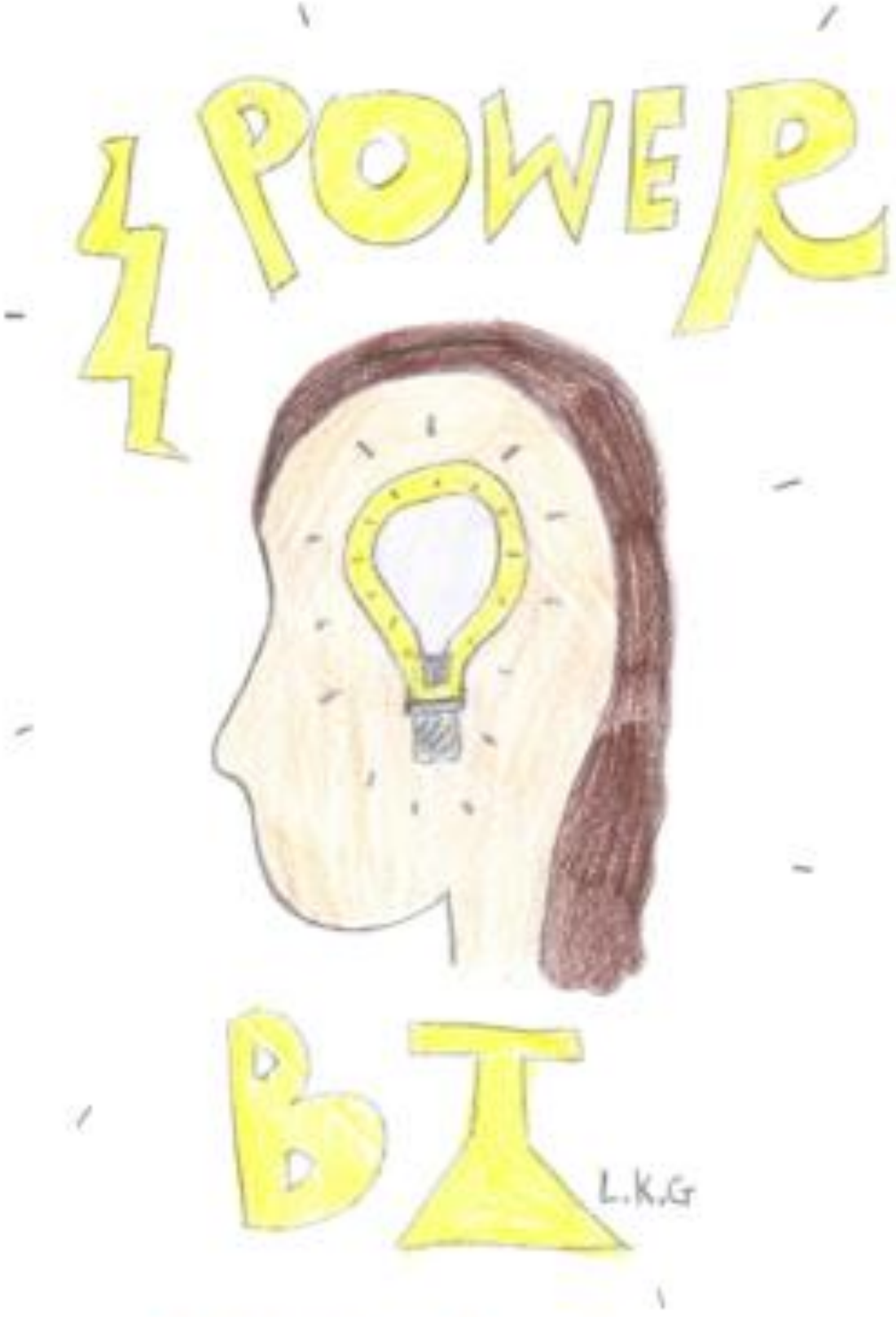
INTRODUCTION TO POWER BI

First edition



Guido L. Geerts

ABOUT THE BOOK

Edition: 1.1
Power BI Version: 2.65.5313.701 64-bit (December 2018)
Publication date: January 1, 2019

ABOUT THE AUTHOR

Guido Geerts is a professor and Ernst & Young faculty scholar at the Lerner College of Business, University of Delaware, where he teaches accounting information systems and big data technologies. Guido has received numerous awards for excellence in teaching, research, and service, including the University of Delaware Excellence in Teaching Award (2015), the Lerner College Outstanding Scholar Award (2014), the Lerner College Outstanding Teacher Award (2013), the American Accounting Association's Outstanding Service Award (2018), and the American Accounting Association's Strategic and Emerging Technologies Section Outstanding Researcher (2010) and Outstanding Service (2016) awards. He is the former chair of the Technology Task Force for the Pathways Commission Recommendation 4 (Curriculum and Pedagogy) and currently serves on the American Institute of Certified Public Accountants (AICPA) Pre-certification Education Executive Committee (PcEEC). Guido earned his PhD from the Free University of Brussels in 1993.

ACKNOWLEDGMENTS

A book like this is obviously not written in isolation. My sincerest thanks to the reviewers, all the practitioners who attended my Intro to Power BI workshops, Dana Rubenstein at the Delaware Society of CPAs for her motivation and for moving this project forward, and all the students at the University of Delaware who have taken my class on big data technologies. The book is dedicated to my daughter, Leanna, and my wife, Myunghee, for their unconditional support.

REVIEWERS

Murthy, Uday. University of South Florida

Ross, Barbara. Eastern Michigan University

Schwartz, *Nadia*. Augustana College

Tan, Kinsun. State University of New York at Albany

CONTENTS

7. CASE STUDY

7.1. THE DATA SET (DATA DISCOVERY)
7.2. DATA EXTRACTION AND ORGANIZATION (COLLECTION)
7.3. DATA ENRICHMENT
7.4. DATA ANALYSIS

GLOSSARY

REFERENCES

APPENDIX

PREFACE

As I discuss in my book *An Introduction to Big Data*,[1] big data technologies are transforming today's business landscape, and self-service business intelligence (SSBI) software plays an important role in this transformation. The two key premises of SSBI software are that:

- SSBI provides extended data processing capabilities for extracting, profiling, cleaning, integrating, and analyzing data.
- SSBI is easy to use. No degree in computer science is required!

One of the leading SSBI products is **Microsoft Power BI**, which is a suite of tools that supports most phases of a big data project: extracting data, profiling data, cleaning and transforming data, building rich information models, and building advanced dashboards for analytical purposes. Microsoft Power BI (the BI is short for "business intelligence") is often called "Excel on steroids": a reengineered version of Excel adapted to the specific needs of the new data environment.

This book targets both business students and professionals. There are no specific pre-requirements. Anyone with basic computer proficiency should be able to understand the materials presented and complete all the exercises and assignments. An important feature of the book is its breadth—you will learn basic skills to help you with the different parts of a big data project. Other books in this series will discuss specific tools and skill sets in more detail.

The main objective of this book is to get you started with Power BI. After completing it, you should be ready to start your own big data projects. A brief overview of the book's overall objectives and chapter-specific objectives is provided below.

Overall Objectives

1. Become familiar with Power BI.
2. Gain a better understanding of the power and potential of SSBI software such as Power BI.
3. Get ready to start your own big data projects.

[1] Guido L. Geerts, *An Introduction to Big Data* (Author, 2017). Available at www.bigdatavillage.com/books.

Chapter-Specific Objectives

Chapter 1. **Installing Power BI and Understanding the Data Sets**
This chapter provides information on how to install the software (Power BI Desktop) and the data sets you will use in this book.

Chapter 2. **The Data Process Chain**
Big data projects are multi-phase processes that follow a prototypical pattern, known as the data process chain. Before you develop a dashboard for analytics purposes, you first need to discover, extract, profile/transform, and enrich data. In this chapter, you will learn about the importance of each of the separate phases and how they relate to one another.

Chapter 3. **The Power BI Desktop Workspace**
This chapter will introduce you to the Power BI Desktop workspace—the main menu, ribbons, panels, and the canvas—and how they all work together. More specifically, we will discuss how the different phases of the data process chain are supported by the Power BI Desktop workspace.

Chapter 4. Dashboards and Visualizations
The most exciting part of any SSBI tool is building powerful interactive dashboards and then playing with those dashboards to generate new insights. A dashboard consists of a number of visualizations that work together.[2] This chapter provides an introduction to dashboards and visualizations. Among other things, this chapter will:

- ➔ demonstrate how Power BI enables you to build powerful interactive dashboards with little effort;
- ➔ provide an overview of the different visuals and their characteristics;
- ➔ employ hands-on exercises to teach you how to use five of the visualizations in much more detail: cards, tables, stacked bar charts, slicers, and map charts. More specifically, for each of these visualizations, you will learn: (1) how to create the visualization, (2) how to enter data into the visualization, and (3) how to format the visualization.

Chapter 5. Information Modeling
The strength of your analysis will largely depend on the information you have available. An important part of big data projects is the enrichment of your raw data, which is primarily done by defining "measures." This chapter provides an introduction to information modeling with Power BI. Among other things:

- ➔ You will learn the basics of the DAX (Data Analysis Expressions) language, which is an extension to Excel functions that enables you to specify enrichments.
- ➔ You will learn the difference between columns and measures.

[2] The terms "visualization" and "visual" are used interchangeably in this book.

➔ Using hands-on exercises, you will learn how to develop information models.

Chapter 6. Data Collection
Most data are dirty and do not come in a format that is easy to analyze, so an important part of big data projects is to organize data for analysis purposes. This chapter provides an introduction to data collection. Using hands-on exercises, you will learn:

➔ how to use connectors to extract data
➔ how to profile data
➔ how to clean data
➔ how to transform and integrate data

Chapter 7. Case Study
In this chapter, you will first define an "active" link to a publicly available data set: licensing data as provided by Delaware's Open Data Portal. You will then take the data set through the different phases of the data value chain: organization, enrichment, and analysis.

To Be Noted
The outline above shows that we will traverse the data process chain in reverse order. Building dashboards (chapter 4) is fun and shows you the added value of an SSBI tool such as Power BI immediately. The discussion should also make you curious about how to shape data in order to build powerful dashboards like this. You will learn how to shape data in chapters 5 (Information Modeling) and 6 (Data Collection). Further, because testing formulas (i.e., information modeling) is done in dashboards, you should already know how to create dashboards by the time you start learning DAX.

Throughout the remainder of this book, we will use the following notation:

 **Assignment**

Learning Objectives

Useful Tips

Further Reading

Terminology

Side Note

CHAPTER 1
INSTALLING POWER BI AND UNDERSTANDING THE DATA SETS

Learning Objectives

- ➔ Understand the different implementations of Power BI and how the software relates to Excel
- ➔ Install Power BI Desktop
- ➔ Understand the data sets we will use in this book.

1.1 SOFTWARE

The software we will be using in this book is Microsoft Power BI **Desktop**. Below, I will discuss the different implementations of Power BI and how to download Power BI Desktop. If you are already familiar with this information, then you can skip this section.

Further Reading: official Power BI website

Microsoft. *Power BI Documentation*, 2018. Available at https://docs.microsoft.com/en-us/power-bi/.

1.1.1 The Different Implementations of Power BI

Microsoft provides a set of tools that supports a wide variety of data processing tasks, such as data extraction, data organization—which includes cleaning and integration—information modeling, and data visualization. These tools are available in different shapes and forms (i.e., different implementations). Figure 1.1–1 below provides a summary.

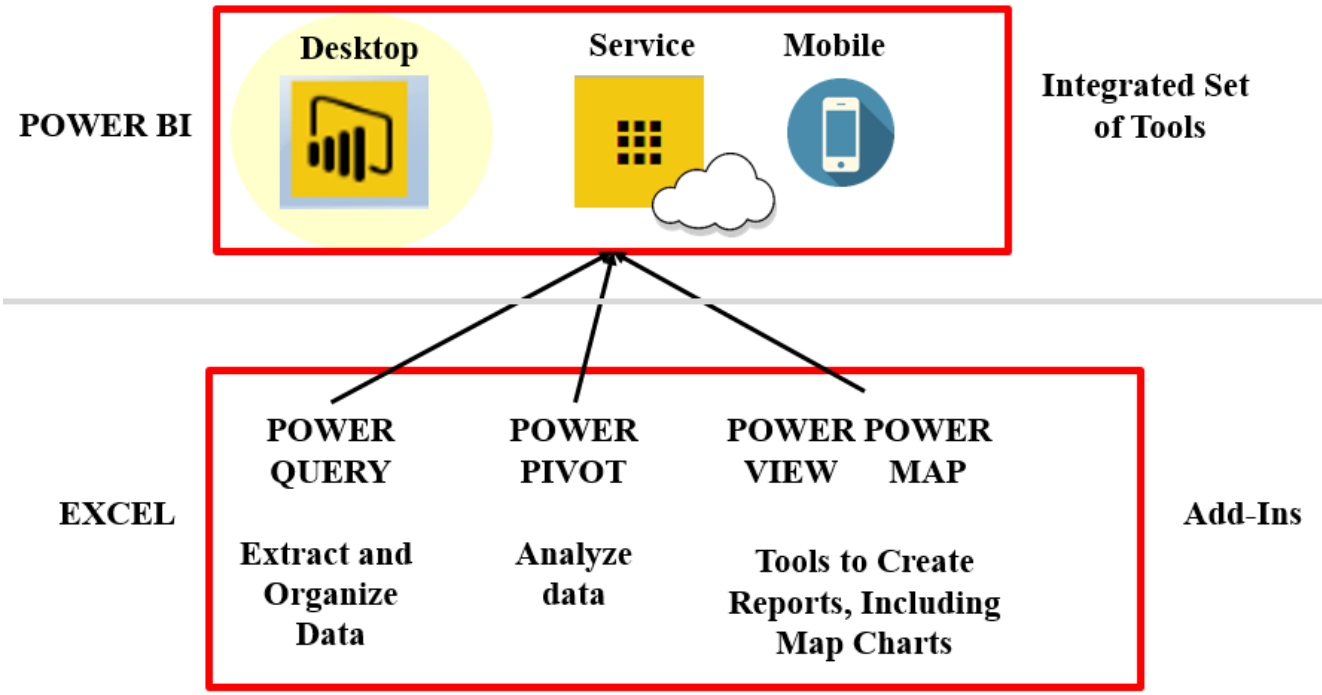


Figure 1.1–1
Power BI's Different Implementations

Figure 1.1–1 shows that there are two different *types* of implementations:

1) as **Excel Add-Ins** (Excel 2013 and Excel 2016);
2) **Power BI**, which integrates all tools into one system.

Excel Add-Ins

Four different tools are available that can be integrated as part of Excel as add-ins (table 1.1–1).

Table 1.1–1
Excel Add-Ins

TOOL	USE
POWER QUERY	A tool for extracting and organizing data
POWER PIVOT	A tool to for analyzing data
POWER VIEW	A tool for creating reports, including map charts
POWER MAP	A tool for creating reports, including map charts

Figure 1.1–2 below shows “Power query” as an Excel 2013 add-in.

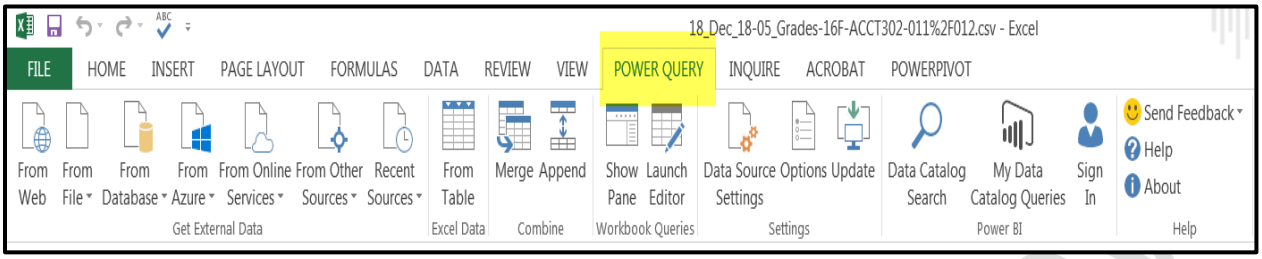


Figure 1.1–2 Power Query as an Excel 2013 Add-In

Power BI

Power BI integrates the different tools into one system. As shown in figure 1.1–1, there are three different implementations of Power BI: Desktop, Service, and Mobile. They have different interfaces as well as differences in functionality.

- **Power BI Desktop** is a stand-alone version that can be downloaded for free.
- **Power BI Service** is a cloud-based solution that makes the sharing of data and dashboards easy.
- **Power BI Mobile** includes apps that allow you to efficiently connect to your data and dashboards through your favorite mobile devices, such as an iPhone or iPad.

All explanations and exercises below refer to Power BI Desktop.

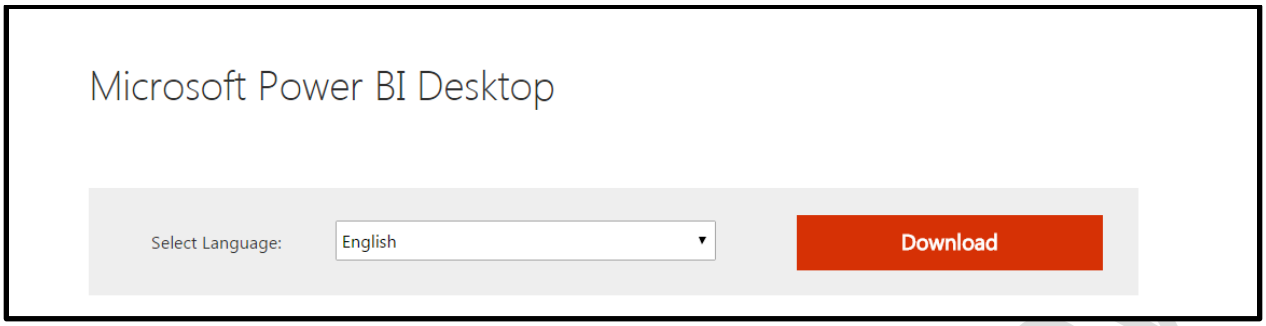
1.1.2 Installing Power BI Desktop

Where can I download Microsoft Power BI Desktop?

Google “Power BI Desktop download,” or go directly to the following website:

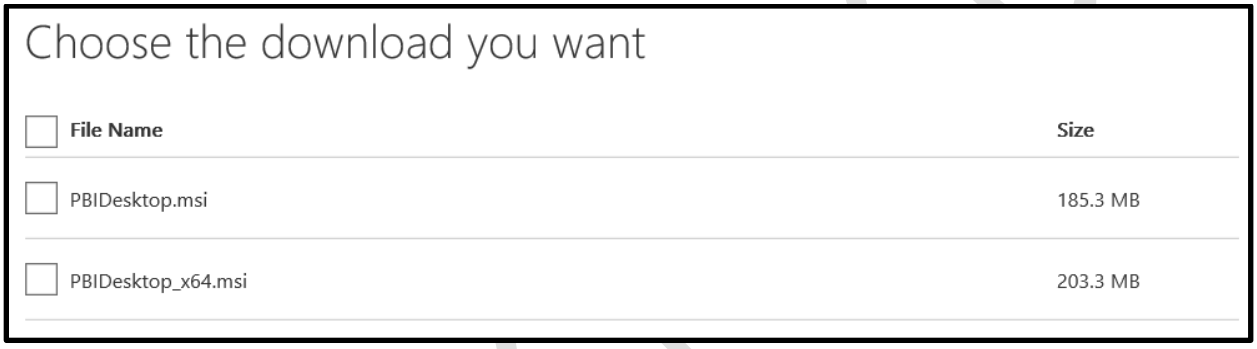
https://www.microsoft.com/en-us/download/details.aspx?id=45331

The website will show the following:



Click the Download button.

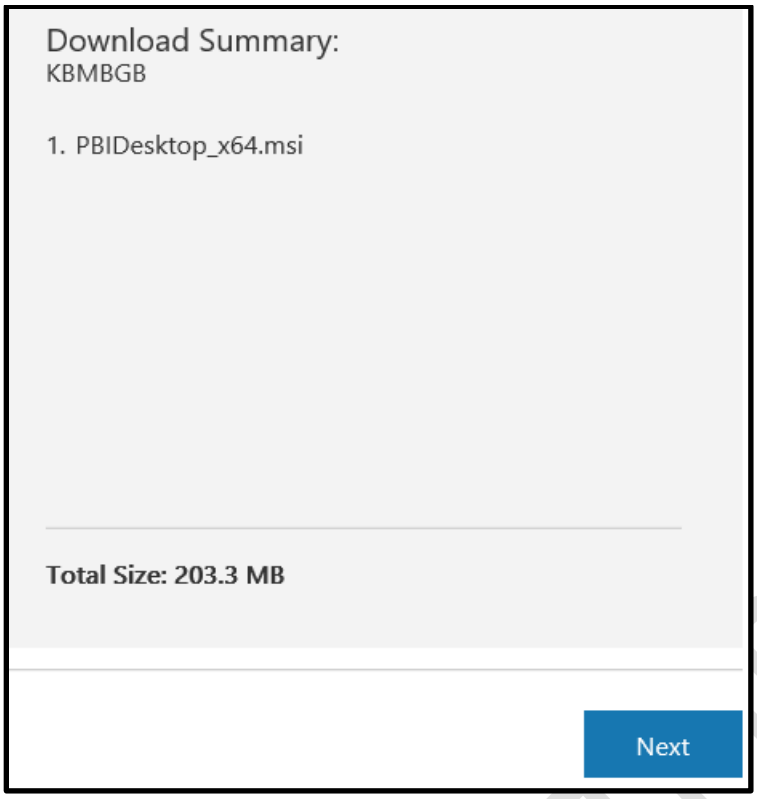
Next, you will be asked to make a choice:



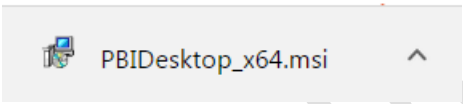
Should I download the 32-bit or 64-bit version of Windows?

The answer for most people is 64-bit, since that version can handle large amounts of memory more efficiently than 32-bit. If you don't know whether you have a 64-bit system, you can always look that information up. For example, on my system (I run Windows 7), if I click Start, Computer, and System properties, then I get the following information:

Next, you will be informed of how much space you need to install Power BI Desktop.



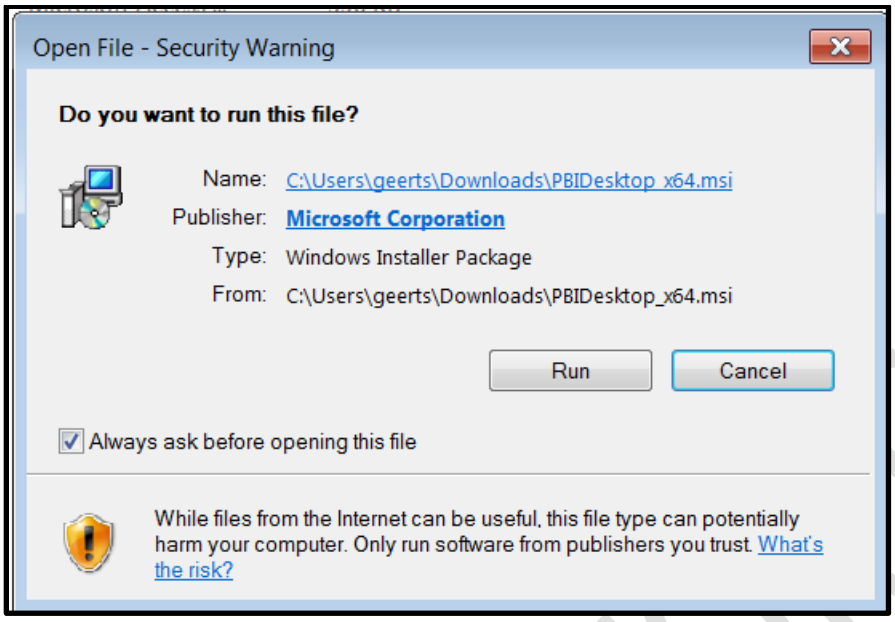
Click Next, and Power BI Desktop will be downloaded.



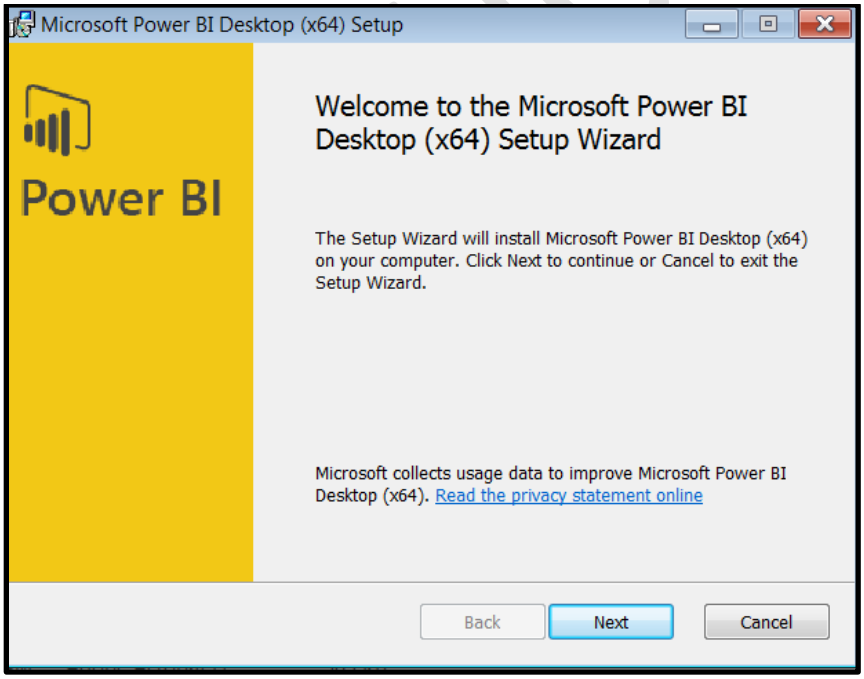
Open the file to start the installation process.

