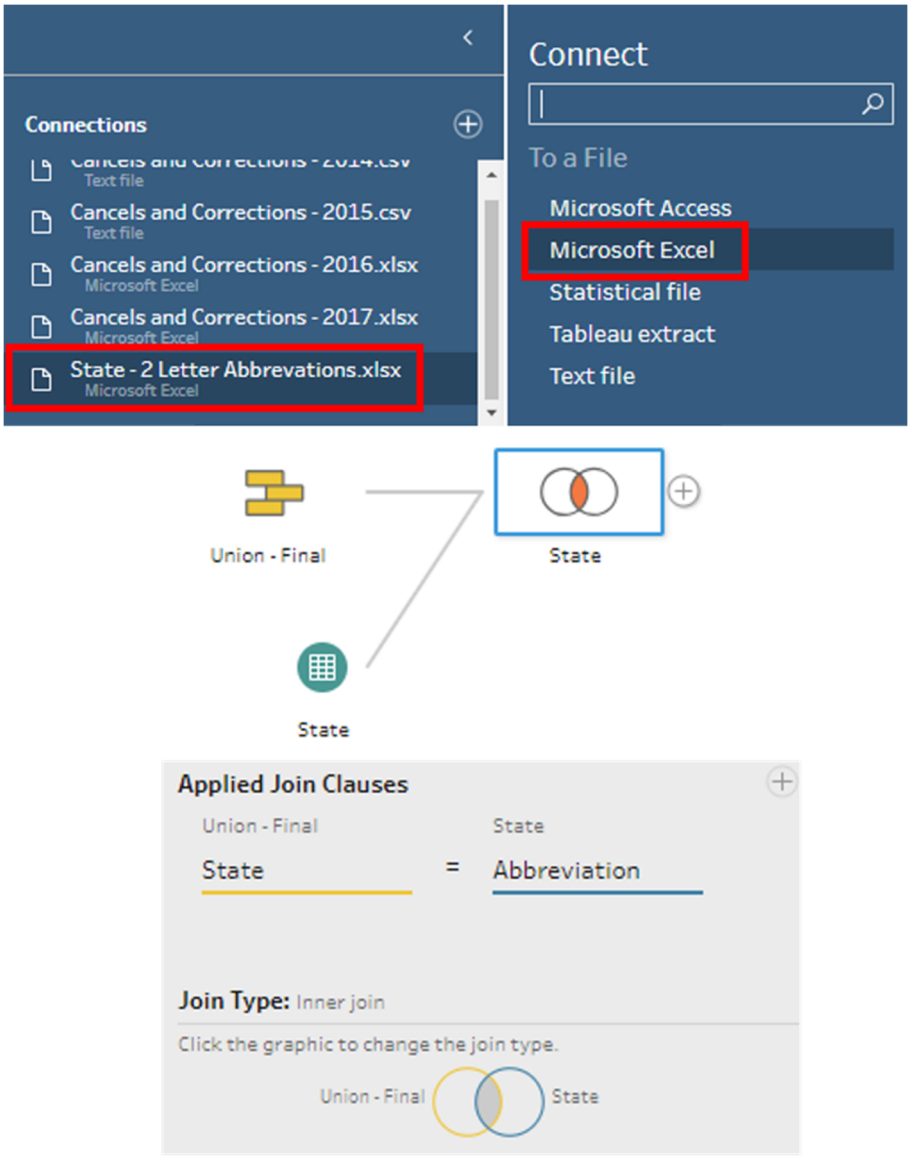


Figure 6.58 – Join for State information

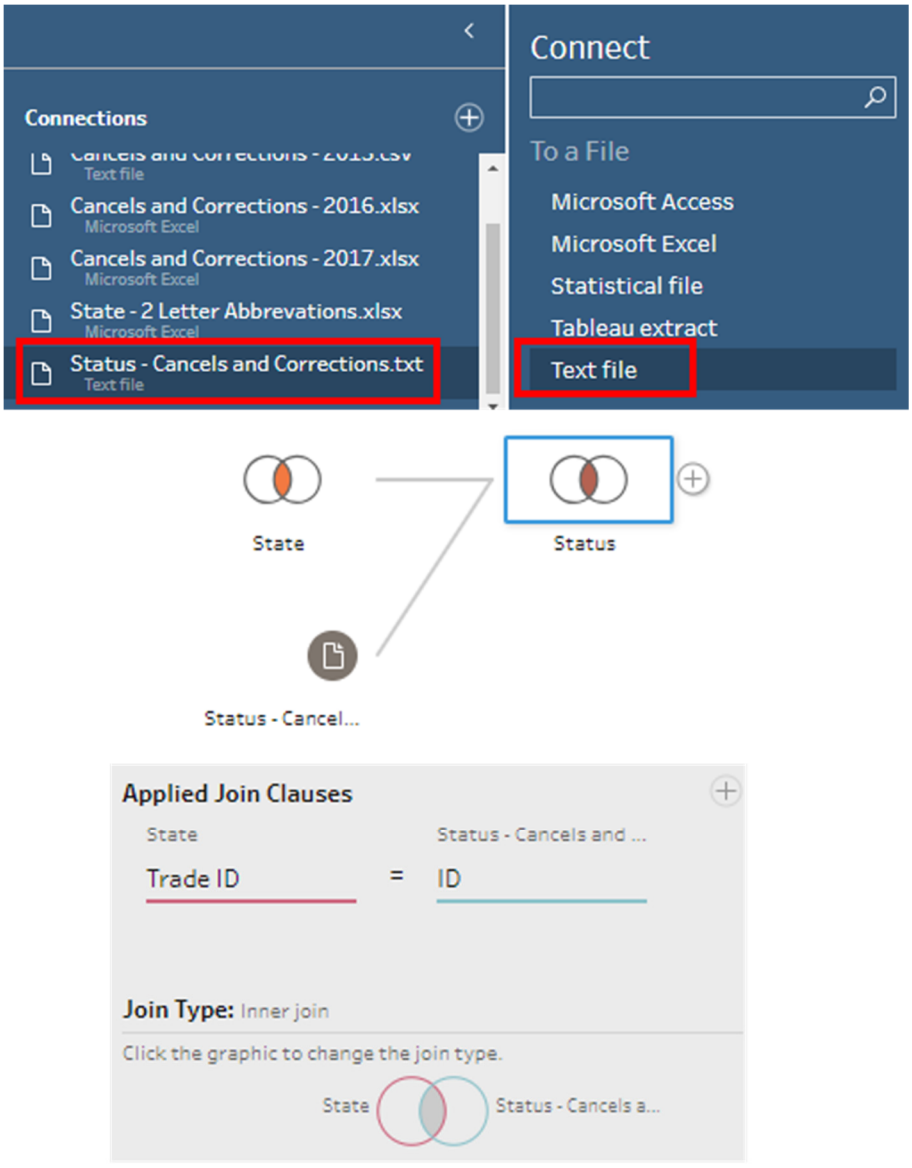


Figure 6.59 – Join for Status information

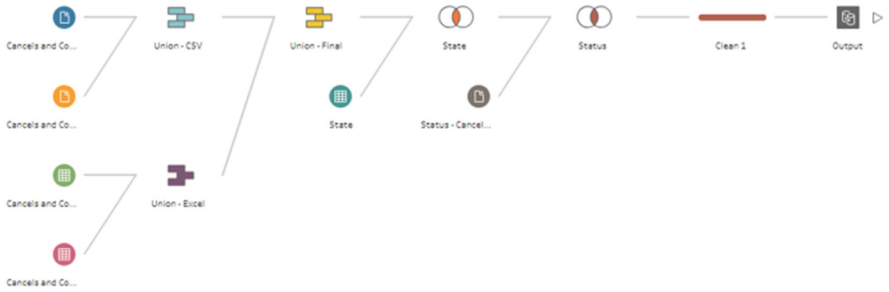


Figure 6.60 – Final workflow

CHAPTER 7

TABLEAU PREP BUILDER CONDUCTOR

7.1 Tableau Prep Builder Conductor

Tableau Prep Builder Conductor is an add-on to Tableau Server and Tableau Online. It lets us schedule and run Tableau Prep Builder flows in a scalable, reliable, and secure server environment. It will be available as a separate license that requires an already licensed Tableau Online or Tableau Server environment.

With this add-on, we can configure Servers and Sites to use Tableau Prep Builder Conductor, publish flows from Tableau Prep Builder Desktop to Tableau Server, manage the permissions and ownership for our Tableau Prep Builder Flows on the server, and create schedules for published flows to run on a regular basis to keep underlying data always up-to-date. The Output of this workflow would be an extracted data source.

7.2 Prerequisites

Set node role feature flag to true. *This is set to true by default*
tsm configuration set -k features.TsmCliSetNodeRole -v true.

Enable node role to run/schedule flows on Tableau Server. *This is enabled by default.* It enables Tableau Prep Builder Conductor service on the Tableau Server.
tsm topology set-node-role -n node1 -r flows – to enable flow jobs
tsm topology set-node-role -n node1 -r no-flows – to disable flow jobs

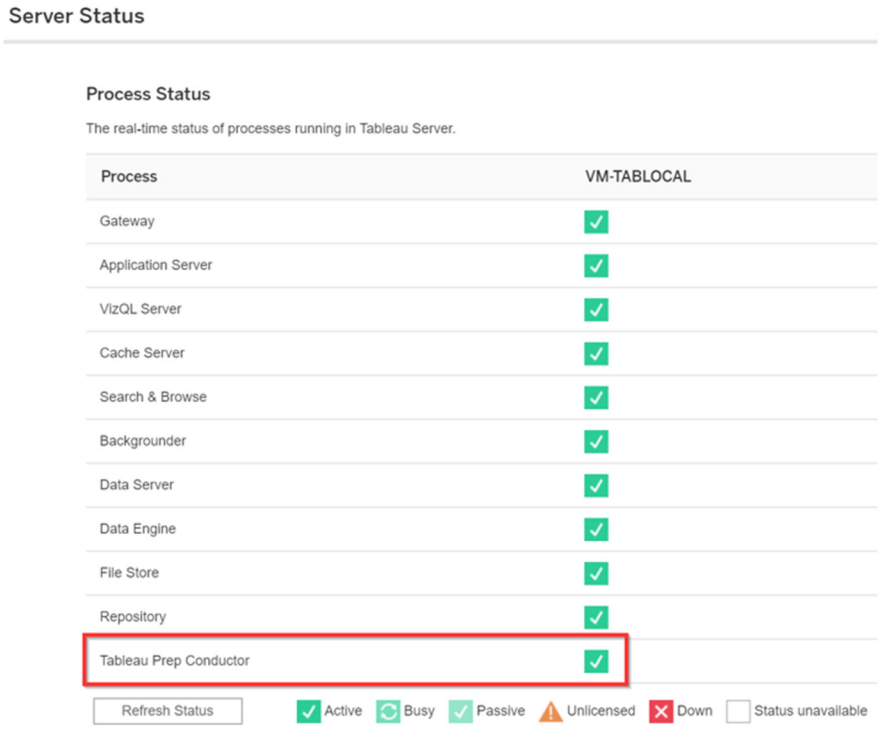


Figure 7.1 – Tableau Server Status

***Note**: There has to be at least one Tableau Prep Builder Conductor in the Tableau installation. The Tableau Prep Builder Conductor can be removed if Server has a multi-node cluster and there is at least one node that is set to allow all types of tasks or specifically set to allow flow runs.

To get a flow execution failure e-mail alert, we need to get SMTP settings and alerting configured on Tableau Server and notification enabled under Site Settings.

Users need to request the Tableau Server (not site) administrator to create a new schedule for Prep workflows based on the required time window.