Installing Power BI Desktop

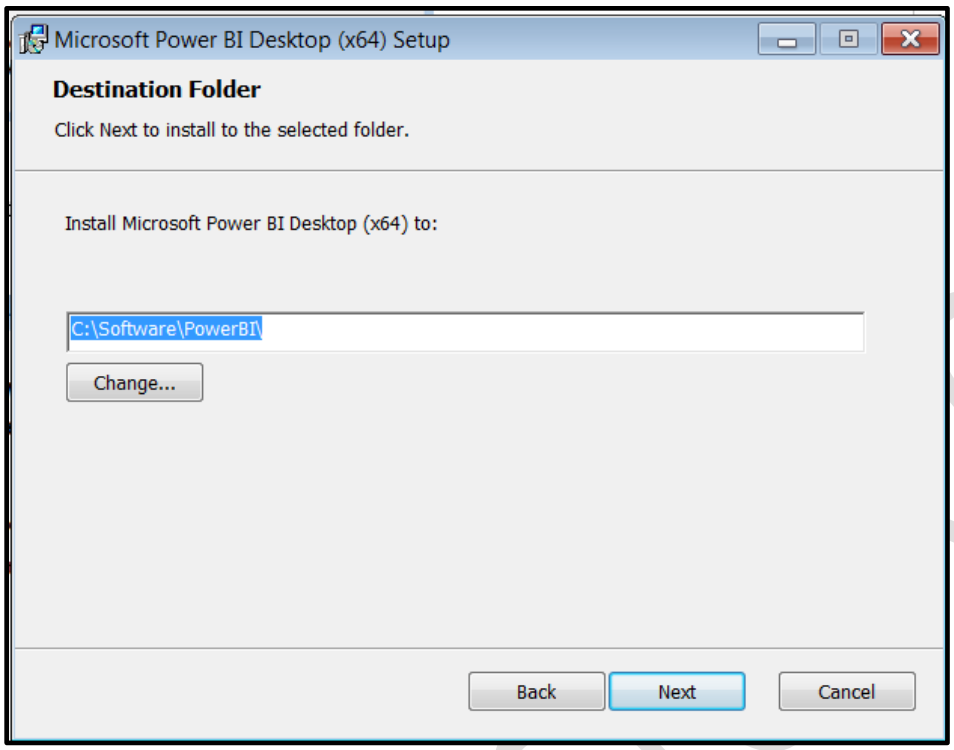
The installation process will start with the following window:



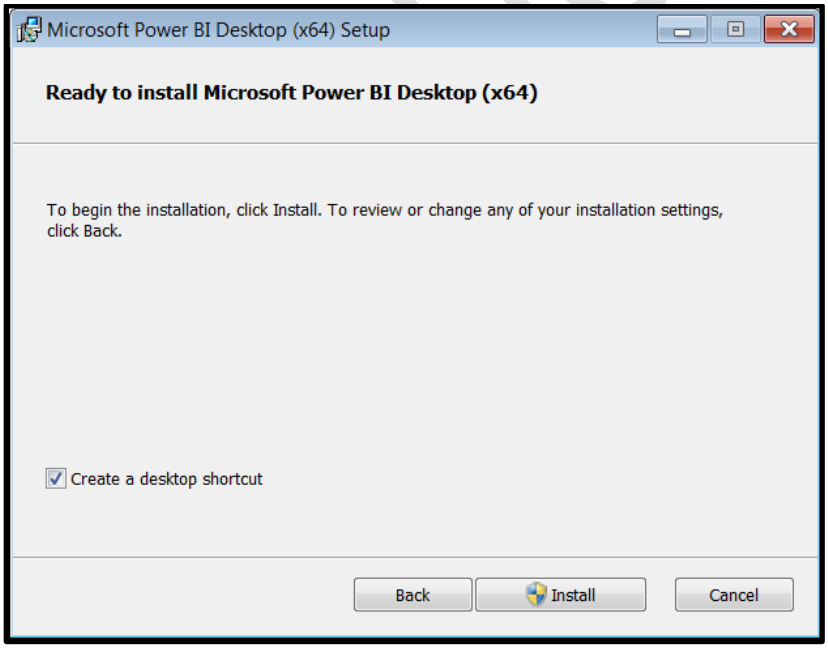
Click the Run button, and the Setup wizard will start.



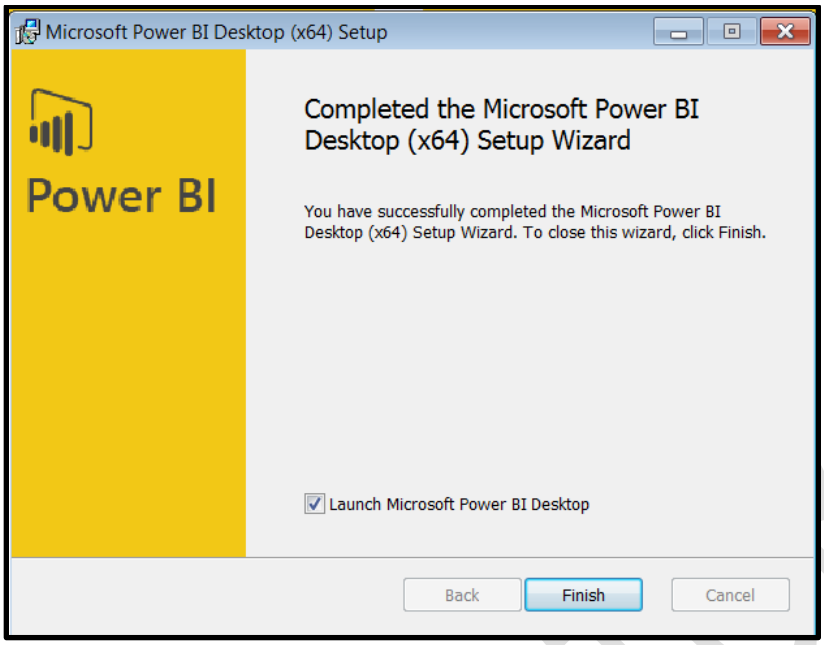
Click the Next button again. After accepting the license agreement and clicking Next, the installation wizard will ask you where the software should be installed.



Type in the directory where you want Power BI to be installed and then click on Next. The following screen will then pop up:



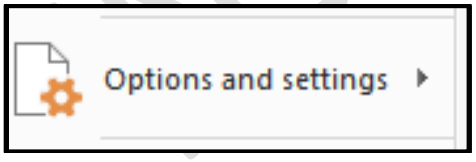
Click the Install button, and the installation process will get started. The wizard will let you know when the installation is completed.



Click the Finish button and you are **ready to go!** ☺

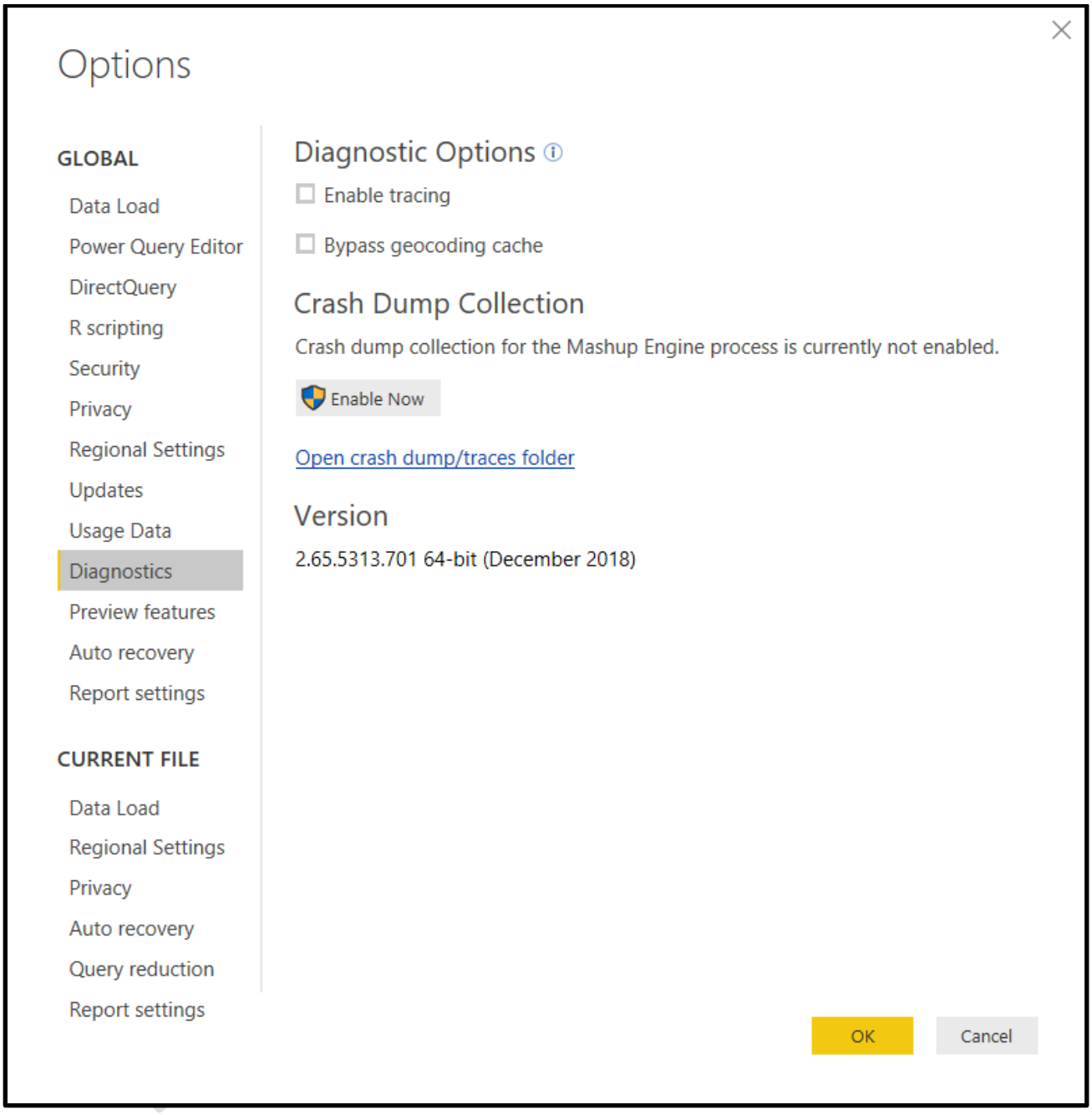
1.1.3 Continuous Software Updates

Microsoft continuously improves Power BI and makes updates available on a monthly basis. To determine what version you are currently using, open Power BI, click File, and then choose "Options and settings."



In the Options menu, click on Diagnostics. As the figure below shows, the version used in this book is 2.65.5313.701 64-bit (December 2018). My goal is to update this book and to integrate new innovations and other changes twice a year: on June 15 and on December 15. For businesses, it is important to

define a “version” policy: when to update, and making sure that everyone is using the same version. For teachers, I recommend using the same version throughout the semester.



1.2 DATA SETS

For the hands-on exercises and assignments found in chapters 4–6, we will use data from the KaDo case which I originally developed with my colleague Kinsun Tam, an accounting professor at SUNY Albany.[3] The case represents the common situation of a retailer: a company that buys and sells items. In the case of KaDo, the company buys and sells kachina dolls—pieces of art that are typically carved from cottonwood root and that convey a certain message.[4] Figure 1.2–1 shows a bear kachina doll.

Figure 1.2–1
Illustration of a Bear Kachina Doll

[3] Guido L. Geerts and Kinsun Tam, "KaDo: An Advanced Enterprise Modeling, Database Design, Database Implementation, and Information Retrieval Case for the Accounting Information Systems Class." *Journal of Information Systems*, vol. 22, no. 2 (Fall 2008): 141–50.

[4] We could have chosen any other item (cars or computers, for example), and the data structures and analysis would have been similar.

The three data sets we will use in chapters 4–6 come from the KaDo case and deal with vendor management, order management, and sales management, respectively. A fourth data set, to be used for the case study, is publicly available online. The four data sets are briefly discussed below.

DATA SET #1 (KADO)

BUSINESS PROBLEM	**Vendor Management (Acquisition)** Finding artists who can carve kachina dolls: who (vendors) can deliver what items (dolls)
FILE	ABILITY.PBIX[5]
LEARNING OBJECTIVES	How to create and use dashboards
DATA	Data on the types of products (types of kachina dolls) a vendor (artist) is able to deliver (carve)
SOLUTIONS	ABILITYSOLUTION.PBIX[6]

DATA SET #2 (KADO)

BUSINESS PROBLEM	**Order Management (Acquisition)** What have we ordered, and from whom? How many items do we have available?
FILE	ORDERS.PBIX
LEARNING OBJECTIVES	How to create and use measures and attributes
DATA	Data on what was ordered (product type) and from whom (vendors/artists)
SOLUTIONS	ORDERSSOLUTION.PBIX

[5] The PBIX extension refers to a Power BI file.
[6] Complete solutions are provided for all exercises and assignments.

DATA SET #3 (KADO)

BUSINESS PROBLEM	**Sales Management (Revenue)** What items/products have we sold?
FILES	ITEMDATA.TXT CUSTOMERDATA.ACCDB SALESDATA.XLSX
LEARNING OBJECTIVES	How to extract, profile, clean, restructure, and integrate data from multiple sources
DATA	Data on the sale of individual items to customers
SOLUTIONS	SALESSOLUTIONS.PBIX

DATA SET #4

An active link to a data set that is stored online.[7]

FILES	Publicly available data set: https://data.delaware.gov/
LEARNING OBJECTIVES	Active link via OData: https://data.delaware.gov/OData.svc/pjnv-eaih
DATA	Public records on professional and occupational licenses in Delaware
SOLUTION	LICENSEINFOSOLUTION.PBIX

[7] I also make a static file available (LICENSEINFOSOLUTION.PBIX) for backup purposes. You can find it in your DropBox folder.

CHAPTER 2
THE DATA PROCESS CHAIN

Learning Objective

Become familiar with the different steps in a big data project.

Interactive dashboards are the ultimate step of a big data project, but getting to that step also requires discovering data, extracting data, profiling data, transforming data, and enhancing data. Each of the different steps in the process requires different skill sets and tools. Figure 2.1 portrays the data process chain. Each step is briefly discussed below.[8] Power BI provides support for the part of the chain indicated by the area shaded in green.

DATA DISCOVERY
This step involves identifying the data you have access to and that are relevant for the decisions you must make. This process requires that you understand the data landscape: spreadsheets, social media, email, enterprise systems, RFID (radio frequency identification) tags, audio, and video, for example. You must also judge the feasibility of gaining access to these kinds of data. Thinking outside of the box is an important part of data discovery.

DATA SOURCES
This is the output of the data discovery process. These are the raw data that you would like to explore further and possibly consider for decision-making purposes.

DATA COLLECTION
This step involves profiling and transforming data into an organized data set that will be useful for decision-making purposes. "Transformation" refers to activities such as cleaning, joining, merging, and integration, among others. The transformation process is generally known as ETL, or extract-transform-load. Power BI provides advanced functionality for data collection.

ORGANIZED DATA
All required data are now cleaned and integrated.

[8] See Geerts (2017) for a more detailed discussion of the data process chain.

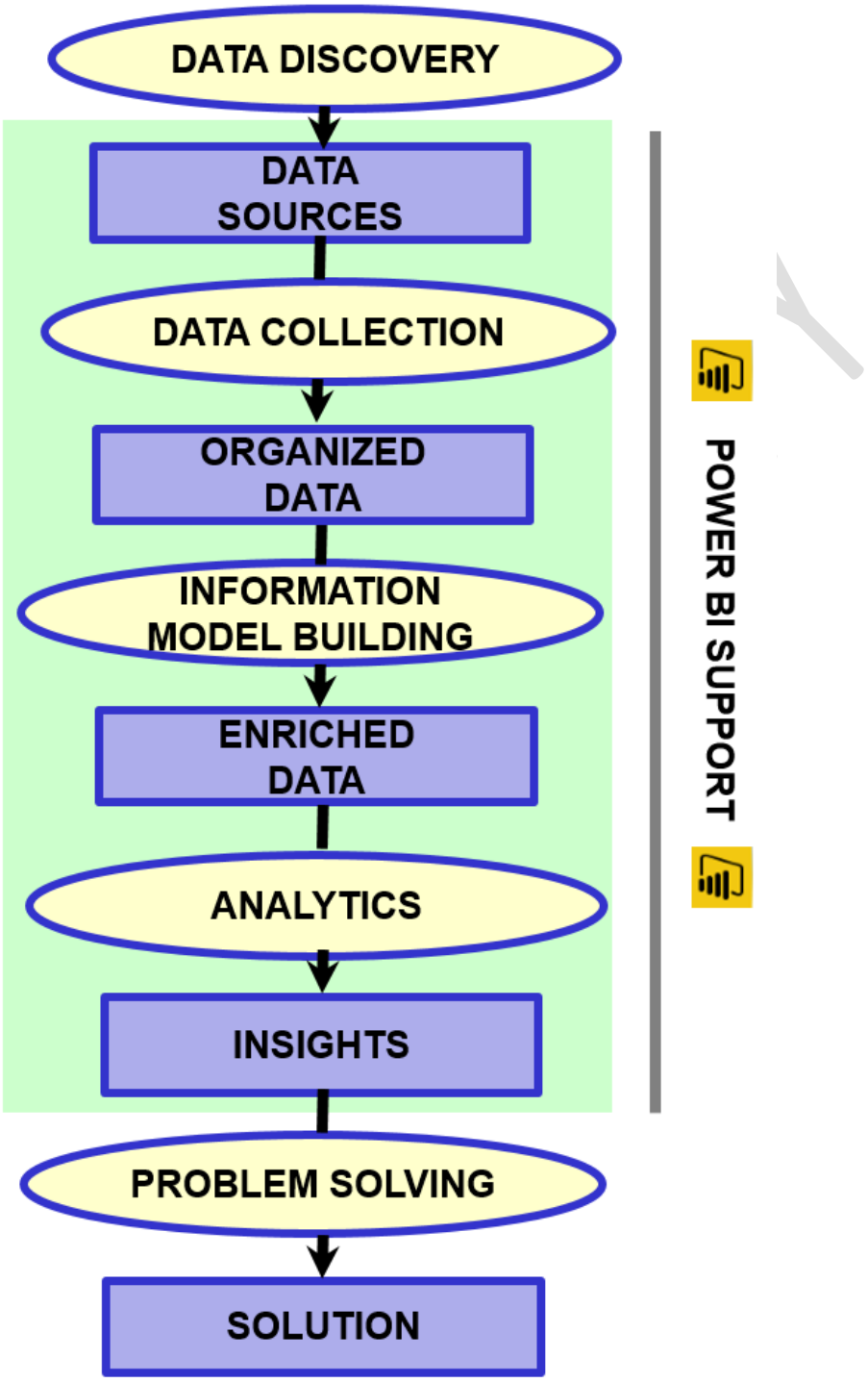


Figure 2.1
The Data Process Chain. Source: Geerts (2017)

INFORMATION MODEL BUILDING
This step involves preparing the information necessary for analysis; this primarily comes down to defining aggregated data by means of formulas, which can then be sliced and diced in many different ways.

ENRICHED DATA
The data set ready for analysis; this is the analytical database.

ANALYTICS
This step involves analyzing the data to generate insights in support of problem solving and decision making. Building powerful interactive dashboards that help identify patterns between data—relationships and trends—is an important part of this step.

INSIGHTS
What you have learned from the data.

PROBLEM SOLVING
In this step, you use the insights that were generated during the analytics phase for problem solving and decision making.

SOLUTIONS
The decisions you have made.

The following are a few general observations about the data process chain and the tools available to support that chain.

- The strength of the tools across the chain might differ.
- Some tools focus on one specific step only.
- Depending on the project, some steps might not be necessary. For example, the data collection step might not be needed if the data you receive are structured, as might be the case for a database or an enterprise resource planning (ERP) system.
- Although not explicitly indicated in figure 2.1, the data process chain is iterative in nature. For example, insights might trigger the need for additional data that need to be identified (i.e., the discovery step), organized, and other actions.

CHAPTER 3
THE POWER BI DESKTOP WORKSPACE

Learning Objectives

- ➔ Become familiar with the Power BI Desktop workspace and learn where you can find things.
- ➔ Link the Power BI Desktop workspace to the data process chain.

Side Note

For the illustrations in this section, I have used the ABILITYSOLUTIONS.pbix file. Assuming you have already installed Power BI, click on this file to open it. You can then follow along with the examples below. Don't hesitate to explore!

The Power BI Desktop workspace is easy to learn, since its structure is similar to that of most Microsoft applications: a main menu with context-specific ribbons. Figure 3–1 below shows both the main menu and the ribbon associated with the Home tab in the main menu.

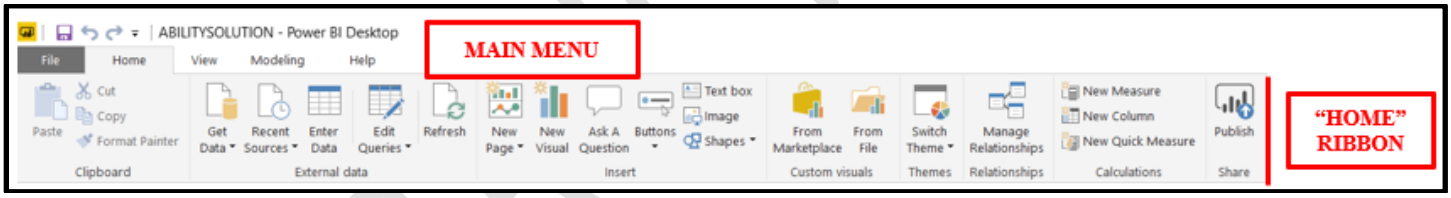
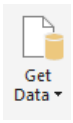


Figure 3–1
Power BI Desktop Workspace: Main Menu and Ribbon

The two objectives of this chapter are (1) for you to become familiar with the Power BI Desktop workspace and (2) to link the workspace to the data process chain. We will limit ourselves to a general overview in this chapter; more specific details about the different parts of the workspace will be discussed in chapters 4–6.

Power BI Desktop integrates five key applications, which are summarized in table 3–1 below.

Table 3–1
Power BI's Desktop Five Key Applications

	APPLICATION	DESCRIPTION
1	 **GET DATA**	Extraction of data through easy-to-use data connectors
2	 **EDIT QUERIES**	Data cleaning and restructuring
3	 **RELATIONSHIPS VIEW**	Data integration
4	 **DATA VIEW**	Look at data and define information models
5	 **REPORT VIEW**	Define powerful interactive dashboards

The five links shown in figure 3–2 illustrate how the different applications support three different steps in the data process chain: data collection, information model building, and analytics. Next, we will elaborate on each of the links.

DATA COLLECTION: GET DATA (#1)

When you click on "Get data" in the Home tab ribbon (see figure 3–3), a new dialog box will appear with a list of data connectors (see figure 3–4). A data connector is an interface that makes extracting data from a specific data source easy. Power BI provides an extensive (and continuously growing) list of data connectors. Data connectors will be discussed in depth in section 6–3.

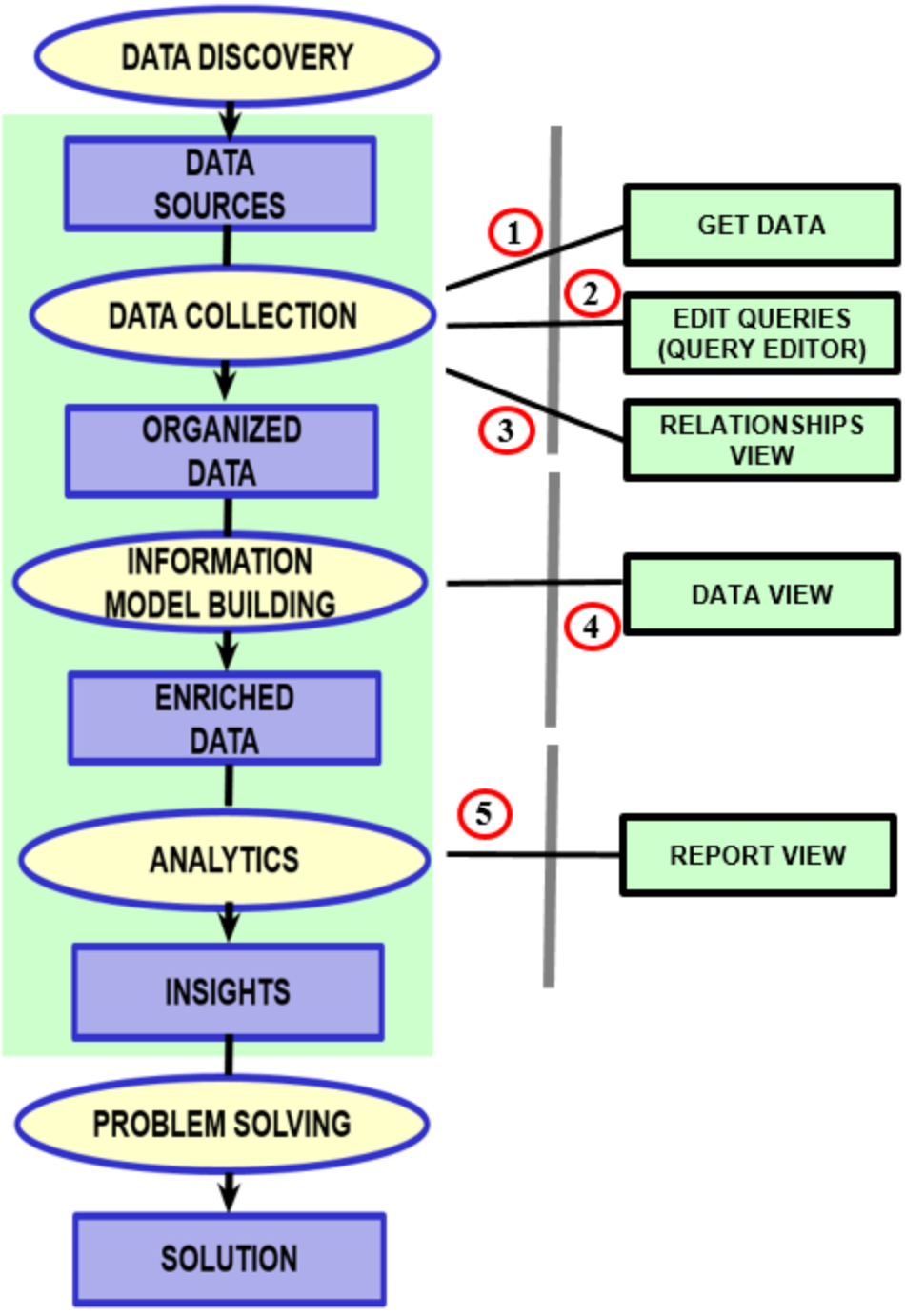


Figure 3–2
Linking the Power BI Desktop Workspace with the Data Process Chain

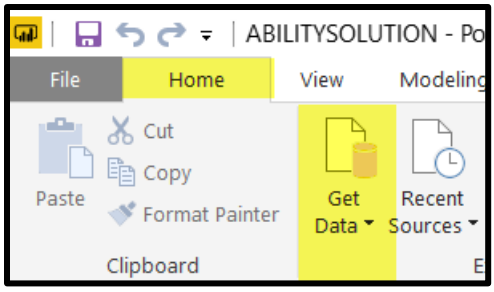


Figure 3–3
The "Get Data" Ribbon in the "Home" Tab (Main Menu)

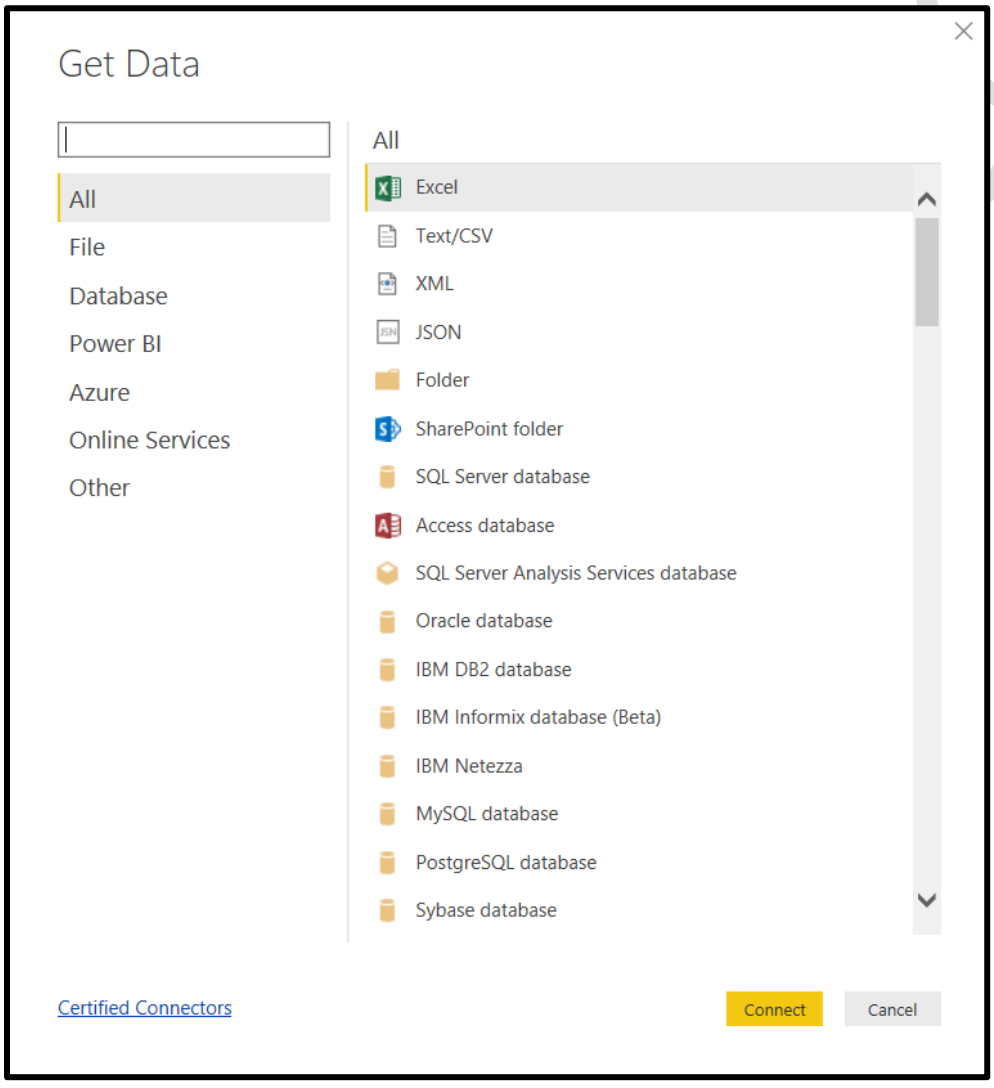


Figure 3–4
The "Get Data" Dialog Box Showing a List of Data Connectors

DATA COLLECTION: EDIT QUERIES (QUERY EDITOR) (#2)

When you click on "Edit queries" in the Home tab (see figure 3–5), a new dialog box—the query editor—appears (see figure 3.6). The query editor provides an extensive list of tools to clean, structure, and integrate data. Figure 3–6 identifies the five main parts of the query editor, each of which will be briefly discussed below. A more detailed explanation of the query editor will be provided in section 6.4.

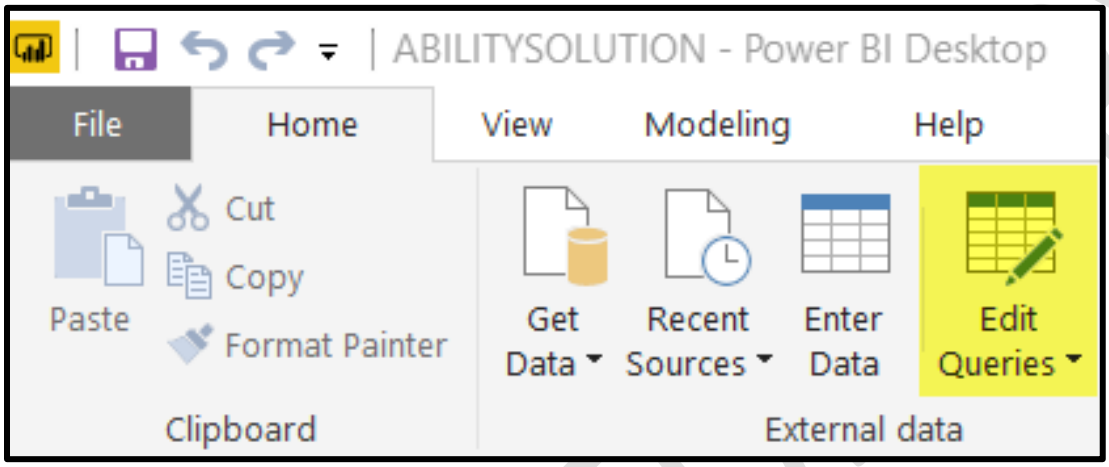


Figure 3–5
The "Edit Queries" Ribbon in the "Home" Tab (Main Menu)

Side Note

You will likely see a lot of yellow, which indicates errors. Don't be alarmed. The reason for the error messages is that the data sources that are loaded into the data warehouse are not on your computer. In chapter 6, we will learn how to fix this problem.

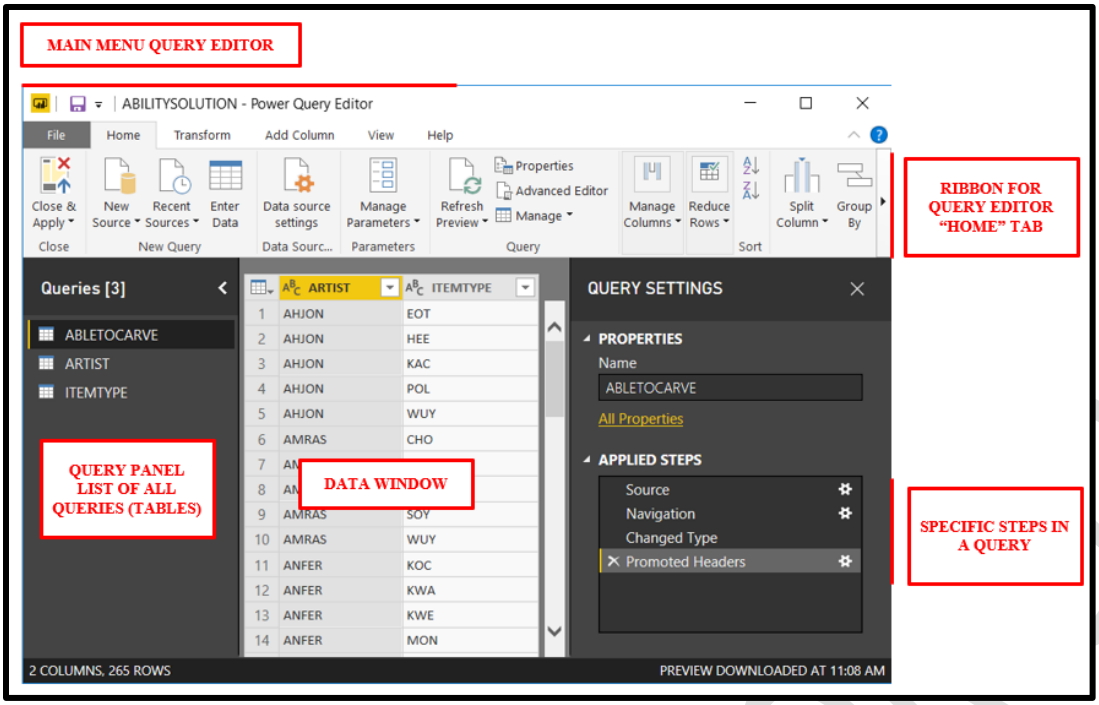


Figure 3–6
The Query Editor

MAIN MENU
The query editor has its own main menu with six buttons: File, Home, Transform, Add column, View, and Help.

RIBBON
Each of the six tabs on the main menu has its own ribbon packed with tools. Click on any of the six tabs and explore the different options that power query has to offer, such as removing rows and columns, defining data types, merging queries, and appending queries.

LIST OF ALL QUERIES (QUERY PANEL)
Each query represents a table. A query defines how the extracted data are cleaned and integrated.

SPECIFIC STEPS IN A QUERY
Every query consists of a number of steps; each step defines a specific cleaning or integration task. Steps can be added, deleted, and rearranged with ease.

DATA WINDOW
This window shows the data resulting from the execution of a transformation step. Only a sample set is shown.

DATA COLLECTION: RELATIONSHIPS VIEW (#3)

Once the tables are loaded, they can be related to one another and thus integrated. This is done by means of a data model definition in the Relationships view. Figure 3–7 shows where you can find the Relationships view and the view's different elements.

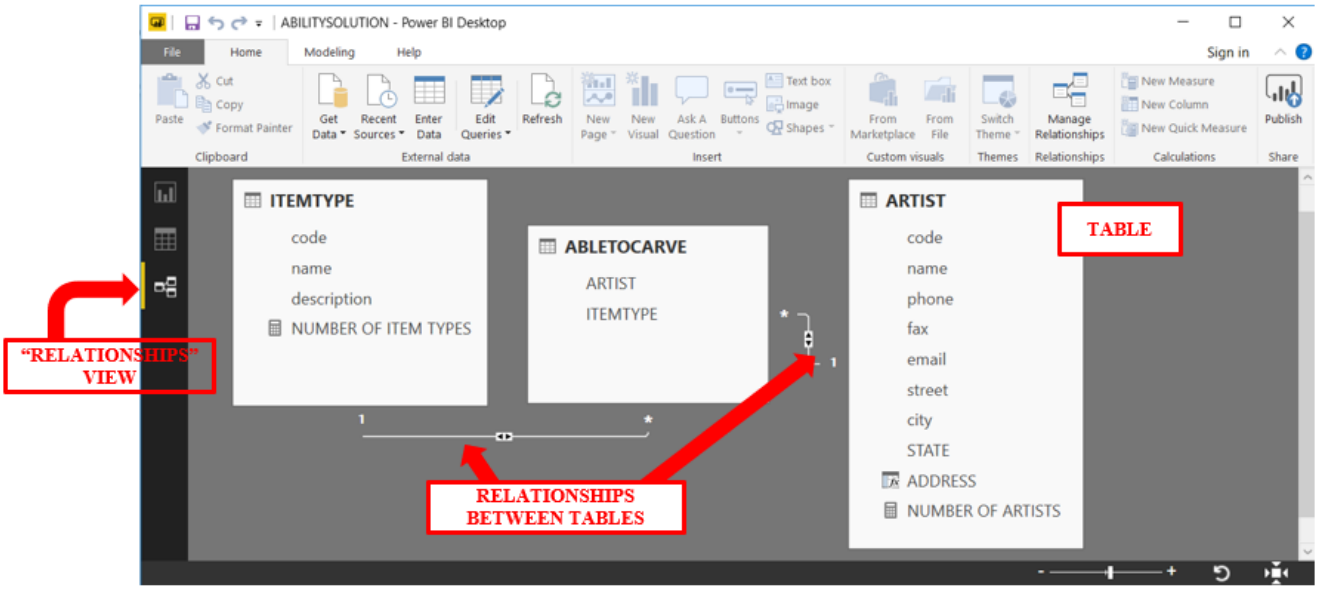


Figure 3–7
The "Relationships" View

A table is represented as a rectangle, and all its fields are shown. Figure 3–7 includes three tables: ITEMTYPE, ABLETOCARVE, and ARTIST. Tables can come from different data sources, which can then be integrated in the data model. Integration is done by defining relationships between shared fields. It is important that the connected fields have the same data type. If you click on "Manage relationships" in the upper-right corner of your screen (shaded in pink in figure 3–7), a dialog box will open that enables a more detailed specification of the relationships. The dialog box is shown in figure 3–8. The two relationships shown in figure 3–8 are the same as those shown in figure 3–7. Select the first relationship and click Edit. The dialog box in figure 3–9 will appear. This screen allows you to make more detailed specifications, such as the definition of the fields between which the relationship exists, the relationship's "cardinalities," and how to navigate the relationship. These are all advanced topics, but we want to show how Power BI enables you to define sophisticated data models.

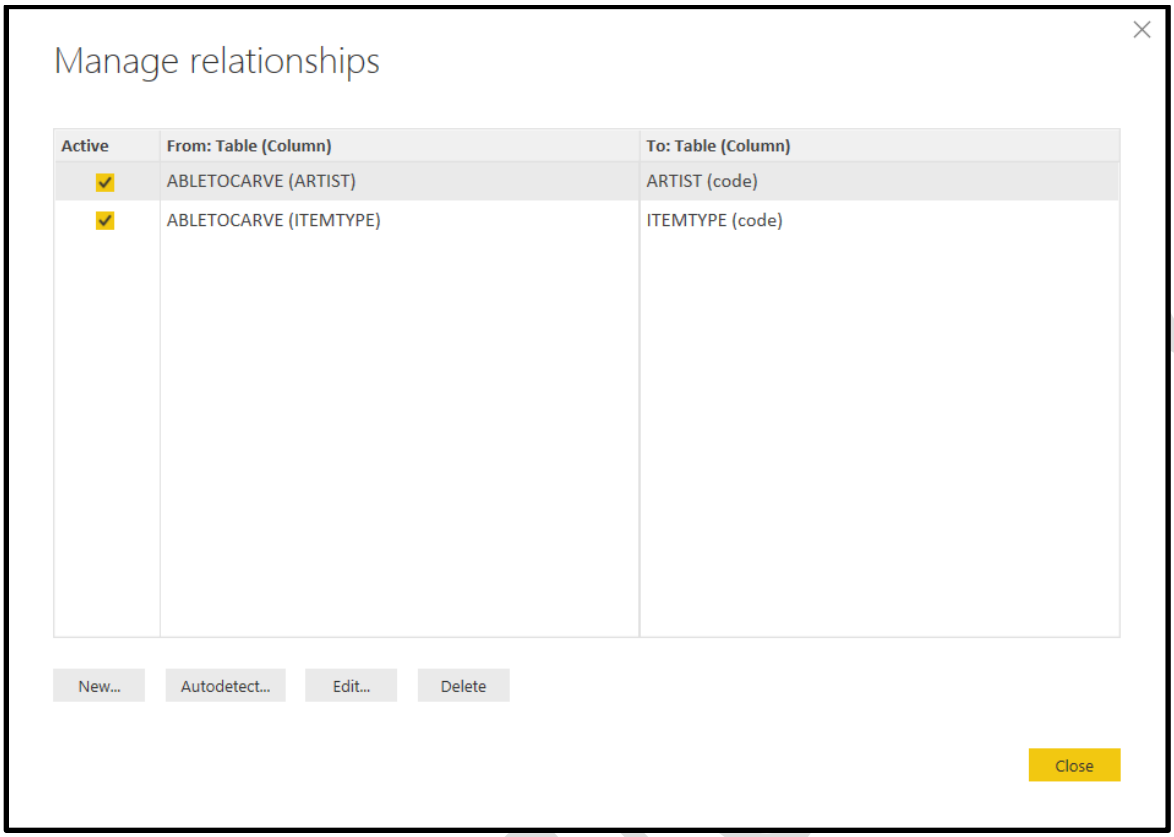


Figure 3–8
The "Manage Relationships" Dialog Box

INFORMATION MODEL BUILDING: DATA VIEW (#4)

Power BI's Data view enables you to look at the data after they have been loaded. The Data view is shown in figure 3–10. The middle part of the screen shows all data in the "active" table, i.e., the table selected in the Fields panel (right side). The Fields panel contains all tables in the data set. If you click on the arrow preceding a table, all the table's fields are shown. This is the case for the Artist table shown in figure 3–10.

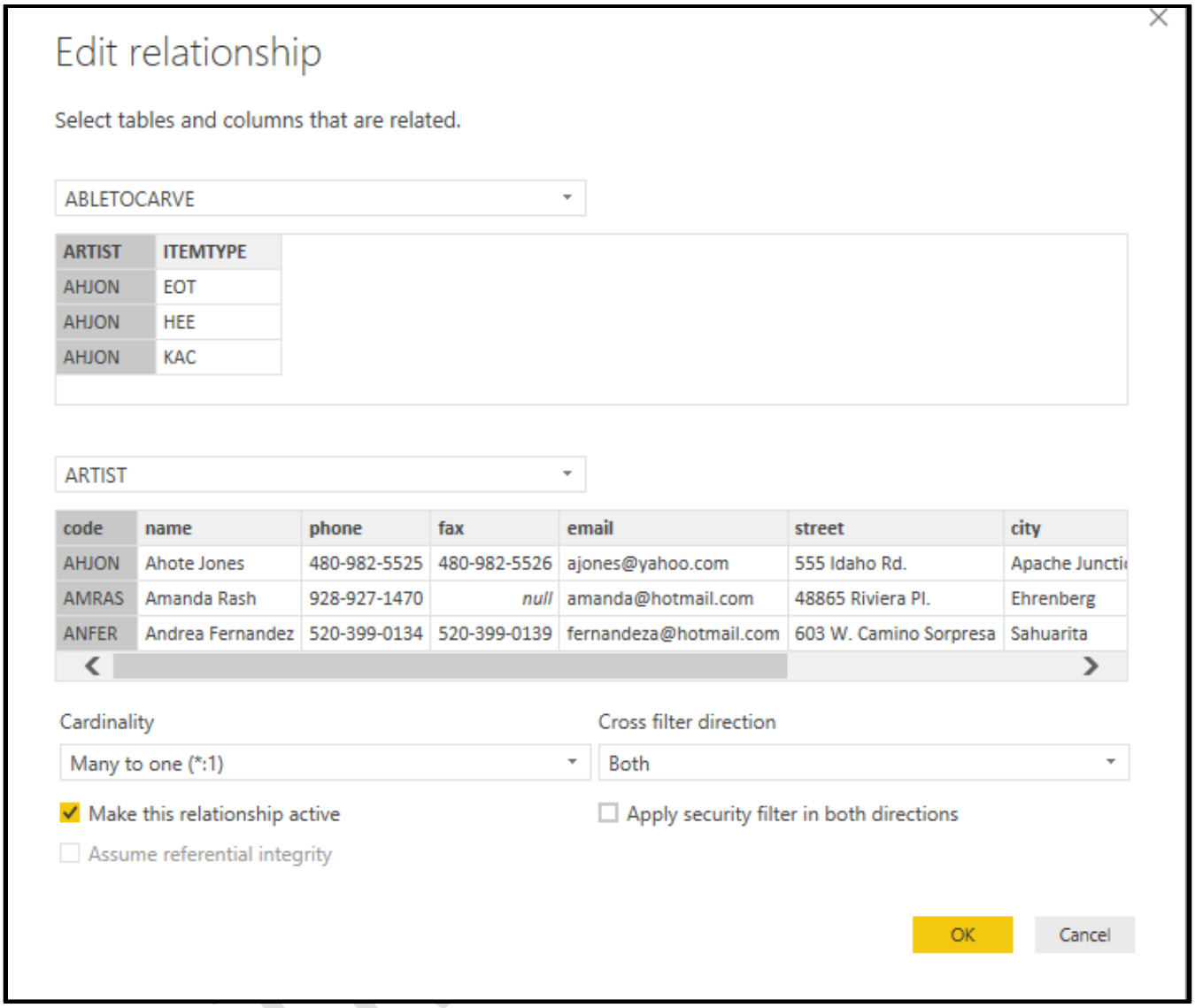


Figure 3–9
Editing a Relationship

As mentioned above, information model building—or information enrichment—is primarily realized by the definition of "columns" and "measures," which can be done in the Data view. Figure 3–11 shows where to define new columns and measures. You can find "New measure" in the Home ribbon (the Calculations group).

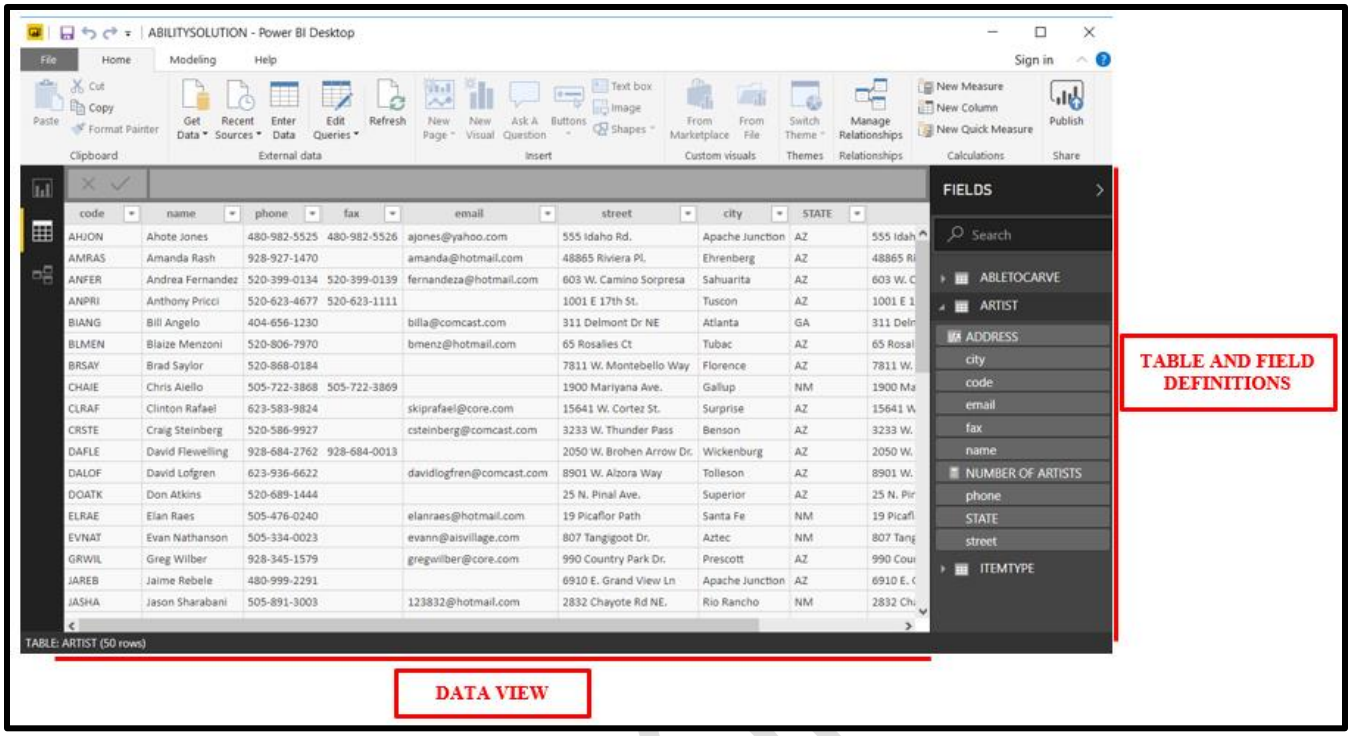


Figure 3–10
The "Data" View

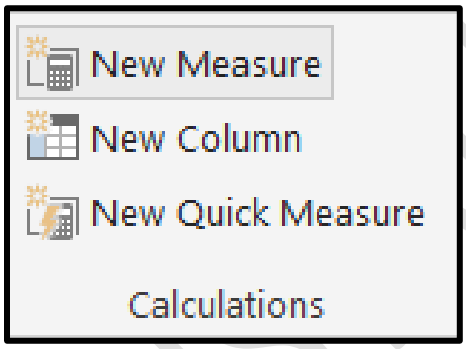


Figure 3–11
"New Measure" Definition

When you click "New measure" or "New column," an area will pop up in which you can define a formula using the **D**ata **A**nalysis e**X**pressions (or DAX) language (see figure 3–12). In chapter 5, you will learn how to define measures and columns using DAX in much more detail.