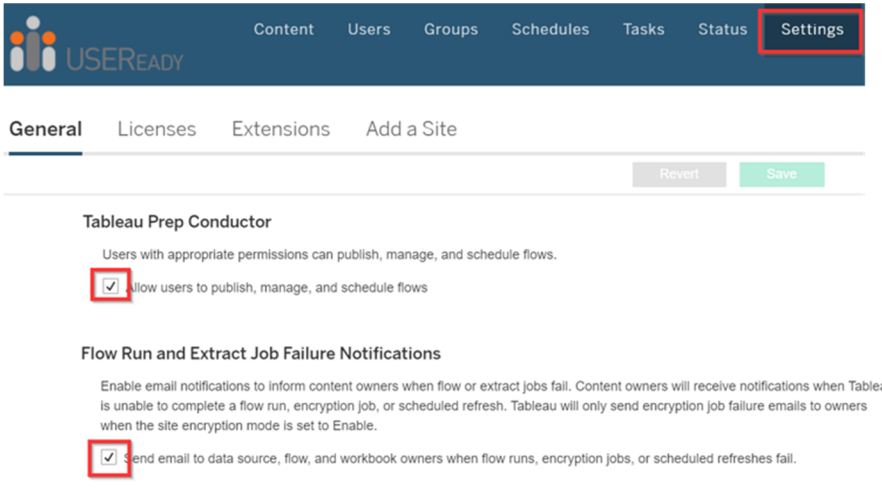


Figure 7.2 – Site Setting to enable Flow run/notification

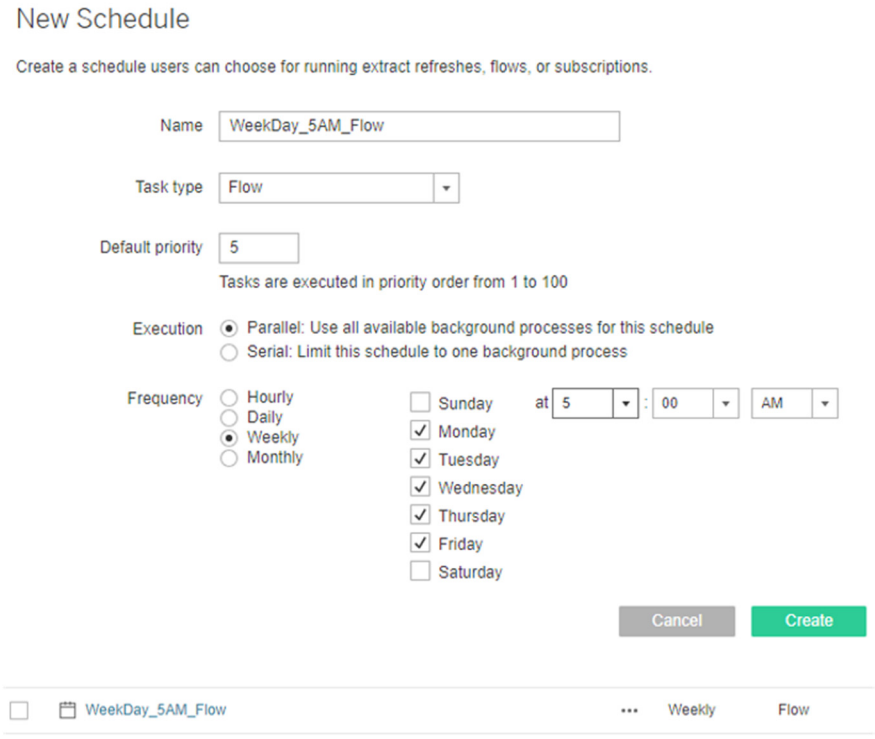


Figure 7.3 – Creating schedule on Tableau Server

7.3 Publishing Workflows to Server

Let us open the workflow used in Chapter 1 – *Chapter 1 Workflow.tflx*.

***Note:** As mentioned in Chapter 1, workflow publishing is possible only with Prep Builder version 2019.1 and above.

Now it is time to sign in to Tableau Server from the Tableau Prep Builder application.

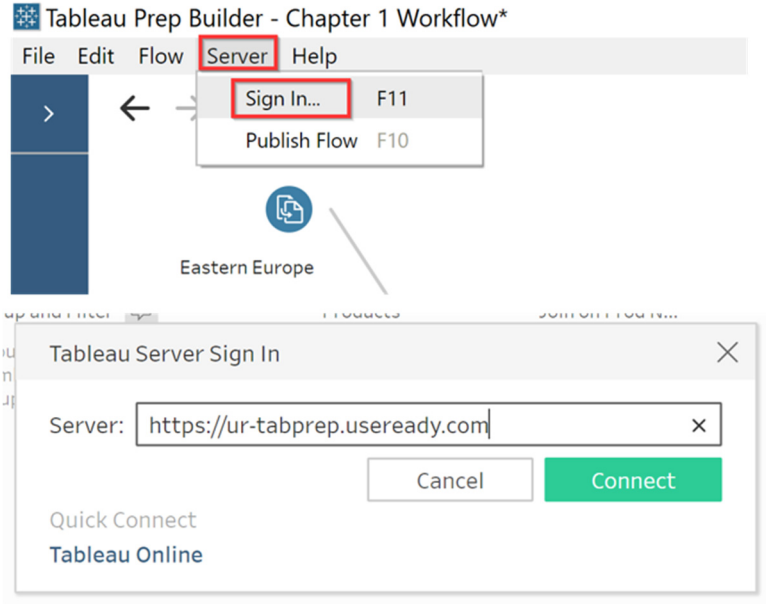


Figure 7.4 – Login to Tableau Server/Site

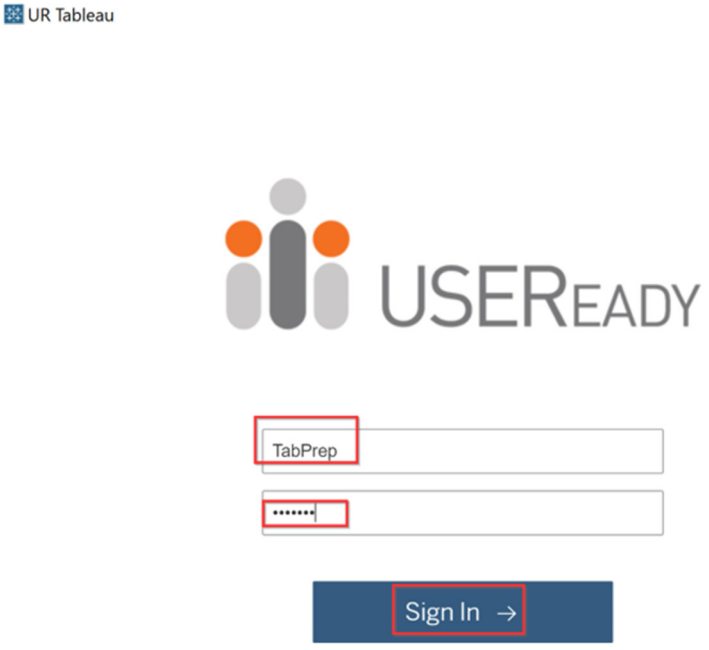


Figure 7.5 – Login to Tableau Server/Site

Once we have successfully logged in to Tableau Server, the next step would be to publish the workflow to Server.

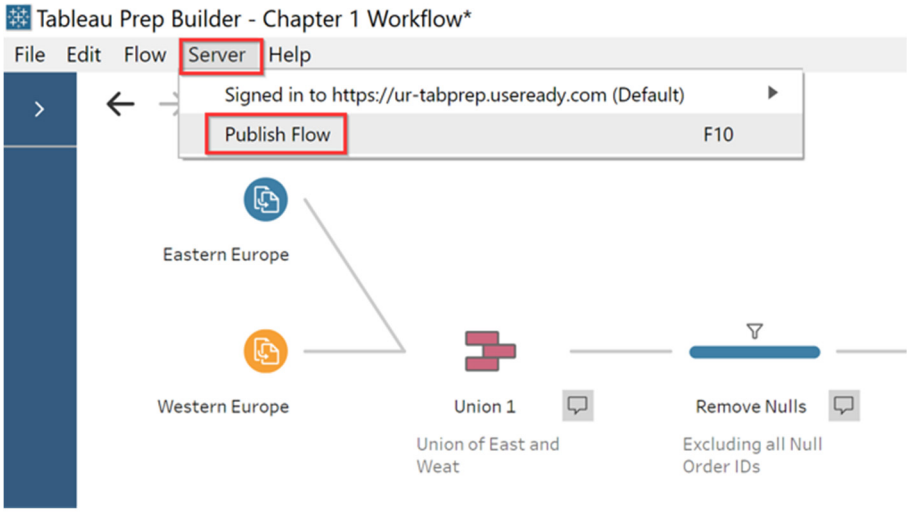


Figure 7.6 – Publishing Flow

We need to select an appropriate project for which we have permission to publish, provide details such as Name and description, and select all the flat files to be uploaded to a server that are used by the flow. If these files are sourced from a network drive using the UNC path, then none of the files need to be selected.