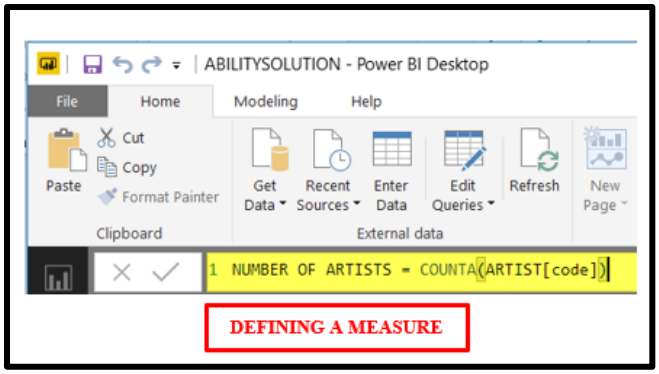


Figure 3–12
Defining a Measure

ANALYTICS: REPORT VIEW (#5)

Power BI's Report view enables you to define powerful interactive dashboards; the interface is shown in figure 3–13. The Fields panel is used to determine which data will be shown in the dashboard. A dashboard consists of visualizations; those currently available in Power BI are shown in the top part of the Visualizations panel—see different visualizations to choose from in figure 3–13. The Fields button (see "Definition of visualization" in figure 3–13) defines how the data will be used as part of the visualization. The Format tab (see "Formatting of visualization" in figure 3–13) defines how the data will be formatted as part of the visualization (e.g., the background color). Finally, the canvas is where you design the actual dashboard. We will discuss the Report view workspace in much more detail in the next chapter.

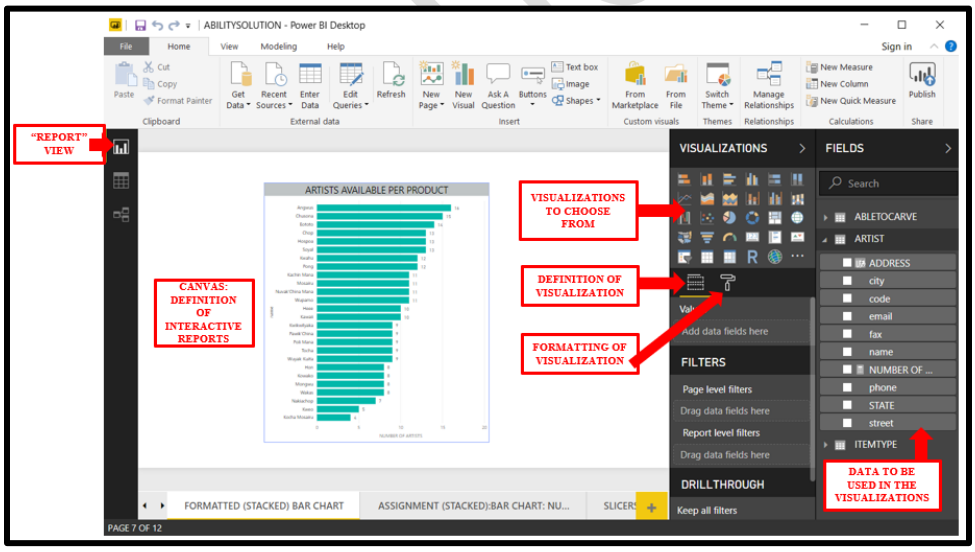


Figure 3–13
The "Report" View

Side Note

- To make things easier, Power BI often provides different "access paths" to the same tool. For example, the "Get data" dialog box (available on the Home tab) can also be accessed by clicking on "New source" in the query editor (see figure 3–14).

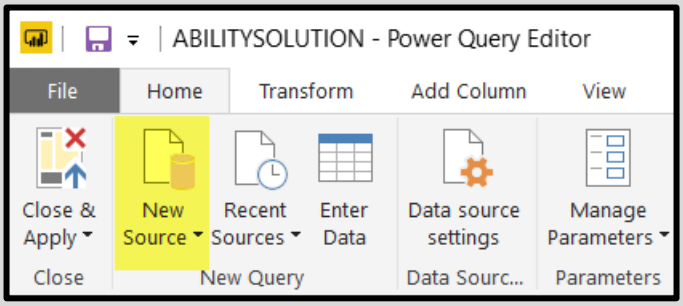


Figure 3–14
Accessing the "Get Data" Dialog Box from the Query Editor

- As pointed out above, Power BI is continuously updated, so the Power BI Desktop workspace on your machine might be slightly different from the version that is used in this book.

CHAPTER 4
DASHBOARDS AND VISUALIZATIONS

Learning Objectives

- ➔ Learn how to build powerful interactive dashboards with little effort.
- ➔ Learn about the many visualizations supported by Power BI, and their characteristics.
- ➔ In-depth, hands-on exercises for the following five visualizations: cards, tables, stacked bar charts, slicers, and maps. For each visualization, you will learn how to specify the visualization before adding data to and formatting the visualization.

Creating powerful interactive dashboards is the fun part of big data. ☺ Let's start by building a few amazing dashboards using a small data set from the KaDo case. We will do this in three steps. **First**, you need to understand the data set (4.1 and 4.2). **Next**, you need to learn about the tools you have available to define visualizations and dashboards—i.e., the Report workspace (4.3). **Finally**, you will build a series of dashboards using the following five visualizations: cards (4.4), tables (4.5), stacked bar charts (4.6), slicers (4.7), and map charts (4.8).

4.1 WHERE TO FIND THE DATA SET

We will use the **ABILITY.PBIX** data set for this exercise, which you can find in your DropBox folder. I have organized and enriched this data set for you. You will use the data set to build dashboards only.

4.2 UNDERSTANDING THE DATA

4.2.1 The Problem to Be Solved

A problem many companies face is knowing who (i.e., which vendors) can deliver the products they need. Products can be raw materials in the case of manufacturers, finished goods in the case of retailers, and so on. A couple of questions you might ask include, "From whom (which vendor) can I buy a computer (item)?" or "Who (which vendor/airline) can fly me from New York to Seattle (item/service)?"

Who can deliver the products we need?

Applied to KaDo, this question becomes:

Which artists can carve (and deliver) which kinds of kachina dolls (product)?

KaDo's success strongly depends on having access to this information. Let's look at the data the company has available to answer this question (i.e., the ABILITY.PBIX data set) in more detail.

4.2.2 Exploring the Data Set and Its Structure

Power BI makes it easy to look at data and data structures, as is illustrated in figure 4.2–1 below. Select Data view in the left panel. Then click on any of the tables in the Fields panel. As shown, if you click on the "ItemType" table, its contents will be shown in the Data panel in the middle section. Also, if you click on the arrow preceding a table name in the Fields panel, the table's specific fields will be shown.

Side Note

In the Fields panel, the ItemType table shows a field that is not part of the Data panel: "Number of item types." This is a measure—a formula used to calculate the number of different item types KaDo offers. Measures can be used in dashboards only and are not part of the table structure. We will explain measures in more detail in chapter 5.

Power BI also visualizes the data structure by means of a data model. Figure 4.2–2 below shows the data model for Ability.pbix. Click on Relationships view to see the data model. Every table is represented by means of a rectangle that contains the table's name and its fields. The lines between the tables further define how they are linked together.

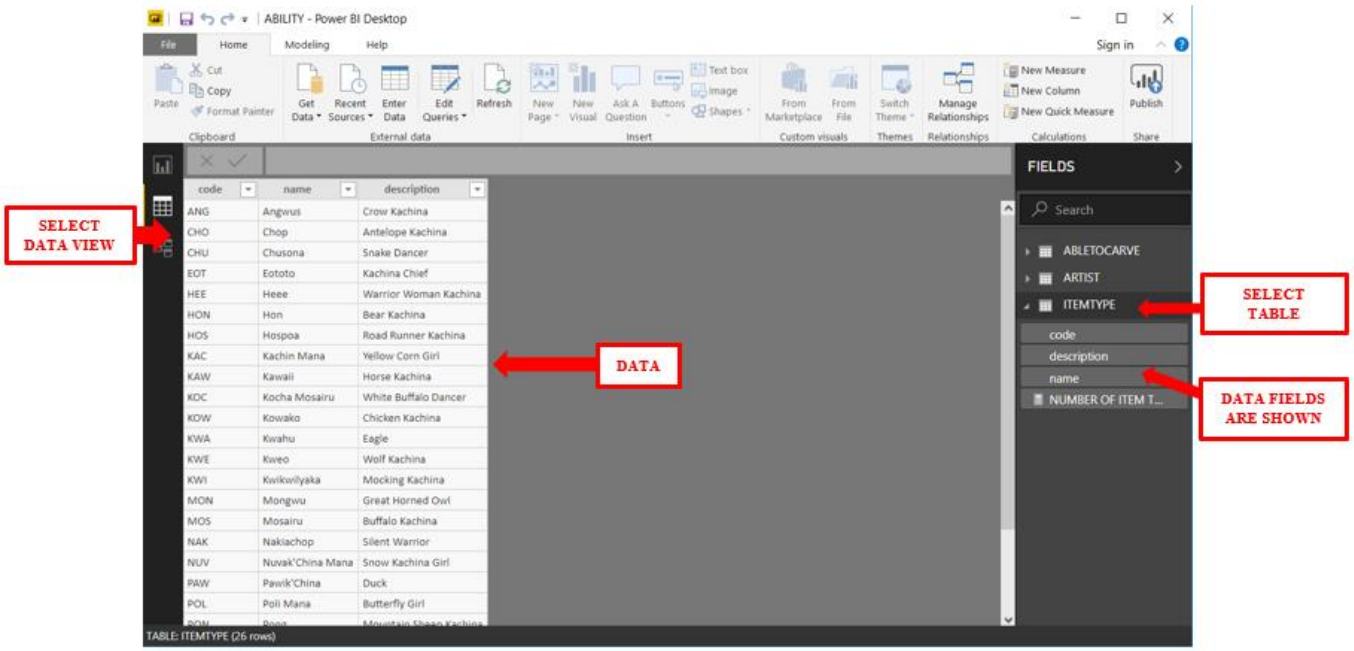


Figure 4.2–1
Looking at Data and Data Structures in Power BI

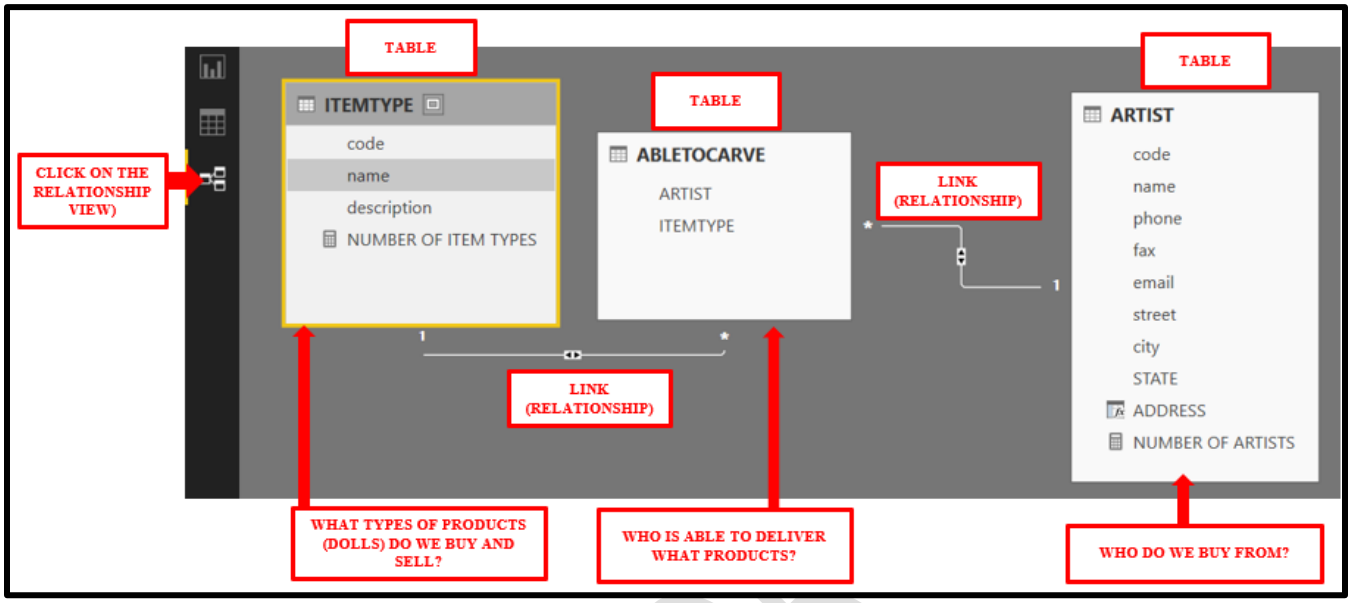


Figure 4.2–2
Data Models in Power BI

4.2.3 The Data Set

Next, let's look at each of the three tables in more detail.

ItemType **The types of products (dolls) that KaDo buys and sells**

Figure 4.2–3 shows the structure of the ItemType table. The ItemType table describes the types of products (dolls) KaDo buys and sells. Information recorded about item types includes a unique identifier (code), name, and description. As mentioned above, "Number of item types" is a *measure* that determines how many different products there are.

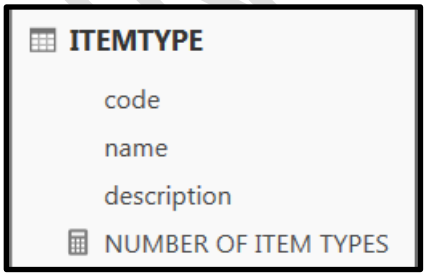


Figure 4.2–3
Structure of the Item Type Table

Figure 4.2–4 below shows the description for one specific item type; this is an **instance**. The Data panel in figure 4.2–1 shows multiple instances of ItemType (products).

code	name	description
ANG	Angwus	Crow Kachina

Figure 4.2–4
Description of Specific Item Type (Instance)

Terminology

An **instance** refers to a specific object. For example, Vincent Van Gogh is an instance of Painter, while the specific Honda Pilot I am driving is an instance of Car.

Artist **Who does KaDo buy products from?**

Figure 4.2–5 shows the structure of the Artist table. It describes whom KaDo buys from (vendors). Information recorded about artists includes a unique identifier (code), name, phone, fax, email, and address information (street, city, and state). Again, "Number of artists" is a *measure* that can be used in visualizations only. "Address" is an additional column that aggregates street, city, and state into one address field. As we will discuss in the next chapter, this column is created as a "data enrichment" that is needed for the definition of map charts. Figure 4.2–6 shows the description for one specific artist—an instance.

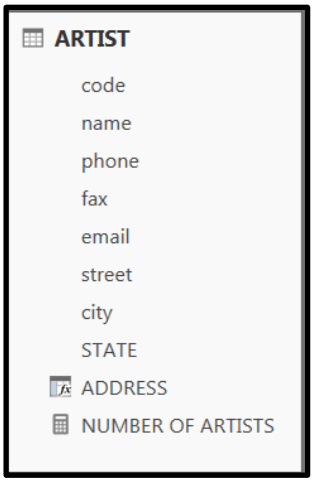


Figure 4.2–5
Structure Artist Table

code	name	phone	fax	email	street	city	STATE	ADDRESS
AHJON	Ahote Jones	480-982-5525	480-982-5526	ajones@yahoo.com	555 Idaho Rd.	Apache Junction	AZ	555 Idaho Rd.,Apache Junction,AZ

Figure 4.2–6
Description of Specific Artist (Instance)

AbleToCarve

What specific products (dolls) can KaDo buy from a specific vendor (artist)?

Figure 4.2–7 shows the structure of the AbleToCarve table, which describes who (which artist) is able to carve what—item type. The table thus links the other two tables—ItemType and Artist—together. Figure 4.2–8 shows a specific instance of this link.

Figure 4.2–7
Structure of the Artist Table

ARTIST	ITEMTYPE
AHJON	EOT

Figure 4.2–8
Description of a Specific Link (Instance) between Artist and Item Type

The data set described above should help to answer questions such as:

- How many products does KaDo offer (product assortment)?
- From how many different vendors does KaDo buy products?
- What products is an artist able to carve?
- How many products can an artist carve (diversity)?
- How many artists can carve a specific product type (choice)? Do any of the artists have a monopoly?
- How are the vendors spread out geographically?
- How are the products spread out geographically?

4.3 TOOLS FOR DEFINING DASHBOARDS

Figure 4.3–1 below shows the development of dashboards as a **process** that relies on different types of knowledge. First, it is important that you understand Power BI's report workspace (i.e., your toolbox): what a canvas is, what a visualization is, how to link data to a visualization, and how to format a visualization. We will briefly discuss the different parts of the report workspace in section 4.3.1 below. Second, dashboards are constructed from visualizations, so it is important that you understand what the purpose of the different visualizations is (i.e., their functionality) and how to implement them. Section 4.3.2 provides an overview of the different visualizations and what they aim to do (i.e., their functionality). Sections 4.4 through 4.8 provide an in-depth discussion of the implementation of five visualizations: cards, tables, stacked bar charts, slicers, and map charts.

Figure 4.3–1 further emphasizes that the design of good dashboards requires other types of knowledge. First, it is important to choose the most appropriate visualizations for the problem at hand—this is the "Select appropriate visualizations" step. To map correlations, what would be the best visualization: a scatter chart? Second, you need to understand the principles of good dashboard design, some of which include providing clear answers to the questions that are asked; creating simple, easy-to-understand dashboards; and making smart choices about colors. A discussion of such choices and principles is beyond the scope of this book. To learn more about good dashboard design, go to https://docs.microsoft.com/en-us/power-bi/power-bi-visualization-best-practices.

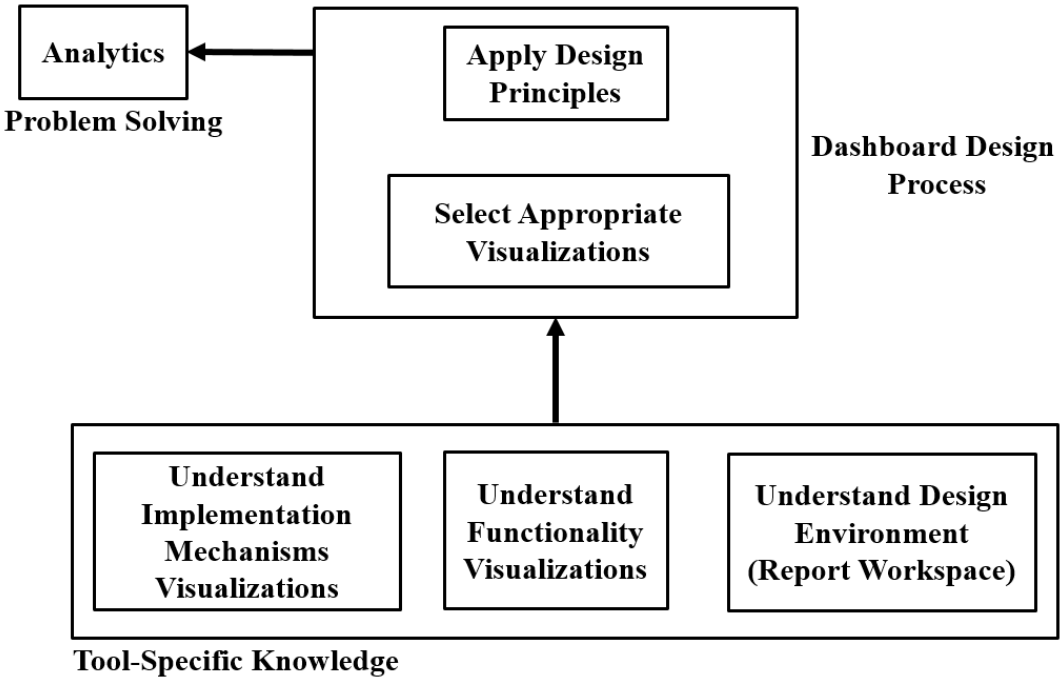


Figure 4.3–1
Designing Dashboards: Knowledge and Process

4.3.1 The Report Workspace

The report workspace is your toolbox for designing dashboards. Figure 4.3–2 divides the report workspace into eight parts, each of which is briefly discussed below.

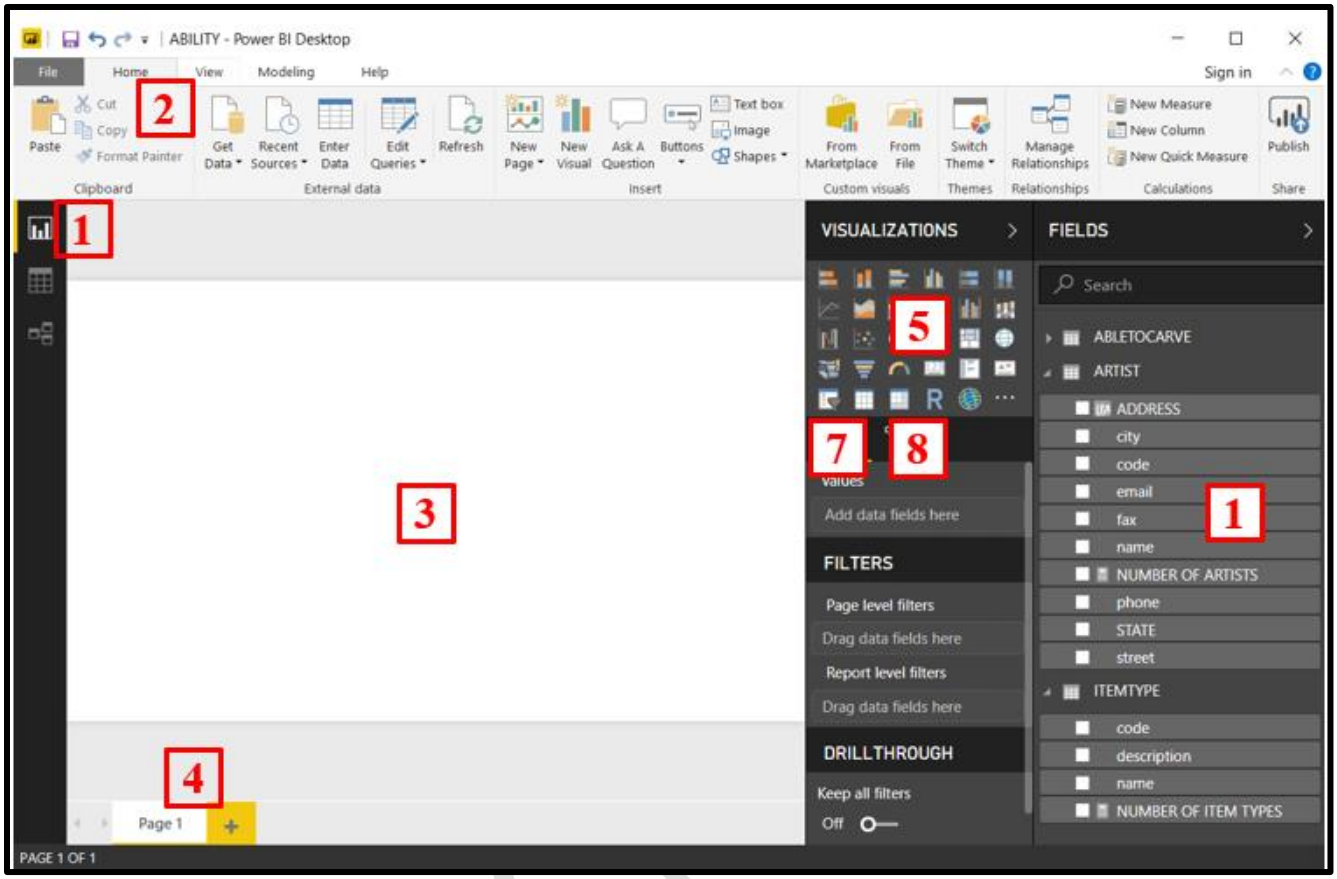


Figure 4.3–2
The Report Workspace

#1	Report View

Click on the Report view button to open the report workspace. This is where you will develop your dashboards.

#2	View Ribbon

Report view has its own main menu and a specific tab: View. Figure 4.3–3 shows the ribbon that goes with this tab. Among other things, it enables you to define a grid.

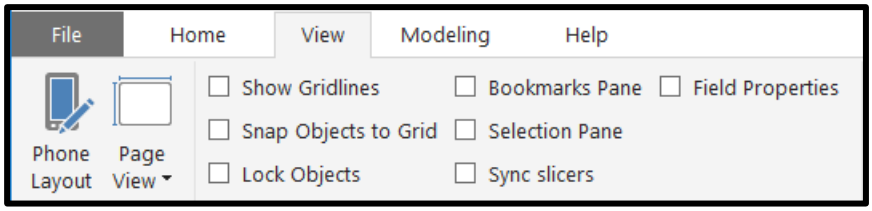


Figure 4.3–3
The "View" Ribbon

#3	Canvas

The canvas is the space in which you will design your dashboards. A dashboard consists of one or more visualizations that can interact with one another. You will develop a few interactive dashboards later in this chapter.

#4	Tabs

The same project can have multiple dashboards or pages; each page gets its own tab. Tabs can be duplicated, renamed, or deleted.

#5	Visualizations Panel

Each visualization has its own unique way of presenting data. The Visualization panel shows the different visualizations you can currently choose from. We will briefly discuss each of them in section 4.3.2 below.

#6	Fields Panel

The Fields panel is identical to the panels shown in the Data view. In this case, the Fields panel is used to select the data that will become part of the dashboard.

#7	Fields Tab

In this tab, you define how the data selected in the Fields panel should be used as part of a visualization. Data can be filtered or aggregated, among other things. The nature of the definitions will vary across the different visualizations.

#8	Formatting Tab

In the formatting tab, you define how the data in the dashboard should look. Characteristics that can be defined include text size, background color, and borders, among other traits. Formatting options will vary across the different visualizations.

4.3.2 An Overview of the Visualizations

Figure 4.3–4 below shows all the visualizations that are currently available.[9] Table 4.3–1 provides a brief description for each: number, icon, name, and the visualization's purpose (i.e., functionality). A shaded row refers to a visualization that we will discuss in more detail below.

In the final five sections of this chapter we will discuss the following visualizations, and their uses, in much more detail:

- Card
- Table
- Bar chart
- Slice
- Map

For each of these visualizations, we will discuss their creation in three steps:

1. How to define the visualization
2. How to put data into it
3. How to format it

[9] As of the December 2018 version. As discussed earlier, Microsoft adds new features to Power BI every month.

Figure 4.3–4
The "Visualization" Pane

Side Note

Next, we are going to create a few actual dashboards. **Make sure to have the ABILITY.PBIX file open at this time.**

Table 4.3–1
Overview of Visualizations[10]

#	ICON	NAME	FUNCTIONALITY
1		STACKED BAR CHART	Helpful for displaying how different categories relate to one another. The bars are oriented horizontally.
2		STACKED COLUMN CHART	Helpful for displaying how different categories relate to one another. The bars are oriented vertically.
3		CLUSTERED BAR CHART	Similar to the stacked bar chart, but the different categories (i.e., variables) can be shown side by side. The bars are oriented horizontally.
4		CLUSTERED COLUMN CHART	Similar to the stacked column chart, but the different categories (i.e., variables) can be shown side by side. The bars are oriented vertically.
5		100% STACKED BAR CHART	Similar to the stacked bar chart. Each bar represents 100%, and the relative importance of each variable is then indicated, using colors. The bars are oriented horizontally.
6		100% STACKED COLUMN CHART	Similar to the stacked column chart. Each bar represents 100%, and the relative importance of each variable is then indicated, using colors. The bars are oriented vertically.
7		LINE CHART	Displays the trend of one or more variables over time, or other sequential data.
8		AREA CHART	Similar to a line chart. An area chart displays the trend of variables over time; the areas between the axis and the line are colored.
9		STACKED AREA CHART	Similar to an area chart, but the values of the variables are now cumulative.
10		LINE AND STACKED COLUMN CHART	A combo chart that combines a line and a stacked column chart.
11		LINE AND CLUSTERED COLUMN CHART	A combo chart that combines a line and a clustered column chart.
12		RIBBON CHART	Similar to a stacked column chart, but sorted; it is especially useful for showing rank changes.
13		WATERFALL CHART	Shows increases and decreases in a variable and thus how the variable changes over time. Color coding can be used to indicate how a variable (e.g., income) changes over time.
14		SCATTER CHART	Plots two numeric variables against each other (X and Y axes) and helps to visualize possible correlations between them.
15		PIE CHART	Shows the relationship of parts to a whole (and thus relative proportions).

[10] Note that there are six shaded rows. The reason is that we cover both "bubble" maps and "filled" maps in the Map Charts section.

Table 4.3–1 (Continued)
Overview of Visualizations

#	ICON	NAME	FUNCTIONALITY
16		DONUT CHART	A doughnut chart shows the relationship of parts to a whole (and thus relative proportions); similar to a pie chart except that the center is blank.
17		TREEMAP	Useful for representing hierarchies with branches displayed in different colors, each containing additional rectangles (leaves).
18		MAP (BUBBLE Map)	Represents specific geographic points, and can be used to display relative proportions across locations. Maps are interactive in nature and can thus be used as a filter to select a more focus data set.
19		FILLED MAP	Represents regions such as states; can be used to display relative proportions across locations. Maps are interactive in nature and can thus be used as a filter to select a more focus data set.
20		FUNNEL	Shows how a sequential process moves through different stages.
21		GAUGE	A goal is represented by a line on a circular arc; shading is used to show progress toward that goal.
22		CARD	Shows a single number.
23		MULTI-ROW CARD	Used to display card-like tables. Every row in the table is presented as a card.
24		KPI	Visually represents the amount of progress made toward a measurable goal. (KPI = key performance indicator.)
25		SLICER	An interactive tool that can be used as a filter to select a more focused data set; slicers are used in combination with other visualizations.
26		TABLE	A grid with rows and columns; this is especially useful when the values of one or more categories need to be compared.
27		MATRIX	A tabular representation of data with the possibility of multiple grouped levels for both rows and columns.
28		R SCRIPT VISUAL	Provides an editor that allows you to write “R scripts,” of which the resulting visuals can be put on the canvas.
29		ArcGIS Maps	Sophisticated maps for advanced spatial analysis.
30		…. (ADDITONAL VISUALIZATIONS)	Custom visualizations provided by a community of developers; the number of visualizations is continuously growing.

4.4 CARDS

SHOWS A SINGLE NUMBER

A card can show any number, such as a sum or an average.

4.4.1 How to Create a Card

First, rename the Page 1 tab at the bottom of your canvas as “Cards.”

Next, click on the “card” icon in the visualization panel (figure 4.4–1).

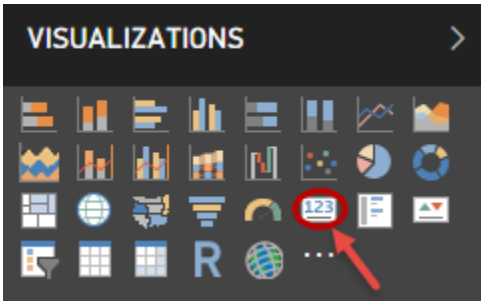


Figure 4.4–1
Creating a Card

An empty “card” will appear on the canvas (figure 4.4–2). Make sure that the visualization is active. The selection of fields and formatting will apply to **active** visualizations only.

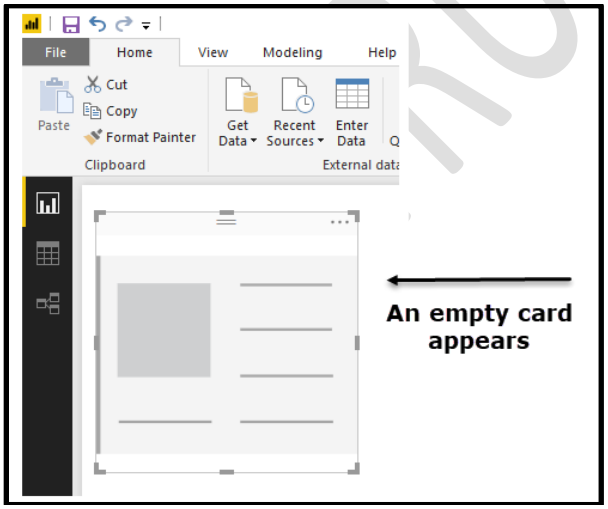


Figure 4.4–2
An Empty Card

4.4.2 Adding Data to a Card

Cards are used to show a single number. We will create a dashboard with two cards that contain the following numbers: “number of item types” (left) and “number of artists” (right).

- Number of item types: the number of different products offered (assortment size).
- Number of artists: the number of different vendors (artists) from whom the products can be purchased.

Let’s start with the first card, which shows the number of item types (i.e., products) KaDo offers. Go to the Fields panel (see figure 4.4–3 below) and click on the box in front of the field for which you want the value to show up in the card. For this exercise, go to the Fields panel and click on the box in front of the “NUMBER OF ITEM TYPES” field in the ITEMTYPE table. Make sure that the card on the canvas is active. You can only define the content of and/or format visualizations that are active.

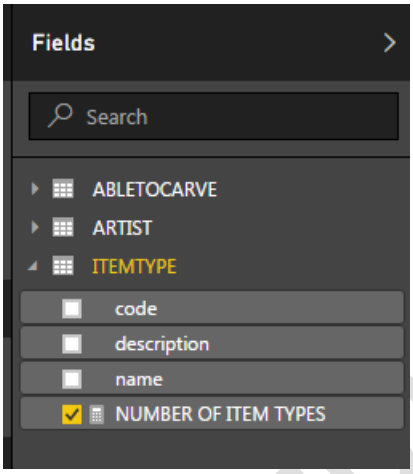


Figure 4.4–3
Selecting a Number to Put in the Card

Congratulations! You have just created your first card, visualization, and dashboard. Your canvas should now look as follows (figure 4.4–4).

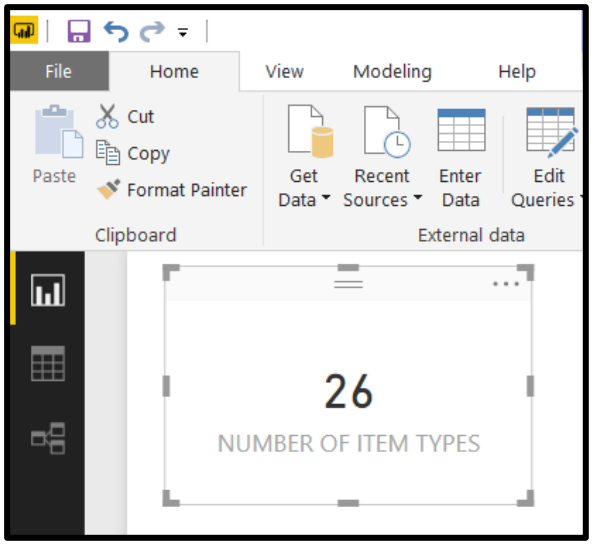


Figure 4.4–4
Definition of a Card with Data in the Canvas

The content of a visualization is shaped in the Fields tab, as is illustrated in figure 4.4–5 below. When you select a field from the Fields panel (right side in figure 4.4–5), the field is automatically added to the list of fields in the Fields tab (middle of figure 4.4–5). The content can then be further shaped in the Fields tab; for example, filters could be defined. We are not making any further changes at this time, but we will discuss more complex content definitions later in this chapter. The content in the canvas (left side of figure 4.4–5) is the result of the definitions in the Fields tab. Each type of visualization has its own specific "content definition" settings.

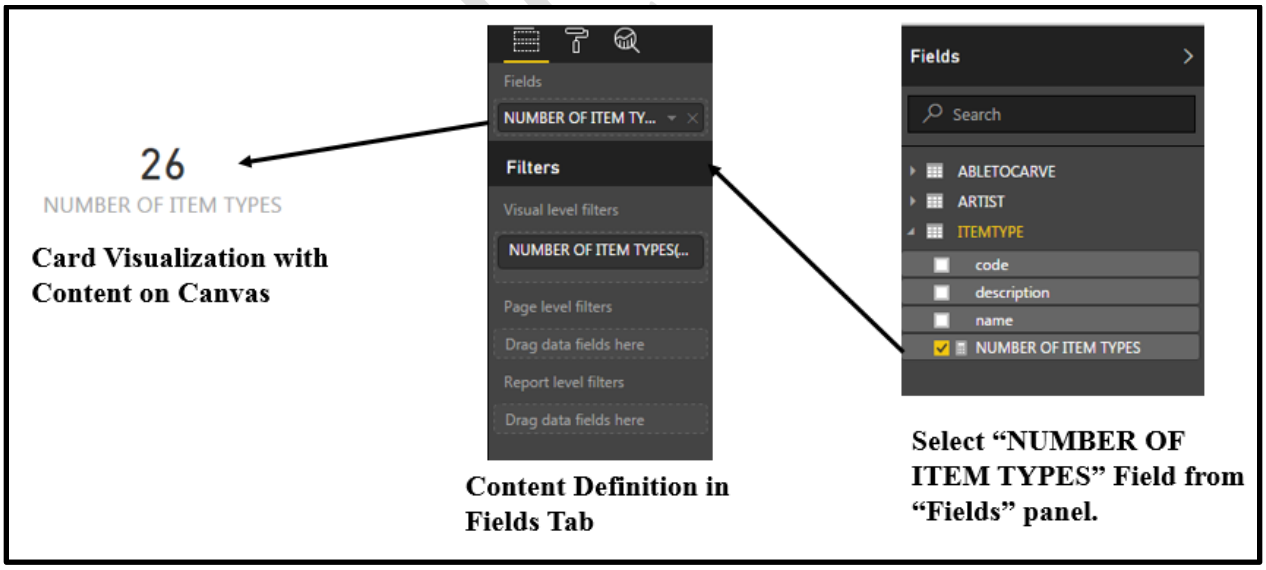
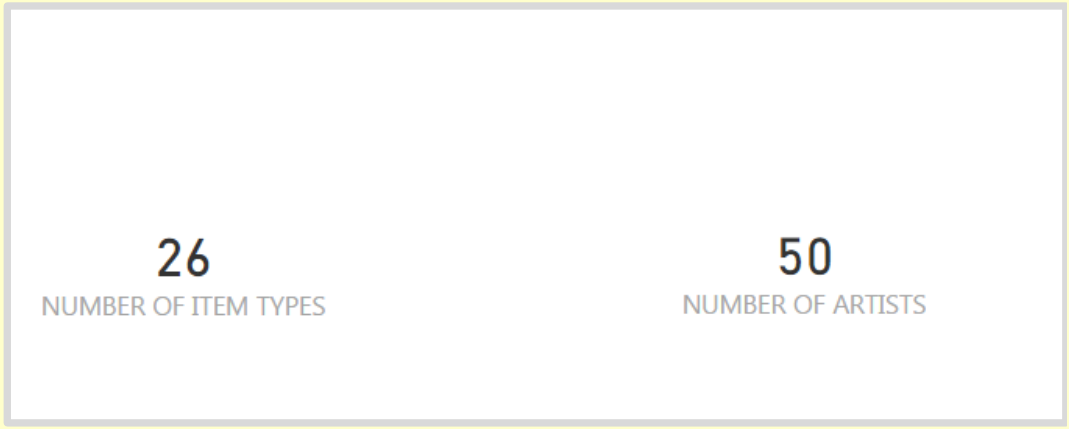


Figure 4.4–5
Content Definition

ASSIGNMENT 4.4-A1

Create a second card on your canvas that shows the "total number of artists." Your canvas should look as follows:



Side Note

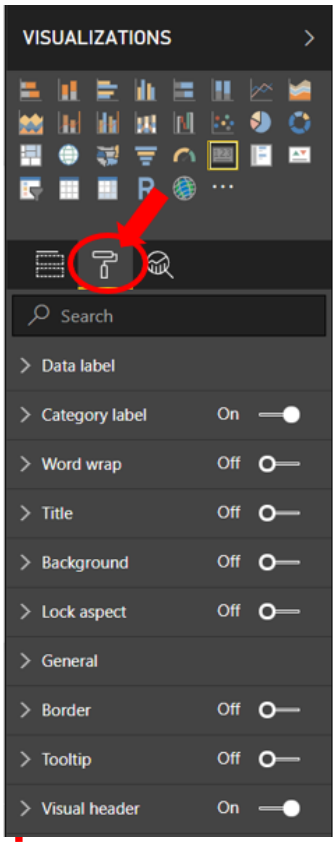
You can find the **solutions** for all exercises in chapter 4, including to the assignments, in the ABILITYSOLUTIONS.PBIX file.

4.4.3 Formatting Cards

Next, you will learn how to format cards, which we will do in a separate tab. Right-click on the Cards tab at the bottom of the canvas and choose "Duplicate page." Then rename the new tab "Cards Formatted."

Make sure your "Number of item types" card is active on the canvas and click on the "Format tab" in the Visualizations pane (see figure 4.4–6 below).

Figure 4.4–6
Formatting Cards



Click on the Format tab (red circle/arrow) in the Visualizations pane.

This menu will appear.

The formatting menu has many items. For most of these items, you first need to decide whether to activate the item by moving the slider from Off to On. Once activated, you can specify further details for most of the items. Our goal is to format the two cards as follows.

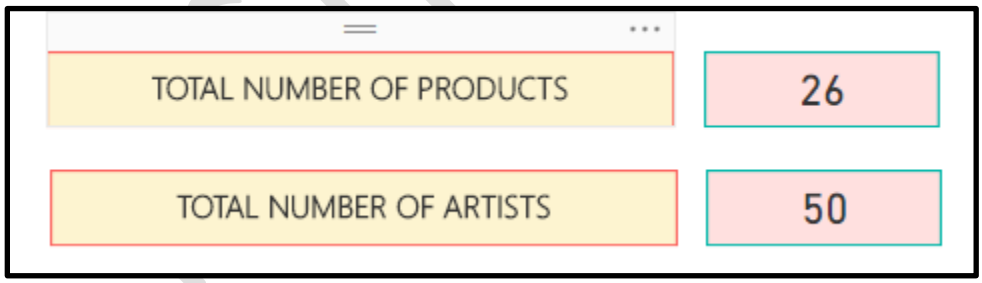


Figure 4.4–7
Formatted Cards

To format the first card, follow the two-step formatting process described below.

Step 1: Delete the “category label” and resize the card.

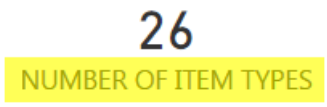


Figure 4.4–8
Category Labels

Cards use the associated field name as their default category label. To delete the label, make sure that the card is active, and change “Category label” from On to Off. Use the slider, as shown in figure 4.4–9 below.

Figure 4.4–9
Turning the Category Label Off

Slide from right (On) to left (Off)

Next, resize the card using the handles and change the text size of the “data label” to 16.
Use the “Text size” option under “Data label” and use decrease the size to 16.

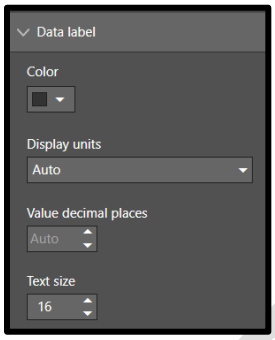


Figure 4.4–10
Changing the Text Size of the Data Label (the Number)

Next, turn the background on and change the color to pink.

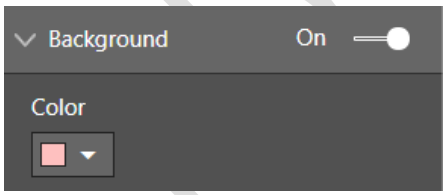


Figure 4.4–11
Changing the Background Color

Finally, add a border to the card.

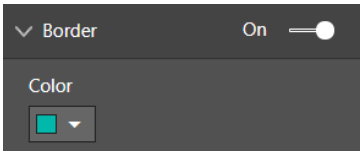


Figure 4.4–12
Adding a Green-Colored Border to the Card

Step 2: Add a text box to the dashboard

Power BI makes it easy to add text boxes, images (such as logos), rectangles, and other items to a dashboard.

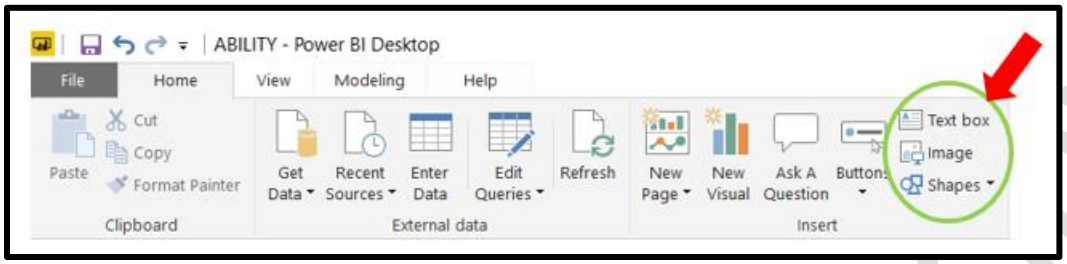


Figure 4.4–13
Adding Text Boxes and Images to a Power BI Dashboard

Next, we will illustrate how to add text boxes to our dashboard. The text boxes will be used to describe what the numbers in the cards mean. Text boxes are visualizations themselves and can therefore be formatted.

Create a text box. Click on the "Text box" icon shown in figure 4.4–13, and a text box will appear in your canvas (see figure 4.4–14).

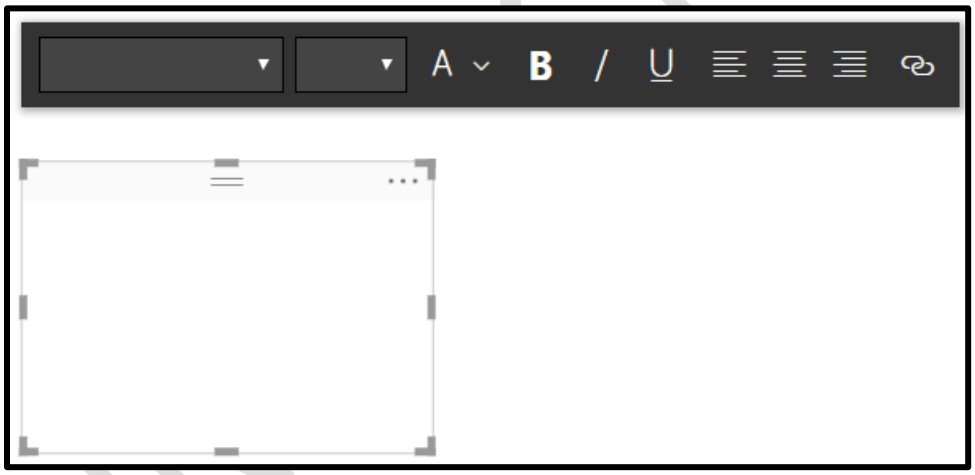


Figure 4.4–14
Creating a Text Box

Enter text, center it, and change the font (figure 4.4–15).

Figure 4.4–15
Editing the Text (Changing the Font)

Activate the background and change the color.

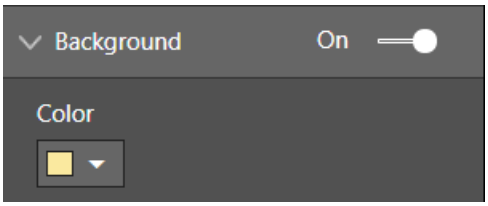
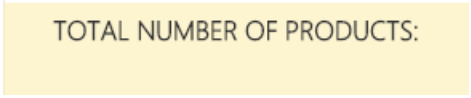


Figure 4–16
Changing the Background of the Text Box



Add a border to the text box.

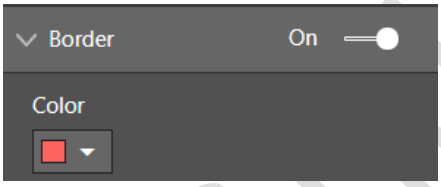
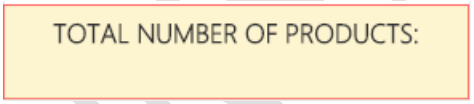


Figure 4.4–17
Adding a Border to the Text Box

Your text box should now look like this:



ASSIGNMENT 4.4.-A2

Format the "Total Number of Artists" text box yourself.

4.5 TABLES

A GRID WITH ROWS AND COLUMNS

A table is a grid with rows and columns, which is especially useful for comparing the values of one or more categories.

4.5.1 How to Create a Table

First, create a new page (Tab) and name it "Table."

Next, click on the Table icon in the Visualizations panel (figure 4.5–1).

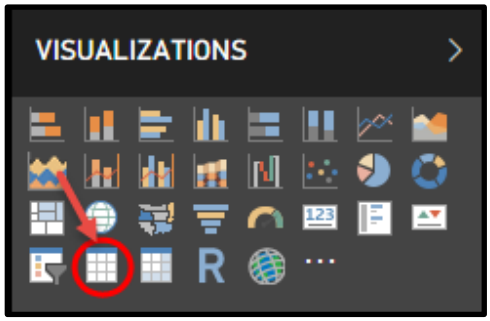


Figure 4.5–1
Create a Table

An empty table will appear on the canvas (figure 4.5–2). Make sure the visualization is active. The selection of fields and formatting will always apply to **active** visualizations.

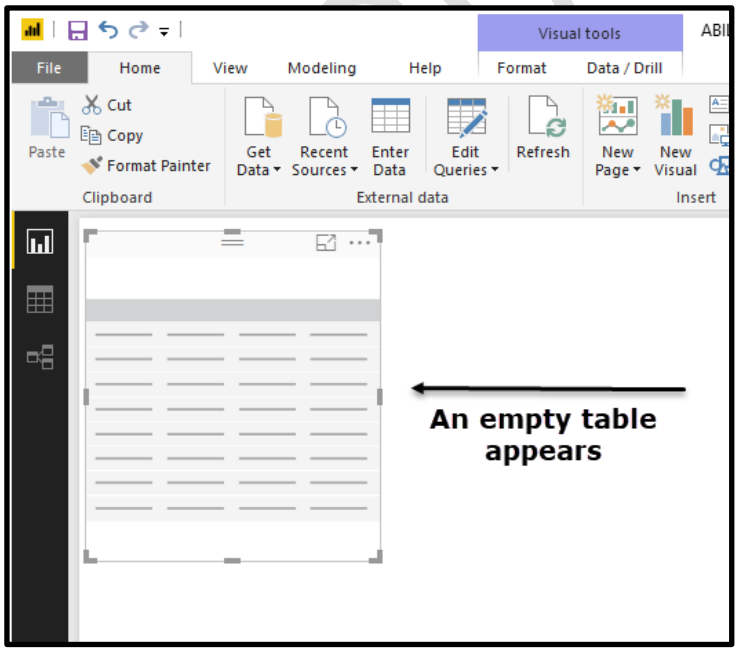


Figure 4.5–2
An Empty Table

4.5.2 Adding Data to a Table

Tables can be used in many different ways—for example, to show detailed transaction records such as sales. Also, tables are often used to show summaries, such as the total revenue ($) generated for a product type, per division, etc. The example below shows this second use of tables; the table shows how many different item types (product types) an artist (vendor) is able to deliver.

Click on the checkboxes in the Fields panel to determine which data should be included in the table. In figure 4.5–3, two fields are selected: “Name” from the Artist table and “Number of item types” from the ItemType table.