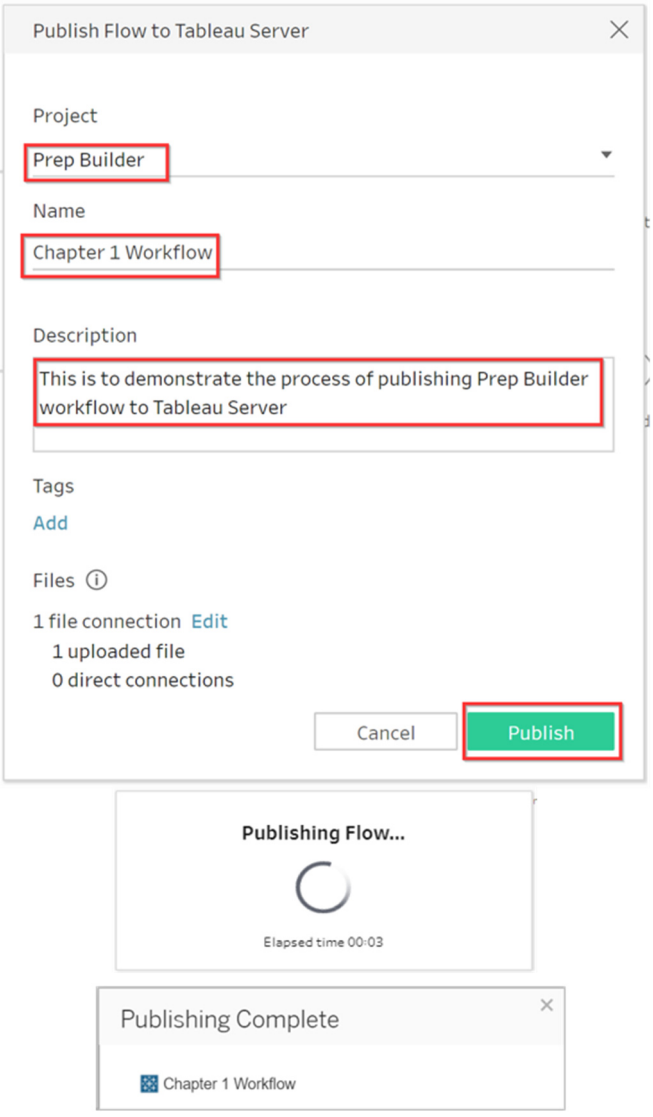


Figure 7.7 – Publish workflow

7.4 Scheduling Workflows on Server

Once the publish process completes successfully, it would automatically take us to the Tableau Server portal where we can schedule this workflow to run automatically based on pre-defined schedule.

We have to click on "+ Create new task" to start attaching a schedule for running a specific workflow.

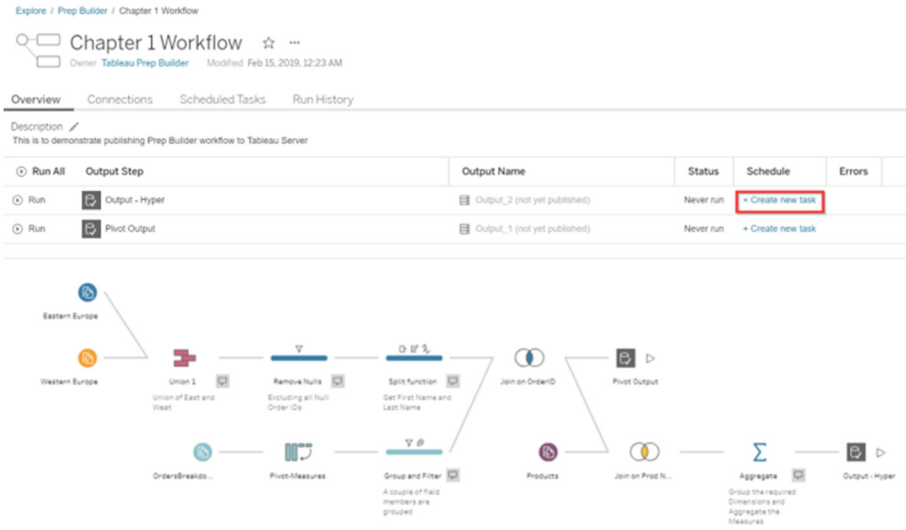


Figure 7.8 – Publish workflow

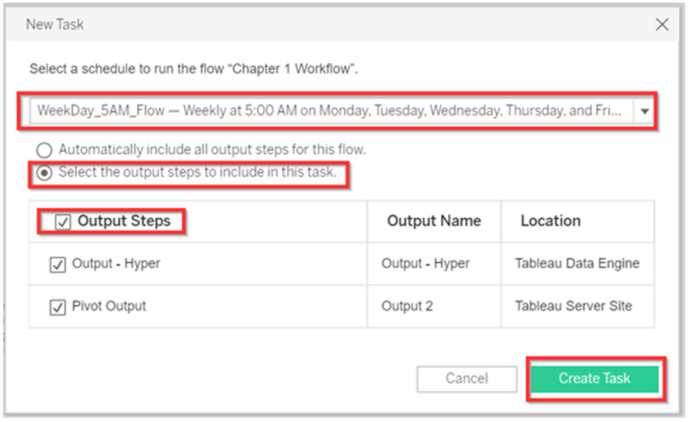


Figure 7.9 – Publish workflow

In the pop-up window, we have to select appropriate schedule, select output steps (if workflows have multiple output options), and click on *Create Task*, as highlighted above.

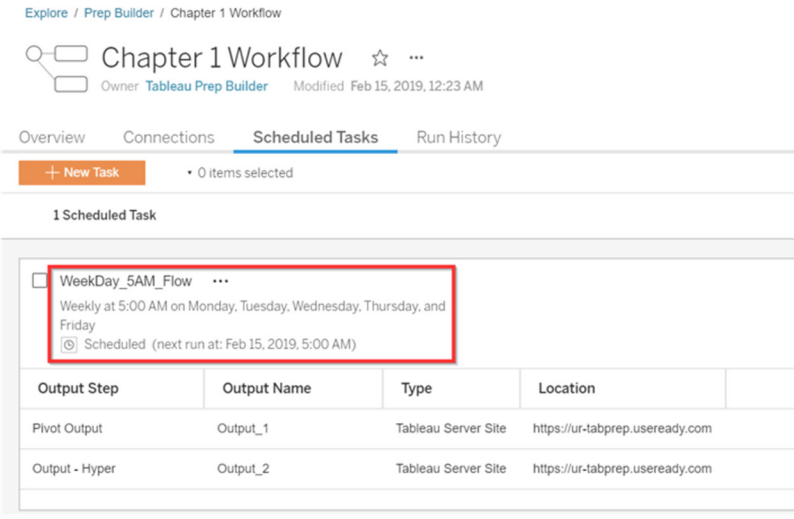


Figure 7.10 – Publish workflow

We can verify the new task that is created to run the flow based on the selected schedule.

To make sure everything works fine on Tableau Server, click *Run Now*.