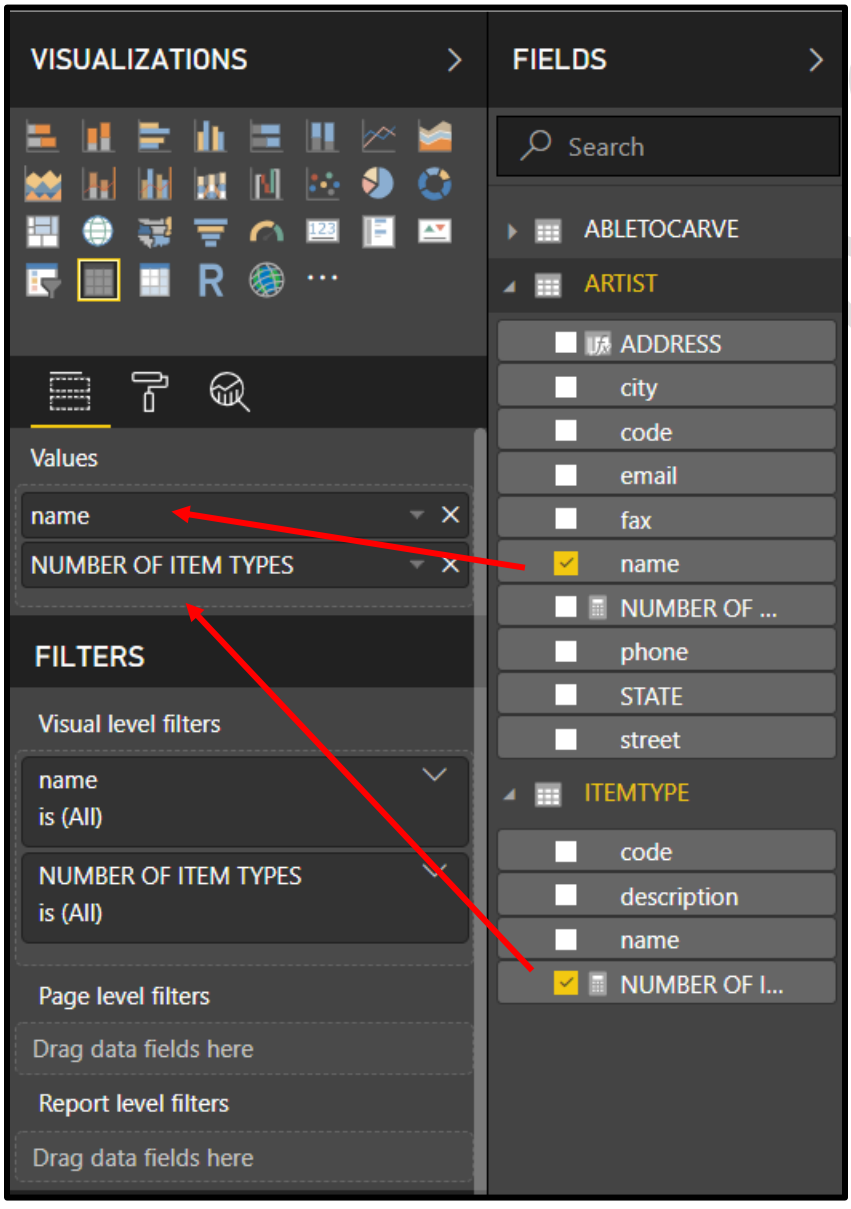
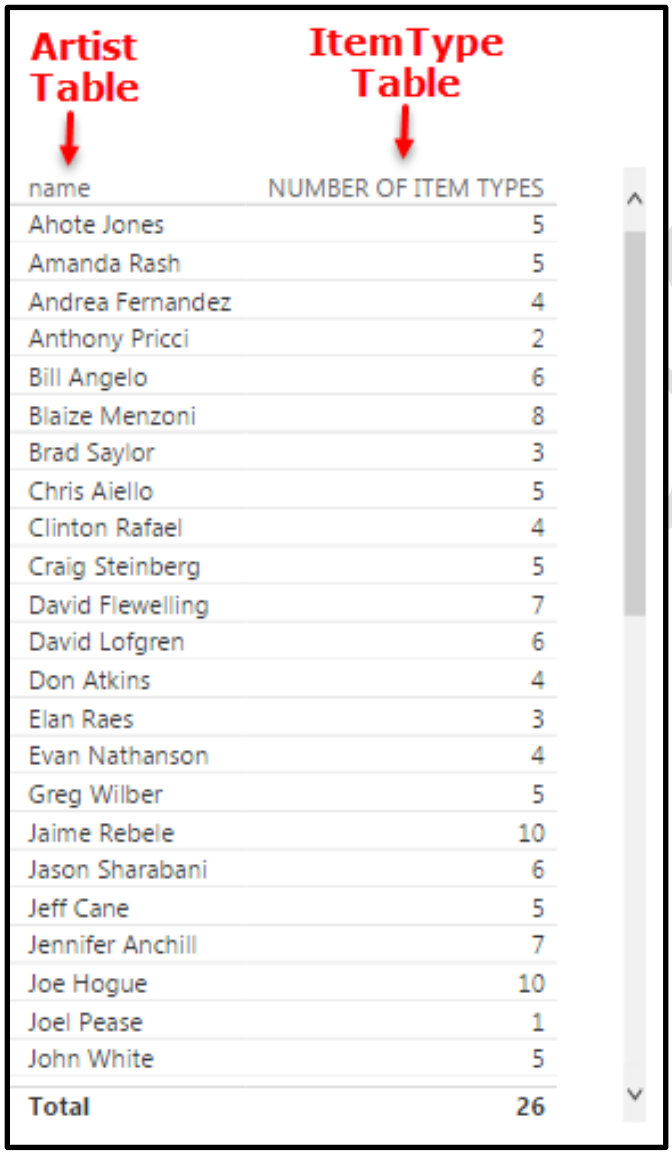


Figure 4.5–3
Content Definition and Field Selection

Side Note

Note how easy it is to build visualizations and dashboards with data from different tables. As discussed above, the relationships between the tables are defined in the "data model."

The Fields tab in figure 4.5–3 (the Visualizations panel) shows how the content of the table is defined. The two selected fields show up under Values. Each field in the Values section of the Fields tab (Visualization panel) is represented as a column in the table. You can easily add columns, delete columns, or rearrange the order of the columns, among other things. The resulting table (on the canvas) is shown below in figure 4.5–4.



name	NUMBER OF ITEM TYPES
Ahote Jones	5
Amanda Rash	5
Andrea Fernandez	4
Anthony Pricci	2
Bill Angelo	6
Blaize Menzoni	8
Brad Saylor	3
Chris Aiello	5
Clinton Rafael	4
Craig Steinberg	5
David Flewelling	7
David Lofgren	6
Don Atkins	4
Elan Raes	3
Evan Nathanson	4
Greg Wilber	5
Jaime Rebele	10
Jason Sharabani	6
Jeff Cane	5
Jennifer Anchill	7
Joe Hogue	10
Joel Pease	1
John White	5
Total	**26**

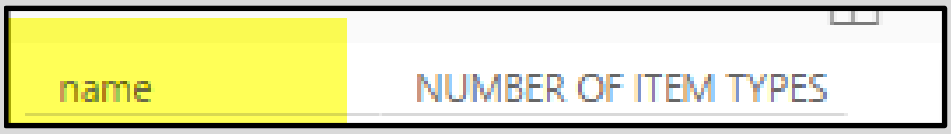
Figure 4.5–4
The Table

This is how the table defined in figure 4.5–3 (Fields tab) looks like on the canvas before any formatting is applied.

Both column names are the same as the field names you have selected. The "Number of item types" counts how many times an artist occurs in the AbleToCarve table—i.e., how many different types of kachina dolls the artist can carve. Totals are added at the bottom of the table by default. In this case, the 26 at the bottom refers to the total number of different item types and is therefore not the same as the sum of all numbers in the "Number of item types" column.

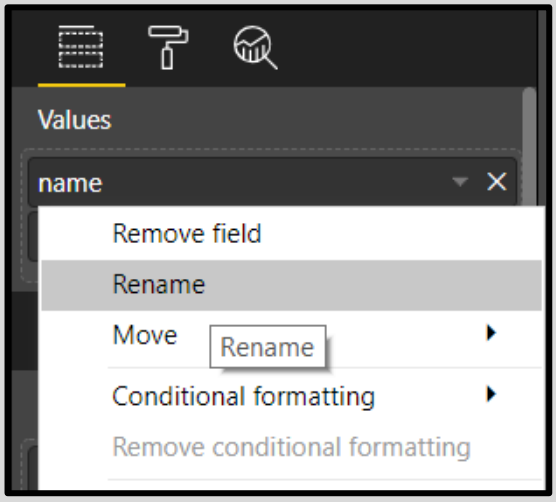
Side Note

When building dashboards, one of your goals is to present what the information means as clearly as possible. For example, you might consider replacing the column headers in figure 4.5–4:



with:

As shown below, to change the header of the first column, select the "name" field in the Fields tab, right-click, and select "Rename." Replace "name" with "ARTIST NAME." This option allows you to change field names for a specific visual (e.g., the table in this example) while keeping the field name in the Fields pane.



4.5.3 Formatting Tables

Make sure that the table (canvas) is active, and then click the Format button in the Visualizations panel. As shown in figure 4.5–5 below, a number of formatting options, specific to tables, appear.

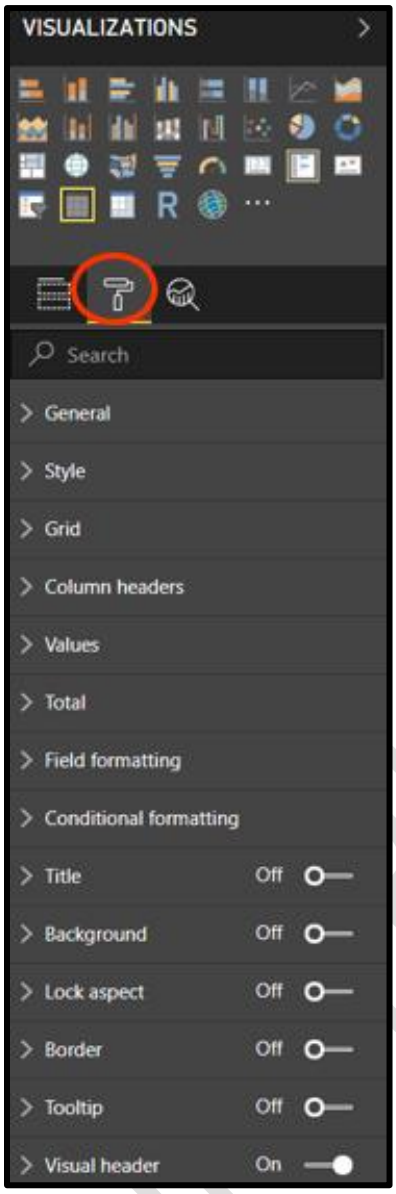


Figure 4.5–5
Formatting Options for Tables

Our goal is to transform the raw table in figure 4.5–4 into the formatted table below (4.5–6). This exercise will teach you about some of the basic formatting options available for tables.

ARTIST NAME	NUMBER OF ITEM TYPES
Ryan Tyas	11
Jaime Rebele	10
Joe Hogue	10
Blaize Menzoni	8
Valerie Baldassari	8
David Flewelling	7
Jennifer Anchill	7
Mark Boaman	7
Maska Kroes	7
Bill Angelo	6
David Lofgren	6
Jason Sharabani	6
Keme Potoms	6
Leslie Hitchens	6
Mark Hofferd	6
Mike Drennen	6
Ray Lenno	6
Steve Forrest	6
Ahote Jones	5
Amanda Rash	5
Chris Aiello	5
Craig Steinberg	5

Figure 4.5–6
Formatted Table

First, change the text size of the data in the table to 16 and change Totals to Off (see figure 4.5–7).

Side Note

Power BI adds the totals by default; i.e., it adds the total number of products: 26 (see figure 4.5–6). In this case, however, given that different artists can carve (i.e., deliver) the same item type (product), the sum of the numbers in the "Number of item types" column does not equal 26.

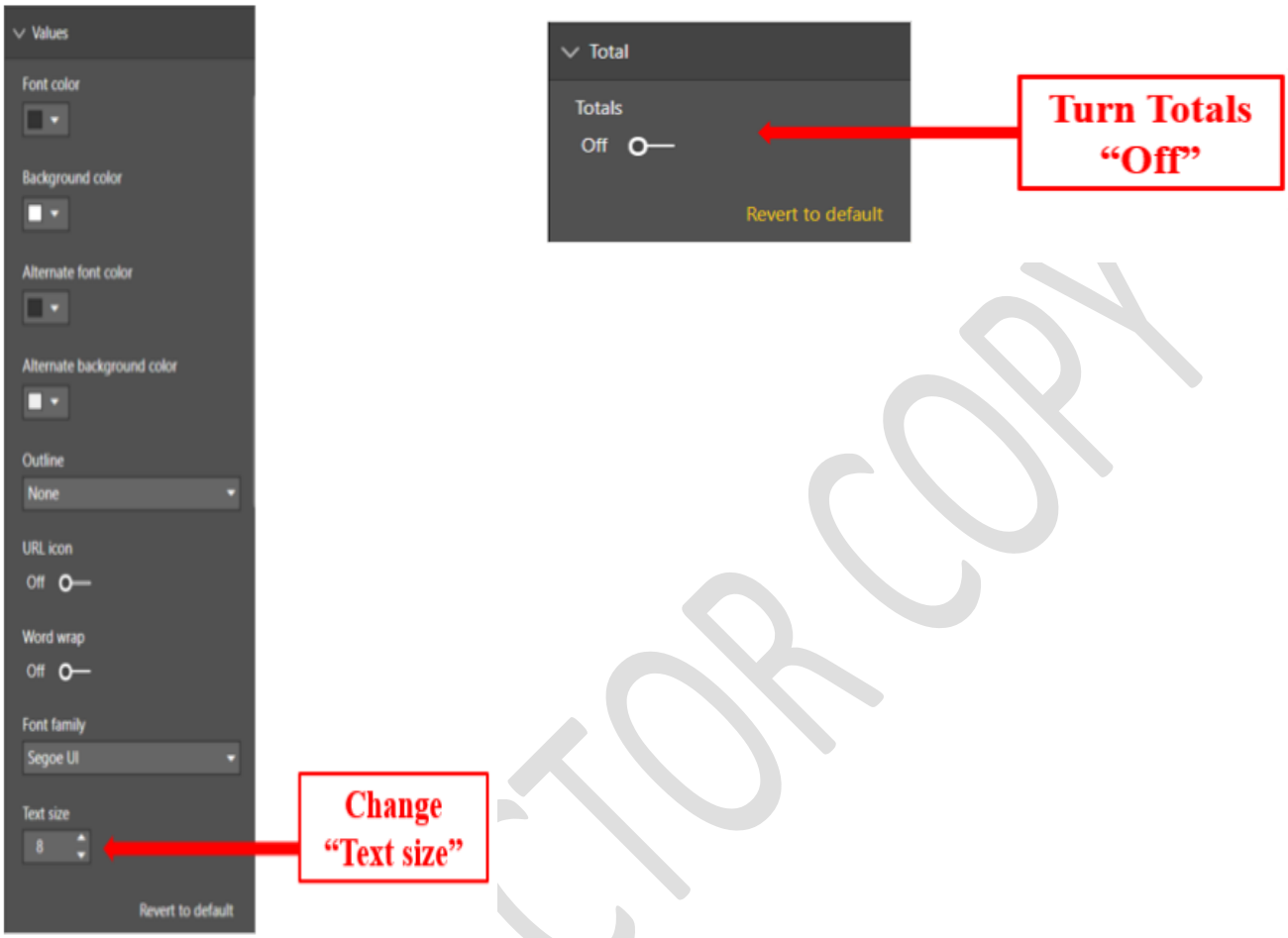


Figure 4.5–7
General Formatting Options

Next, turn on the horizontal grid to show lines between the different data rows (figure 4.5–8).

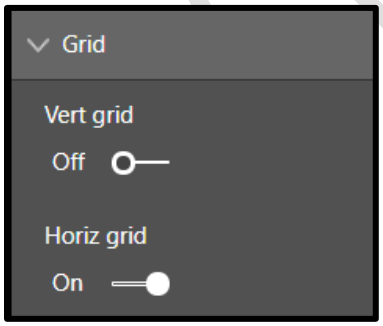


Figure 4.5–8
Turning the Horizontal Grid On

Next, change the formatting of the column headers. Change the background to black and use white as the color for the text (figure 4.5–9).

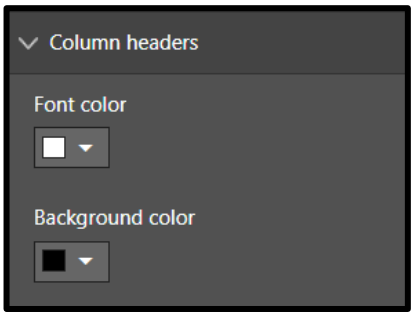


Figure 4.5–9
Changing the Format of the Column Headers

Next, change the color of the data (Values) in the table to blue (figure 4.5–10).

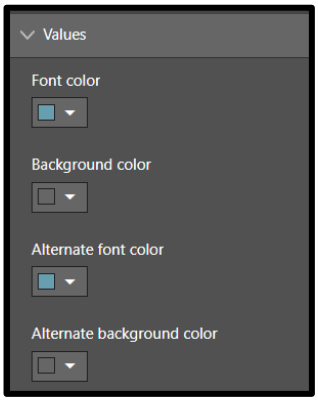


Figure 4.5–10
Changing the Color of Data in the Table

Then, add a border to the table (figure 4.5–11).

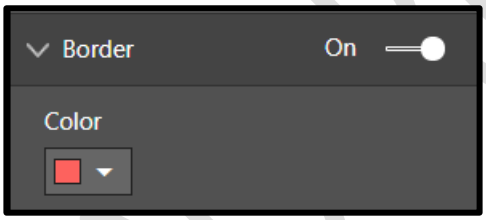


Figure 4.5–11
Adding a Red Border to the Table

Finally, we want to sort the "Number of item types" in descending order. Doing so will help with understanding who the most diverse artists (vendors) are—the ones who are able to carve many different types of kachina dolls. Click on the table on the canvas to make the table active. Then click on the triangle pointing down for the "Number of Item Types" column, as shown in figure 4.5–12.

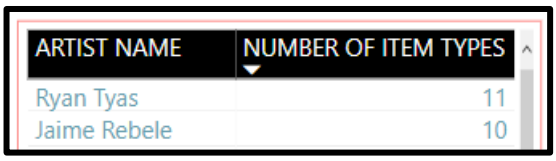


Figure 4.5–12
Listing "Number of Item Types" in Descending Order

ASSIGNMENT 4.5-A1

Create a new tab and name it ASSIGNMENT TABLE: NUMBER OF ARTISTS PER ITEM TYPE. Then create a similar table that will determine how many artists are able to carve each of the different item types. The results are shown below (figure 4.5.-13). Now, come up with your own formatting.

name	NUMBER OF ARTISTS
Angwus	16
Chusona	15
Eototo	14
Chop	13
Hospoa	13
Soyal	13
Kwahu	12
Pong	12
Kachin Mana	11
Mosairu	11
Nuvak'China Mana	11
Wupamo	11
Heee	10
Kawaii	10
Kwikwilyaka	9
Pawik'China	9
Poli Mana	9
Tocha	9
Wuyak-Kuita	9
Hon	8
Kowako	8
Mongwu	8
Wakas	8
Nakiachop	7
Kweo	5

4.6 STACKED BAR CHARTS

USE HORIZONTAL BARS TO MAKE COMPARISONS BETWEEN CATEGORIES

A bar chart shows how different categories relate to one another using a numeric field that is represented as a bar. A stacked bar chart uses multiple fields to compare the categories.

4.6.1 How to Create a Bar Chart

You can create a bar chart by clicking on the stacked bar chart icon in the Visualizations panel (see figure 4.6–1). But **don't click**!

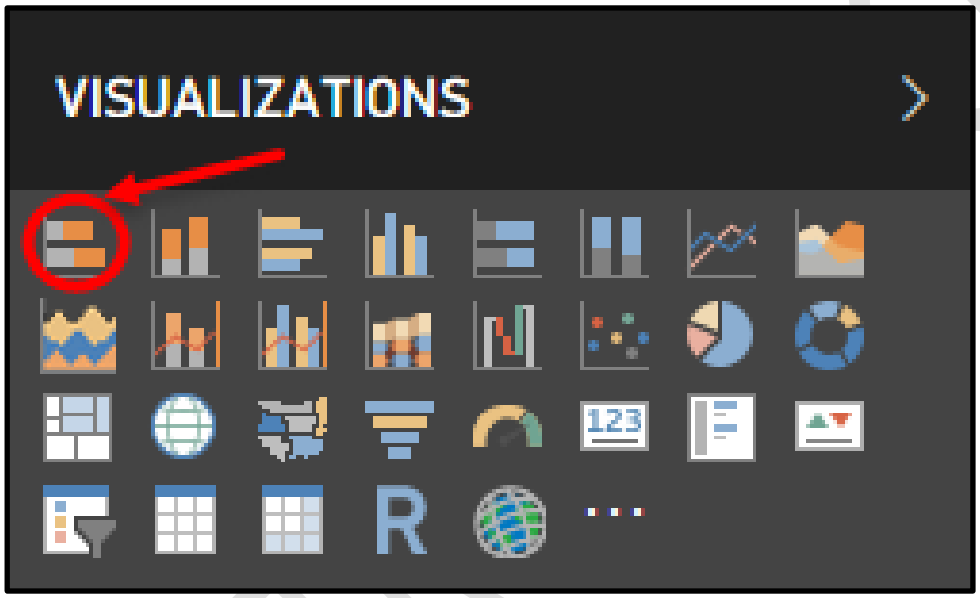


Figure 4.6–1
Creating a Bar Chart

Instead of creating a new bar chart, let's transform the table you created for the assignment in 4.5 (4.5–A1) into a bar chart. Transformations from one type of visualization to another type are common practice. Go to the "ASSIGNMENT TABLE: NUMBER OF ARTISTS PER ITEM TYPE" tab, right-click, and select "Duplicate page." To replace the table with a bar chart, click on the "Stacked bar chart" icon in the Visualizations panel. In its raw format (i.e., without formatting), the bar chart will look as follows (figure 4.6–2):

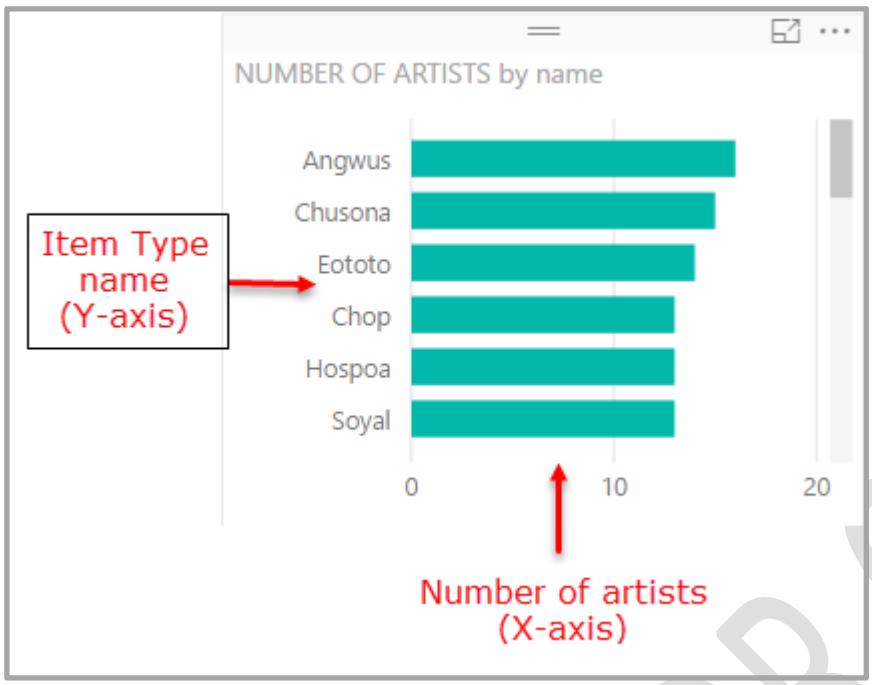


Figure 4.6–2
Unformatted Bar Chart

This chart shows how many artists are able to carve each item type.

4.6.2 Adding Data to a Stacked Bar Chart

Given that we are using the same data set for the definition of two different visualizations—a table and a stacked bar chart—we will next compare the two visualizations' content definitions (see figure 4.6–3 below). The left side shows the content definition for the table, while the right side shows the content definition for the stacked bar chart.

The two fields involved—"Name" and "Number of artists"—are shaped in different ways in the Fields button. For the table, both fields are found under Values and represent two columns. For the bar chart, the field under Axis is used to show different categories on the Y-axis (name). The field under Value is used for comparison purposes; the length of the bar is defined by "Number of artists" (X-axis). Dragging two or more fields into the Values section results in a "stacked" bar chart (see Side Note below).

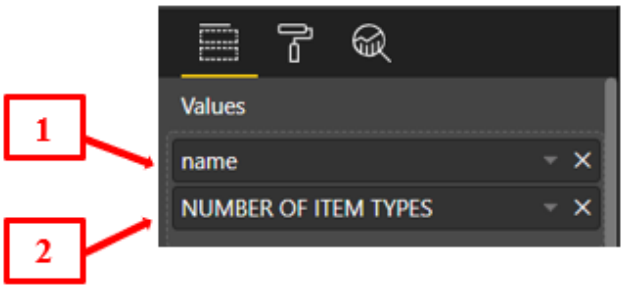
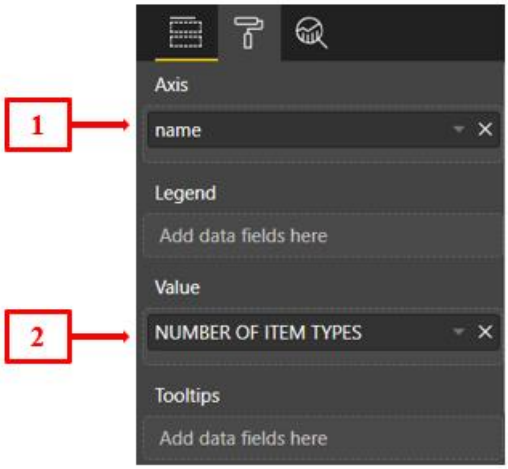
Table Definition	Stacked Bar Chart Definition
	

Figure 4.6–3
Comparison of "Content Definitions" for Table and Bar Chart for the Same Data Set

4.6.3 Formatting Bar Charts

Formatting bar charts is similar to the formatting of cards and tables, discussed above. The following are a few formatting options specific to bar charts.

As shown in figure 4.6–4 below, the formatting of both axes is straightforward. Among others, the Y-axis can be repositioned (left versus right), and the title (name) can be reformatted; for example, the color can be changed.

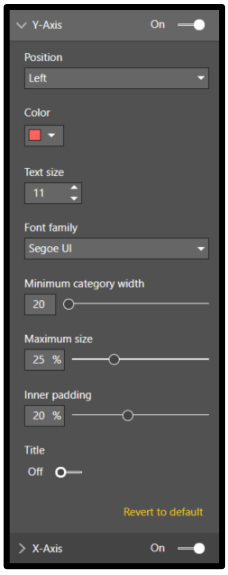


Figure 4.6–4
Formatting the Y-Axis of a Bar Chart

Also, as illustrated in figure 4.6–5 below, by turning the data label on, the actual numbers representing the length of each of the bars (i.e., the number of artists) are shown. The numbers can then be formatted further, by color, text size, etc.

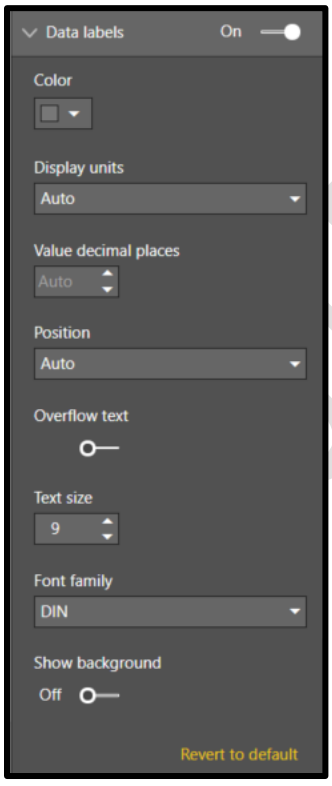
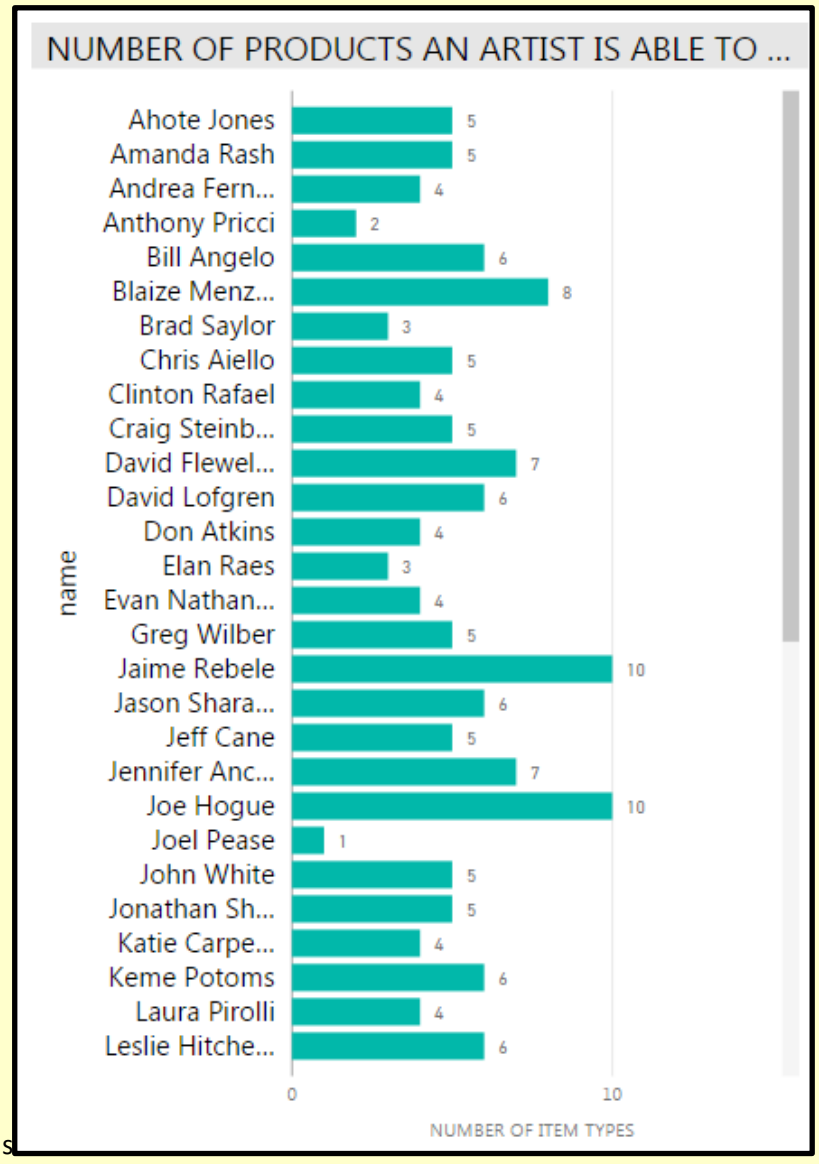


Figure 4.6–5
Adding Data Labels to a Bar Chart

ASSIGNMENT 4.6-A1

Create a similar bar chart that shows the number of products (type) an artist is able to carve. Figure 4.6–6 shows such a chart, but feel free to experiment! Note that I have included labels for both the X-axis (number of item types) and the Y-axis (name).

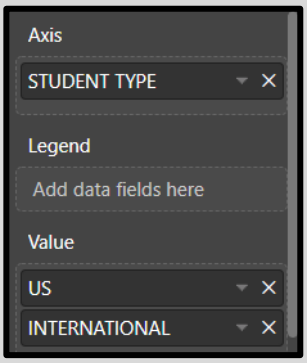


Side Note

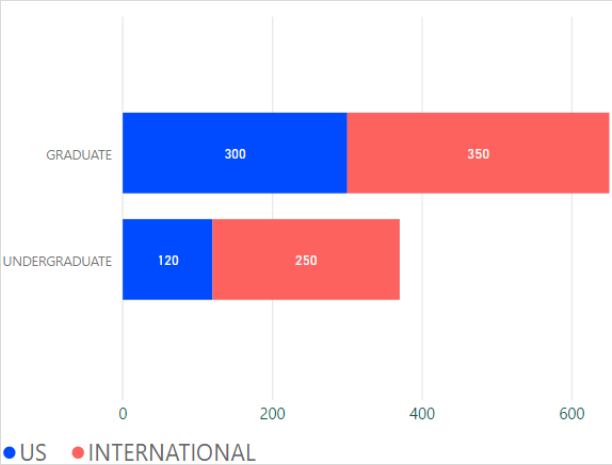
The examples above discuss how to create a bar chart, although the concept of a stacked bar chart has not been illustrated. We will do that now with a simple example. Open the STACKED.pbix file in your DropBox and look at the contents of the enrollments table.

STUDENT TYPE	US	INTERNATIONAL
UNDERGRADUATE	120	250
GRADUATE	300	350

Next, let's create a new dashboard. Rename the "Page 1" tab as "Stacked Bar Chart." Use the following content definition:



Similar to what we discussed above, the Y-axis represents the different types of students (STUDENT TYPE): "undergraduate" and "graduate." What is new is that we are dragging two different fields into the Value section: "US" and "INTERNATIONAL." As a result, the total length of the bar shows the total number of students of that type enrolled (e.g., undergraduate students). By dragging both fields, the "composition" of the total number of enrollments in terms of US and International students is indicated as well. The bars are now divided into US and International stacks. The resulting dashboard, with a bit of additional formatting, is shown below:



4.7 SLICERS

INTERACTIVE VISUALS THAT ENABLE THE CREATION OF FOCUSED DATA

A slicer is an interactive tool that can be used as a filter to select a more focused data set. The fact that any combination of values can be chosen makes the slicer a powerful tool. Given the purpose of slicers, they always interact with other visuals. They are similar to slicers in Excel.

4.7.1 How to Create a Slicer

Let's use the assignment from section 4.5 (4.5–A1) as a starting point again here. Go to the Assignment table: "Number of artists per item type" tab and duplicate it—right-click and select "Duplicate page." Then rename it SLICER.

Next, let's add a slicer. You can create one by clicking on the slicer icon in the Visualizations panel (see figure 4.7–1).

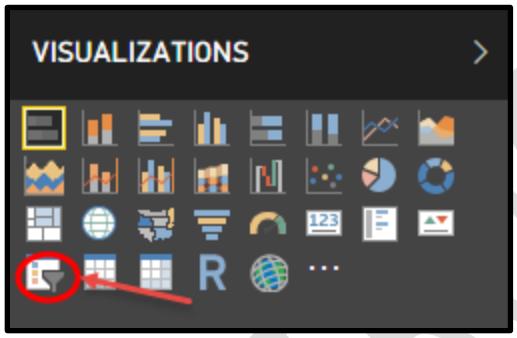


Figure 4.7–1
Creating a Slicer

An empty slicer will appear on the canvas (see figure 4.7–2). Make sure the visualization is active. The selection of fields and formatting will always apply to **active** visualizations.

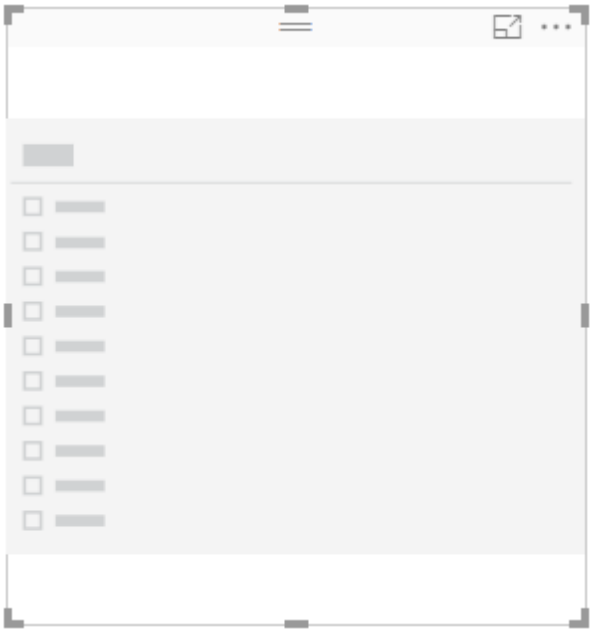


Figure 4.7–2
An Empty Slicer

4.7.2 Adding Data to a Slicer

Slicers are used to dynamically select a more focused data set, such as all sales transactions for a specific customer or all sales transactions for a specific region. The dashboard we are now creating makes use of two visualizations to demonstrate how slicers work. The first visualization, the slicer itself, allows us to select a state or states. The second visualization, a table (the focused data set), shows us the product types (items) and the number of different artists available per product (item) type. This is the visualization you created for your assignment in section 4.5 (4.5–A1). It currently should be on your canvas, since you duplicated the "ASSIGNMENT TABLE: NUMBER OF ARTISTS PER ITEM TYPE" page.

Let's start with the content definition of the slicer.

Click on the box preceding STATE in the Fields panel. The Field section of the Fields button in the Visualizations panel then shows which field will be used to instantiate the slicer in the canvas (STATE).

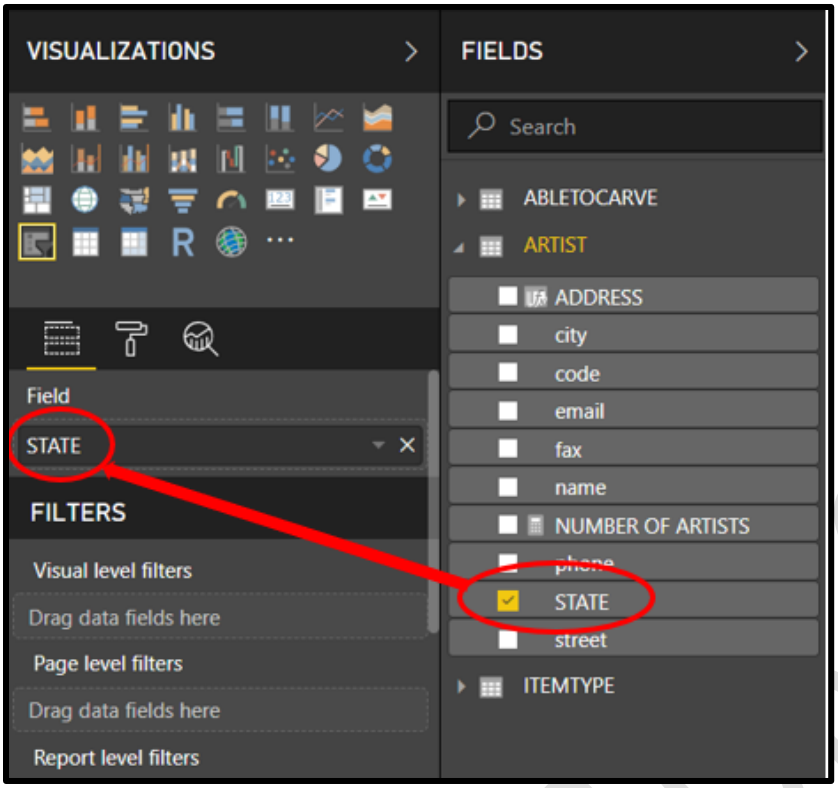


Figure 4.7–3
Content Definition: Which Content Goes into a Slicer

The result is the slicer shown in figure 4.7–4 below. The figure shows all the states in which vendors (artists) whom KaDo does business with live. No duplicates are shown.

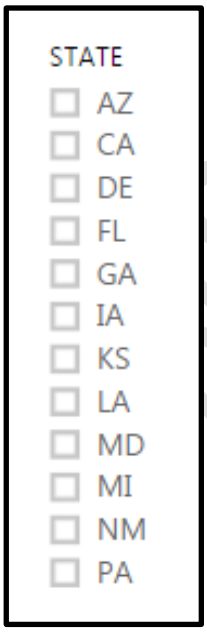


Figure 4.7–4
A Slicer for States

Slicers are interactive visualizations that allow you to dynamically create focused data sets. As mentioned above, they always interact with other visualizations. At this point, your canvas should look as shown in figure 4.7–5 below. The right side shows the slicer. The left side shows the focused data set, of which the content dynamically changes, depending on the selections made in the slicer. The two visualizations thus interact. We are only using one slicer in this case, but you can include multiple slicers in the same dashboard if you wish.[11]

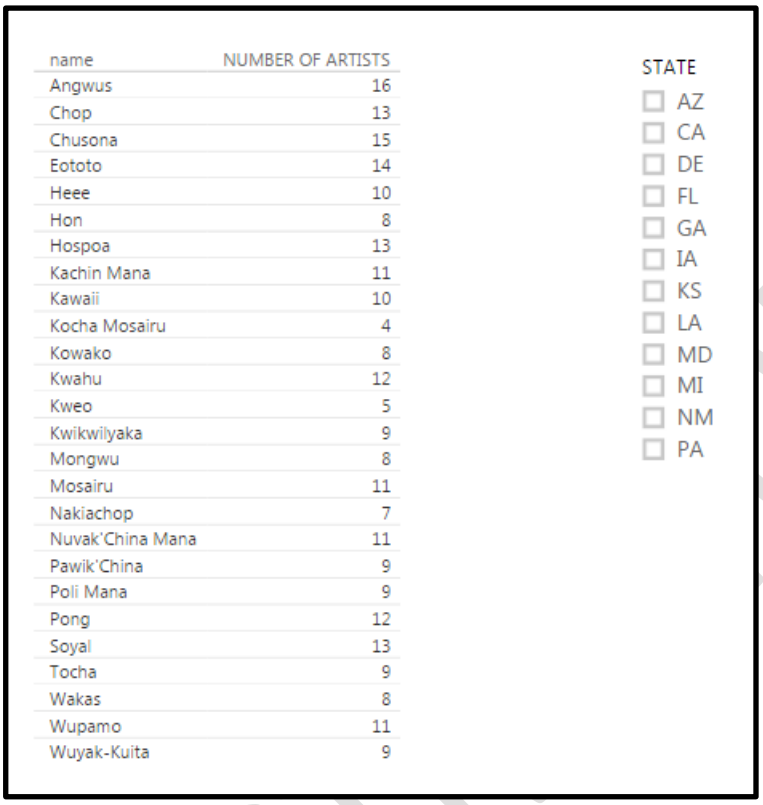


Figure 4.7–5
Interaction between a Table and a Slicer

Test it out! Click on any combination of states to obtain a more focused data set. For example, to better understand the offerings in the mid-Atlantic states, click on DE, MD, and PA. The left side in figure 4.7–6 below shows the focused data set for the mid-Atlantic area.

Useful Tip

Use the Ctrl key to select multiple boxes.

[11] This is common practice. The case study in chapter 7 illustrates the use of multiple slicers.

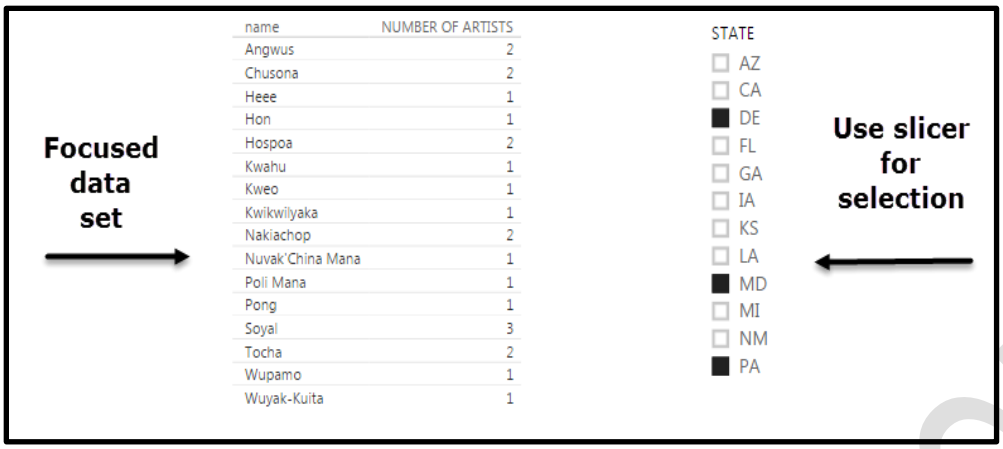


Figure 4.7–6
Interactive Use of a Slicer

4.7.3 Formatting Slicers

Similarly to other visualizations, slicers have their own unique set of formatting options. Next, we will use some of these options to transform the dashboard shown in figure 4.7–6 into the dashboard shown in figure 4.7–7.

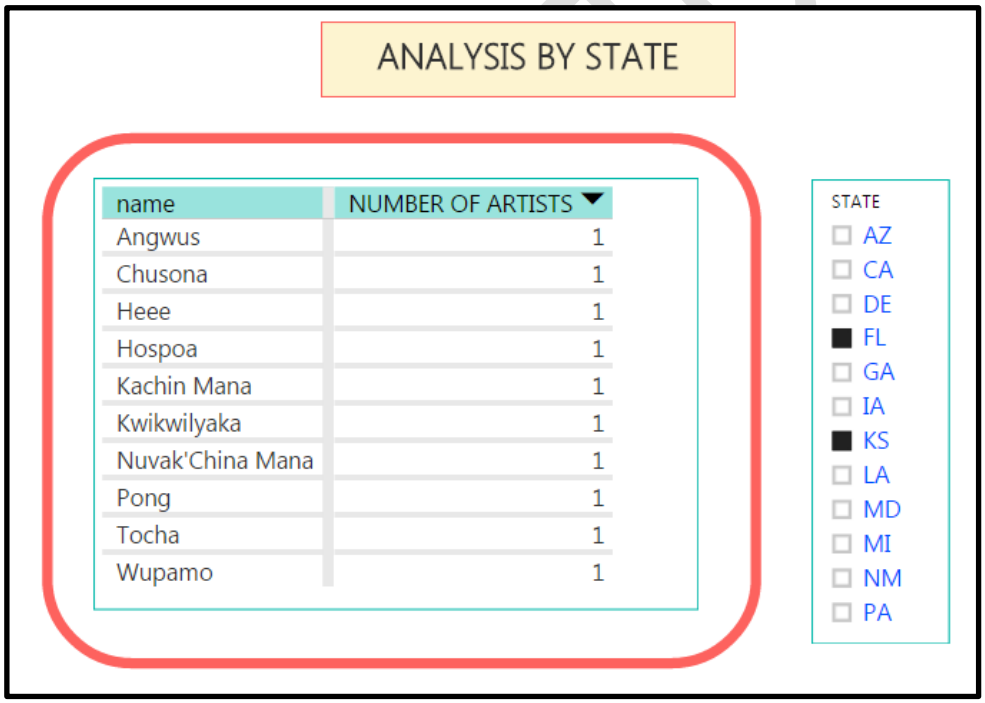


Figure 4.7–7
Formatted Interactive Dashboard with Slicer

First, according to our discussion in 4.4.3, add a text box to your dashboard.

Second, change the appearance of the data in the slicer—both font color and text size—as shown in figure 4.7–8 below. Refer to the discussion in section 4.5.3 to help you format the table.

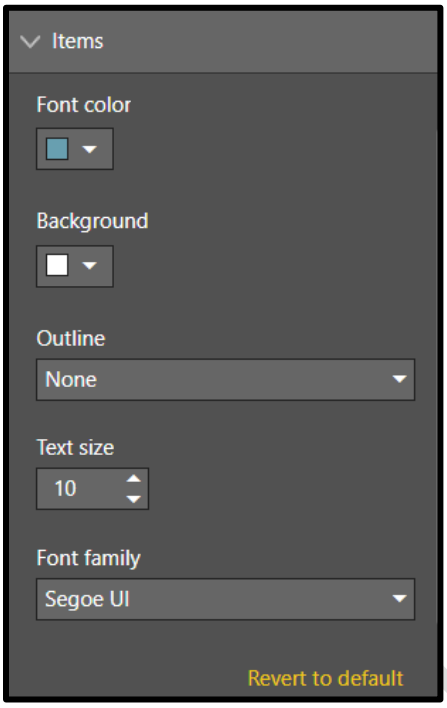


Figure 4.7–8
Changing the Appearance of the Data in a Slicer (Changing Text Size)

Third, draw a border around the slicer. Figure 4.9–10 shows how to do this. Do the same for the text box.

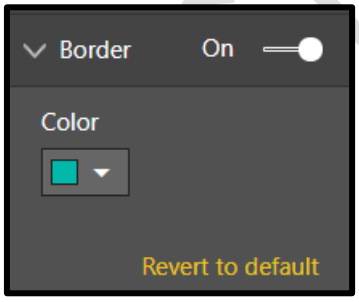


Figure 4.9–10
Drawing a Border around the Slicer

Finally, we need to add a rectangular box around the table. Choose Home→Shapes→Rectangle, as shown in figure 4.7–11 below.

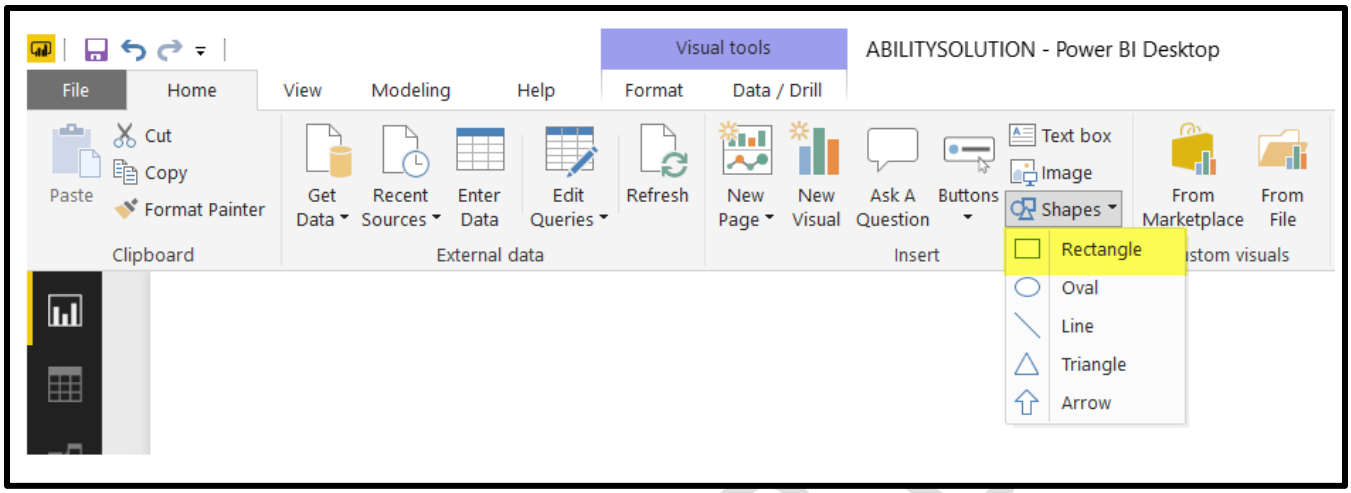


Figure 4.7–11
Drawing a Rectangle on the Canvas

It should be noted that Power BI lets you decide the orientation of a slicer. For the dashboard in figure 4.7–7, we have opted for the default vertical orientation. Try your skills and see how a horizontal slicer would look. Figure 4.7–12 below shows how to change the orientation of the slicer.

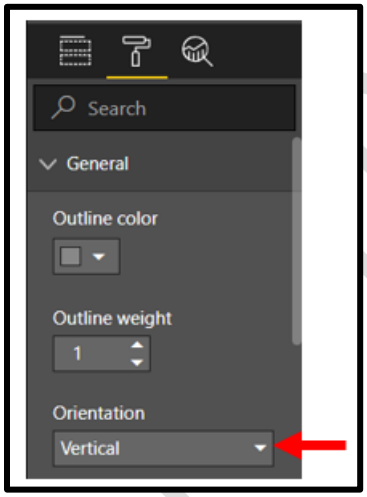


Figure 4.7–12
Select Slicer Orientation

Explore some of the other options—such as defining round corners for the rectangle—yourself!

4.8 MAP CHARTS

ENABLE COMPARISON BY GEOGRAPHIC LOCATION

Power BI supports three types of maps: bubble maps, filled maps, and ArcGIS maps. A bubble map represents specific geographic points. A filled map represents regions such as states. ArcGIS maps were more recently added and support the definition of sophisticated maps for advanced spatial analysis. We will limit our discussion here to bubble and filled maps.

Maps have several uses, including:

1. the display of relative proportions across locations using different colors and color shades;
2. as an interactive tool for geographical analysis.

Both uses will be illustrated in this section.