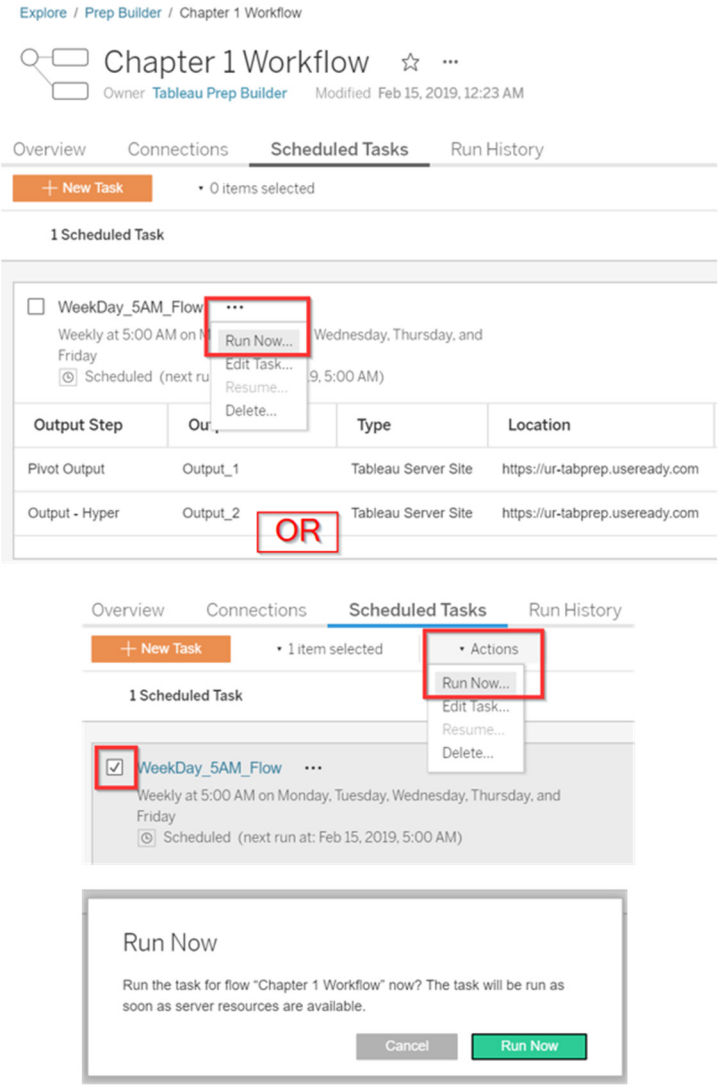


Figure 7.11 – Run workflow on-demand

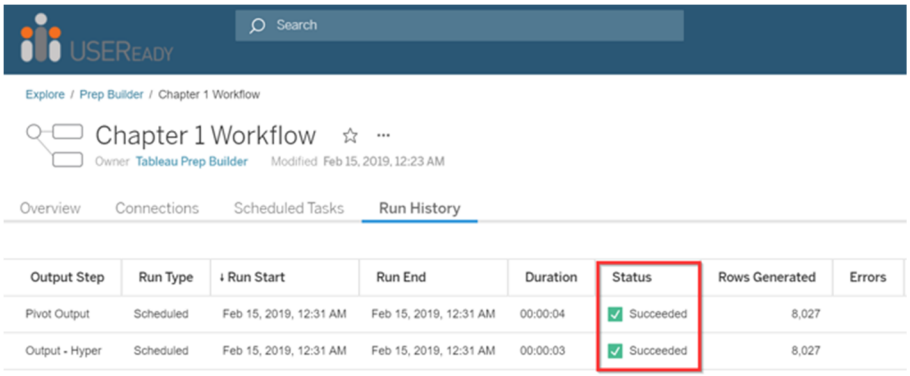


Figure 7.12 – Workflow execution result

The above steps demonstrate using Tableau's new feature of *Tableau Prep Builder Conductor*.

USEREADY

Jeff Black, Tableau

For those wanting to learn Tableau Prep, this practitioner's guide—written by USEReady Tableau consultants—will build the foundation needed, whether you're an individual in business, a journalist who reports data findings, or a student at an educational institution. Given their experience as a certified training partner and collaborating with enterprises on strategic data projects, USEReady is positioned to help you succeed with data.

Jeremy Walsh, Tableau

Now, more than ever, there is a demand to empower users to clean, combine, aggregate, and prepare data for analysis. USEReady is a leader in this discipline and has once again created a training manual not only to help users adopt Tableau Prep quickly and easily but to further Tableau's mission: Help People See and Understand Data.

Authored By

Akshay Venkatesh | Anirban Ghosh | Sachet Kashyap | Srini Ramanathan | Moyses Ramirez

Rahul Shetty | Rajesh Hegde | Anish Bhola | Fred Ferrari | Vijai Narasimha

ISBN 978-1-64516-475-3

GET IN TOUCH : www.useready.com | contact@useready.com