4.8.1 How to Create a Map

Let's create a new dashboard. Name your tab Maps, and add both a filled map and a bubble map.

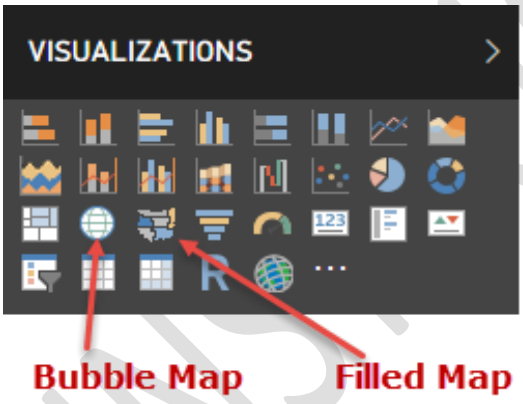


Figure 4.8–1
Creating Maps

Click on the "Filled map" icon first. Then click on the "Bubble map" icon.

Rearrange your canvas so that it looks like the following (figure 4.8–2).

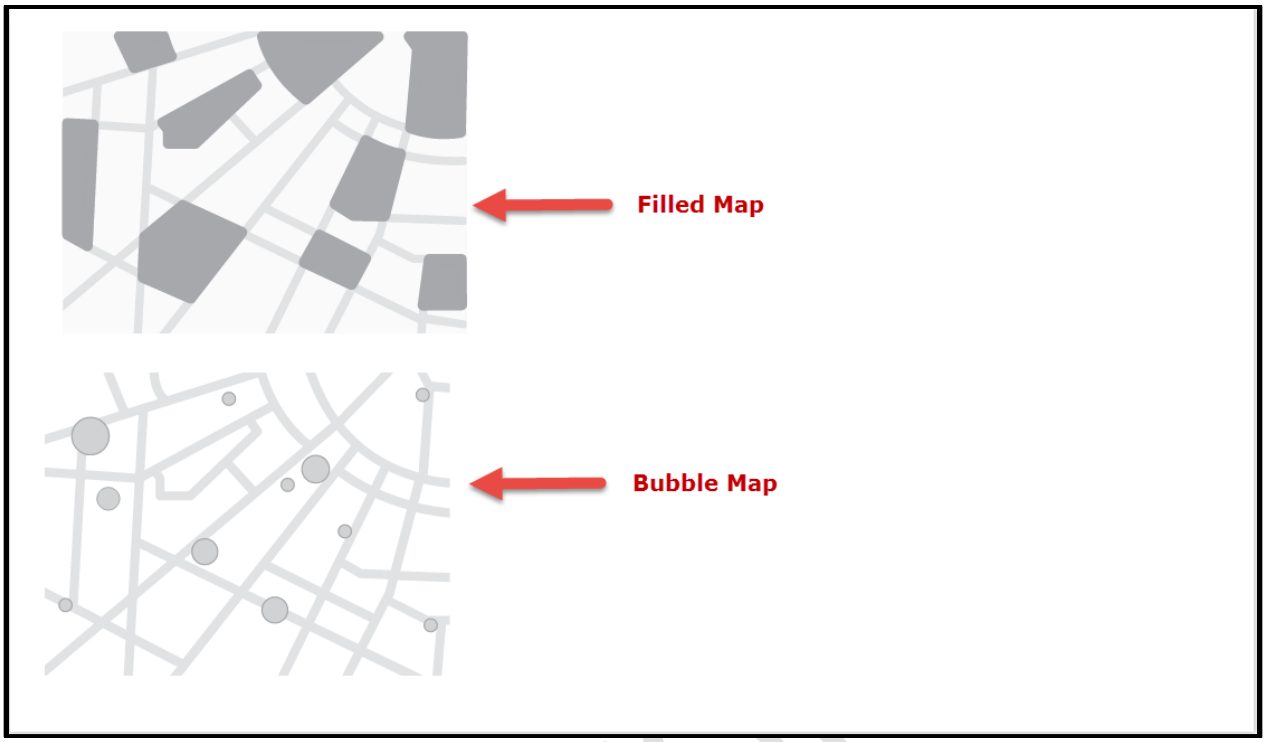


Figure 4.8–2
Empty Maps on the Canvas

4.8.2 Adding Data to Maps

The most important part of creating maps is having data available that will enable the accurate identification of geographic locations.[12] Power BI relies on Bing Map Services.[13] For this exercise, we have created two such fields: state (filled map) and address (bubble map). Let's start with the content definition of the "filled" map.

First, click the State field, which will go under Location in the Fields tab (Visualizations panel). This will enable geographic analysis at the state level.

[12] You will need an active Internet connection to do this.
[13] Bing Map Services is owned by the Microsoft Corporation.

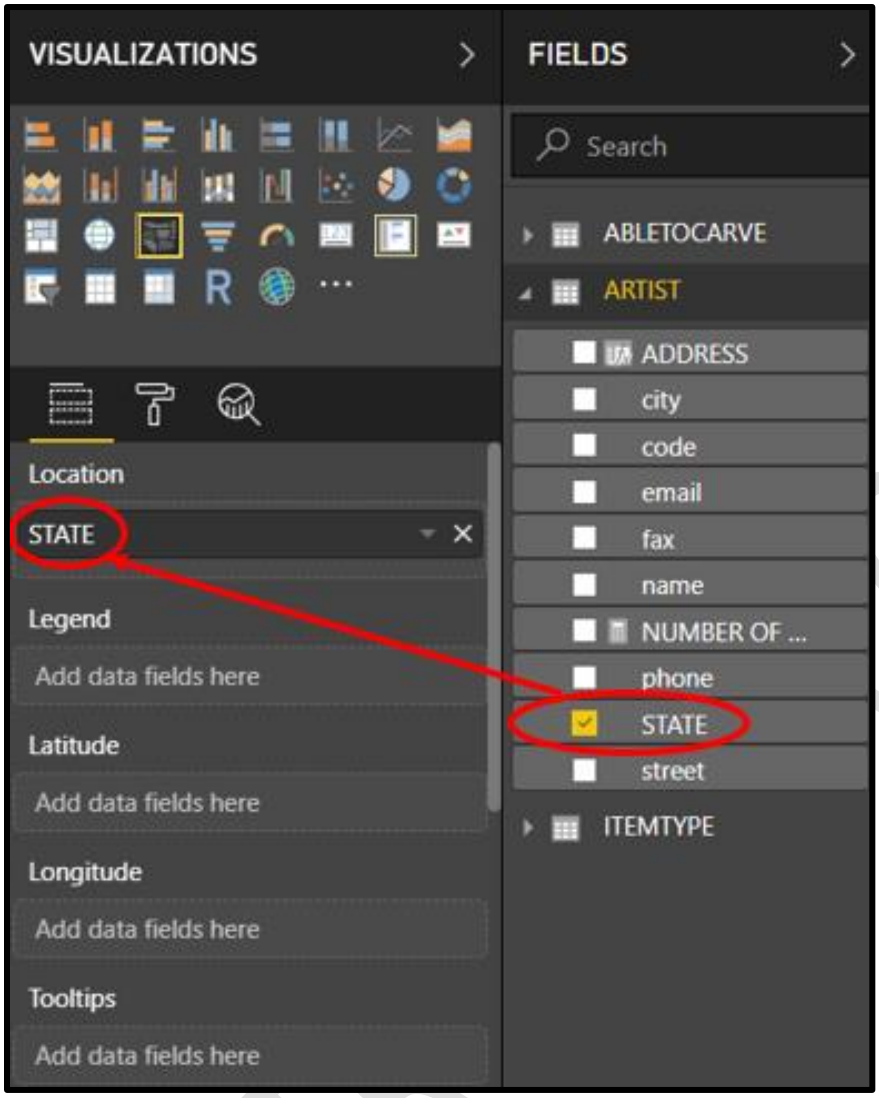


Figure 4.8–3
Content Definition: What Data Goes into a Filled Map

At this point, your filled map (upper-left corner of your canvas) should look like the one shown in figure 4.8–4 below. All states that have artists with whom KaDo works are shaded.

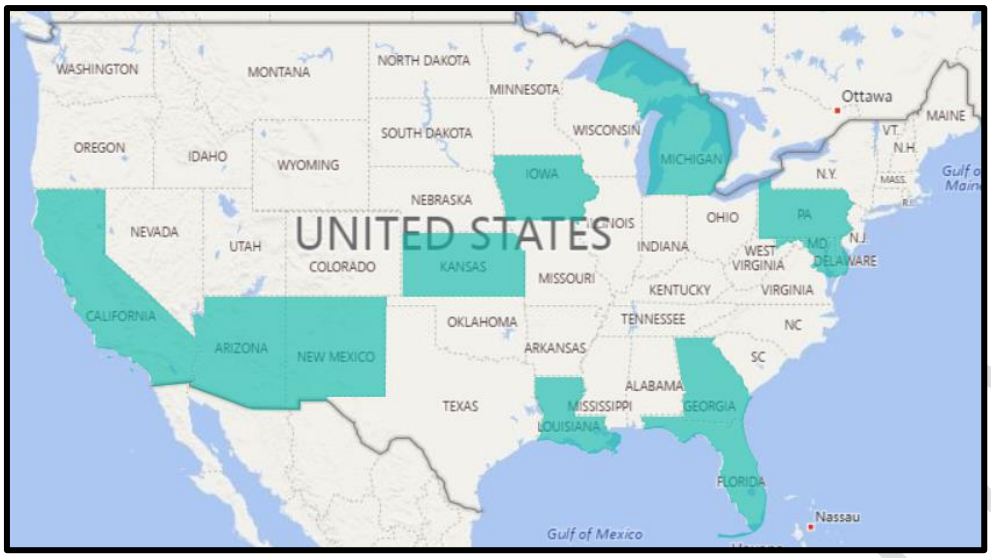


Figure 4.8–4
Filled Map

Next, let's define the content of the bubble map. As shown in figure 4.8–5, use the Address field in the Artist table to instantiate the bubble map.

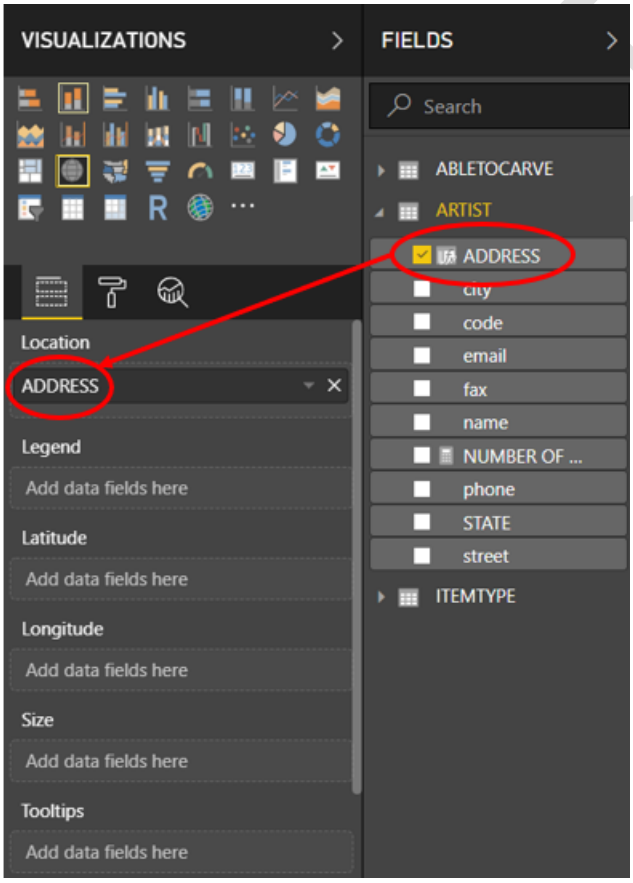


Figure 4.8–5
Content Definition; "Address" Field Used to Instantiate the Bubble Map

At this point, your canvas should look as follows (figure 4.8–6).

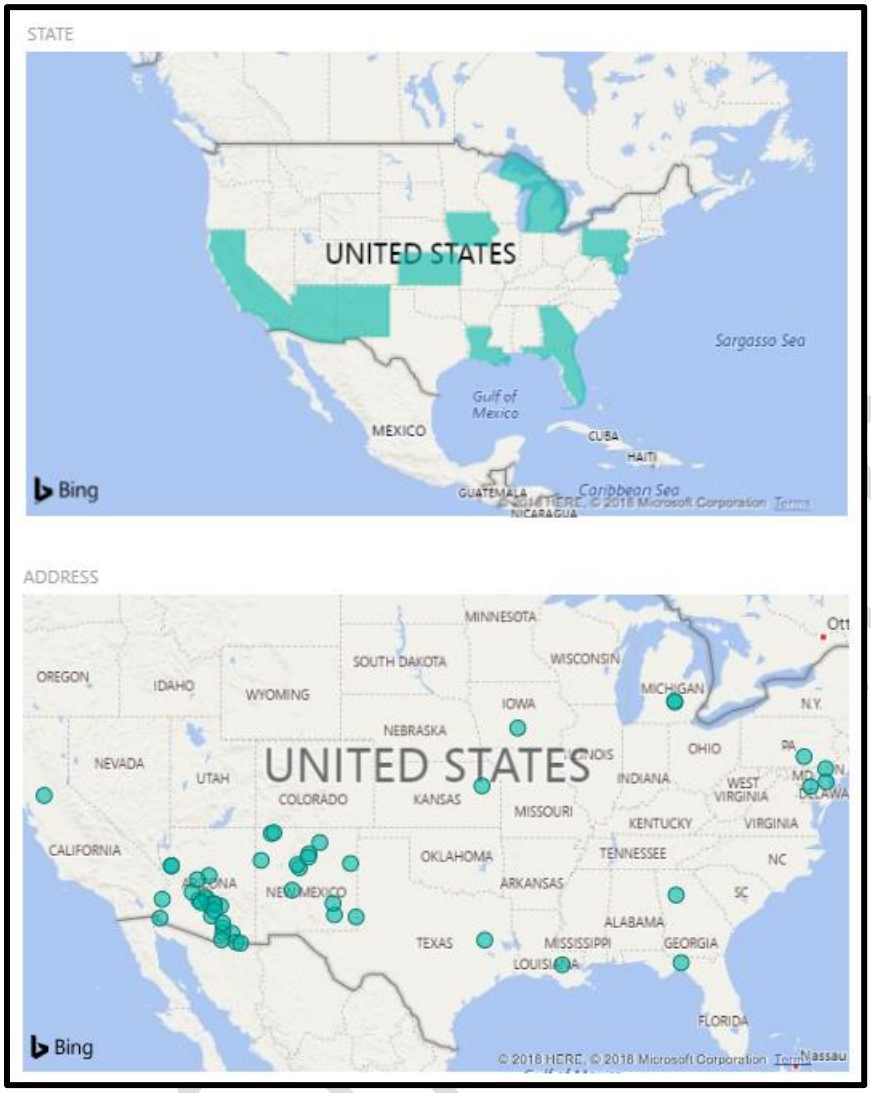


Figure 4.8–6. Canvas with a Filled Map and a Bubble Map

Explore the interactive nature of these maps. Click on a state, then on an address (bubble).

ASSIGNMENT 4.8-A1

Add two bar charts: one that shows the number of item types per artist and one that shows the number of artists by product (item) type. These are the two bar charts we created in section 4.6. Figure 4.8–8 shows how your canvas should look.

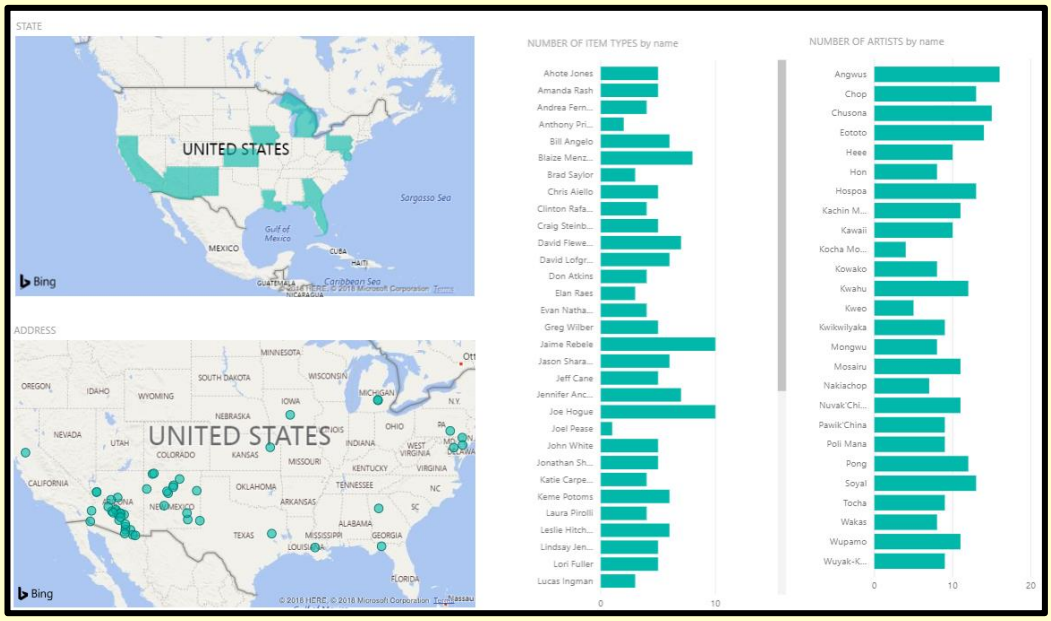


Figure 4.8–8
Canvas for assignment 4.8–A1

Use this interactive dashboard to answer questions such as:

- In what states is a specific product type (type of doll) made (carved)?
- Who makes (carves) a specific product type?
- In what state and city is a specific vendor (artist) located?
- Which vendors (artists) work in specific states?

4.8.3 Formatting Maps

Similar to other visualizations, maps come with a series of formatting options. Next, we will discuss two such options: color differentiation, which works for both filled and bubble maps, and bubble size, for bubble maps only.

Make the filled map the active visualization, click on "Focus mode" (Figure 4.8–7), go to the "Formatting Tab," and select "Data colors."

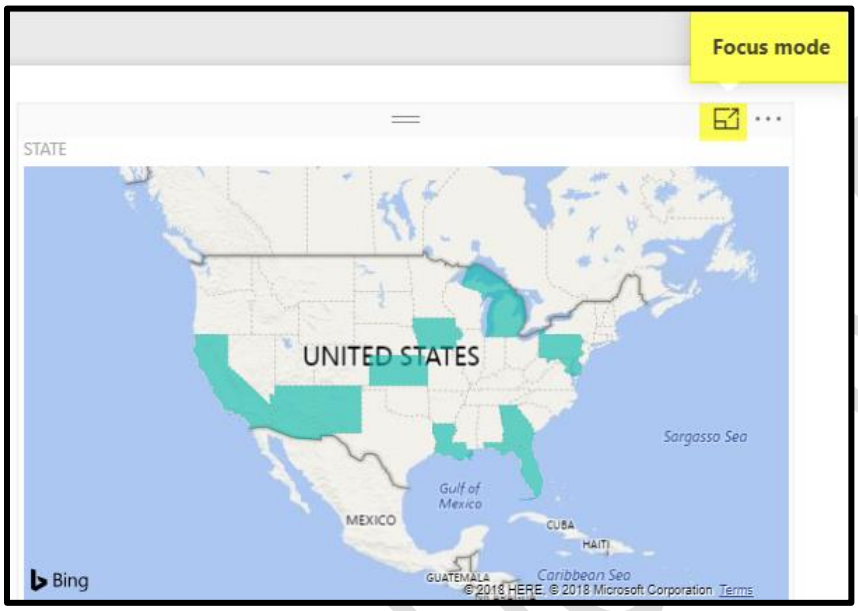


Figure 4.8–7. Switch to Focus mode (enlarge)

As shown in Figure 4.8–8 below, you can choose among three coloring options: "Default color," "Advanced controls," and "Show all."

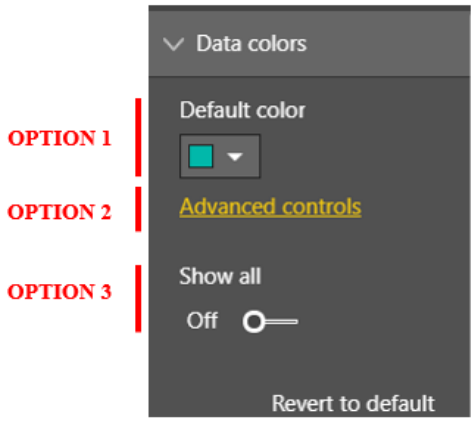


Figure 4.8–8. "Data colors" options

Currently, all states in which Kado does business are the same color, green. That color can be changed by using option 1: "Default color." Give it a try; select a different color!

The "Show all" option (Option 3 in Figure 4.8–8) allows you to define a specific color for each of the states in which Kado does business. Figure 4.8–9 below shows what happens when you move the slider from "Off" to "On." Try it out by selecting specific colors for the states.

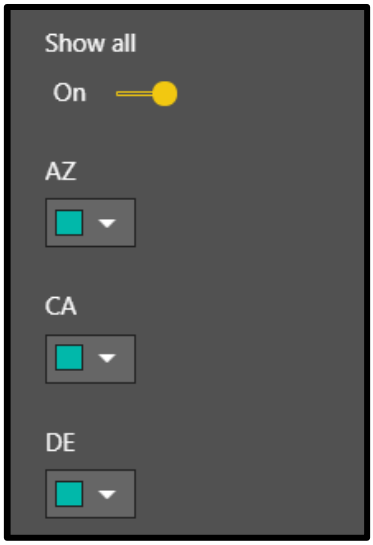


Figure 4.8–9. The "Show all" option

Finally, click on the "Advanced controls" link (Option 2 in Figure 4.8–8). The window shown in Figure 4.8–10 will appear. It provides three additional ways of coloring the filled map: "Color scale," "Rules," and "Field value." We'll discuss the first two in more detail below.

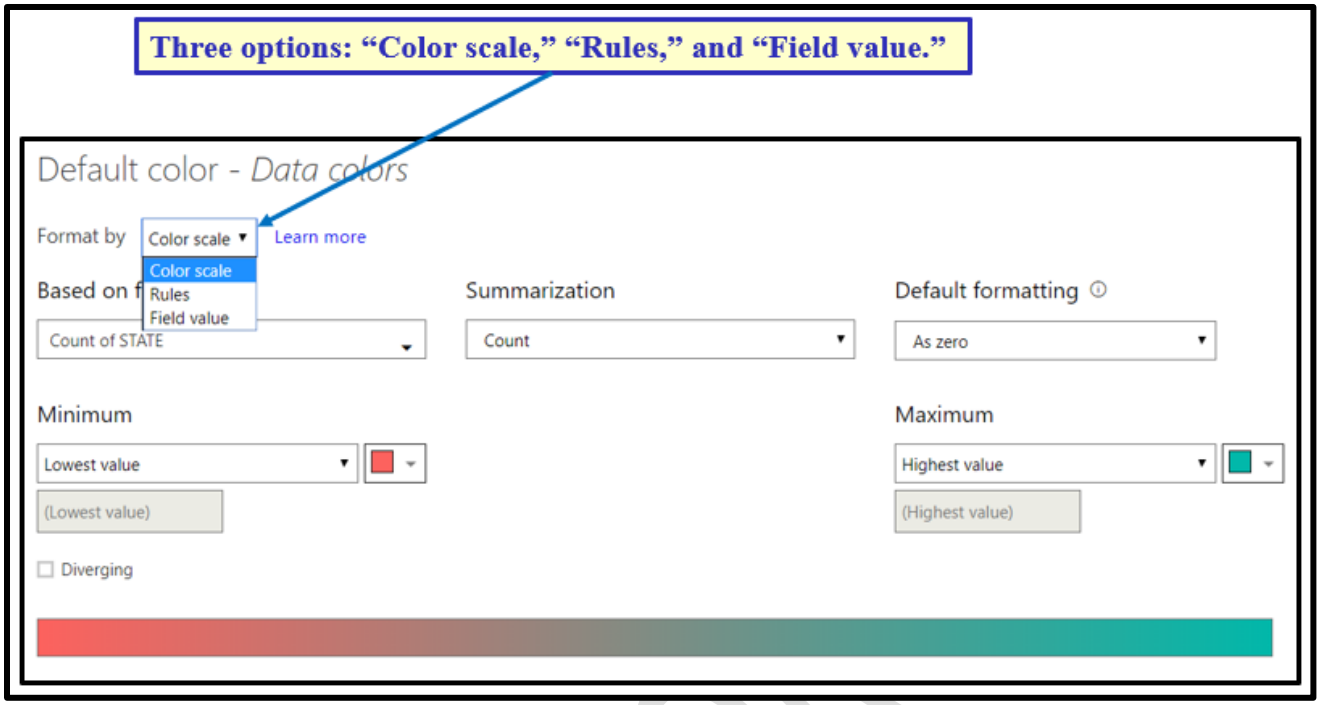


Figure 4.8–10. Advanced controls

For filled maps, the advanced controls are used to shade regions (states) based on the values of a specific field. The "Color scale" option shown in Figure 4.8–10 asks for several inputs to define such shading.

- First, specify what (numeric) field needs to be used for shading purposes. Select "NUMBER OF ARTISTS." For non-numeric fields, specify how they are summarized. To do this, select the SUMMARIZATION option.
- Second, specify which colors you would like to use for shading purposes and how to use them.
  - By default, the left color applies to the "Lowest value" and the color on the right applies to the "Highest value." The gradient bar shows how the color will gradually change between the minimum and maximum values.
  - A second option is to define specific values; see the "Number" option in Figure 4.8–11. All states that have a number of artists smaller than or equal to the "Minimum" number will be the color on the left. All states that have a number of artists that is larger than or equal to the "Maximum" number will be the color on the right. The specification of minimum and maximum threshold values is useful to eliminate outliers.
  - A third option is to mix up the VALUE and NUMBER specifications; e.g. placing "Lowest value" on the left and "Number" on the right.
- Third, "Diverging" allows you to specify an additional (middle) color, resulting in a wider range of colors to be used for shading purposes.

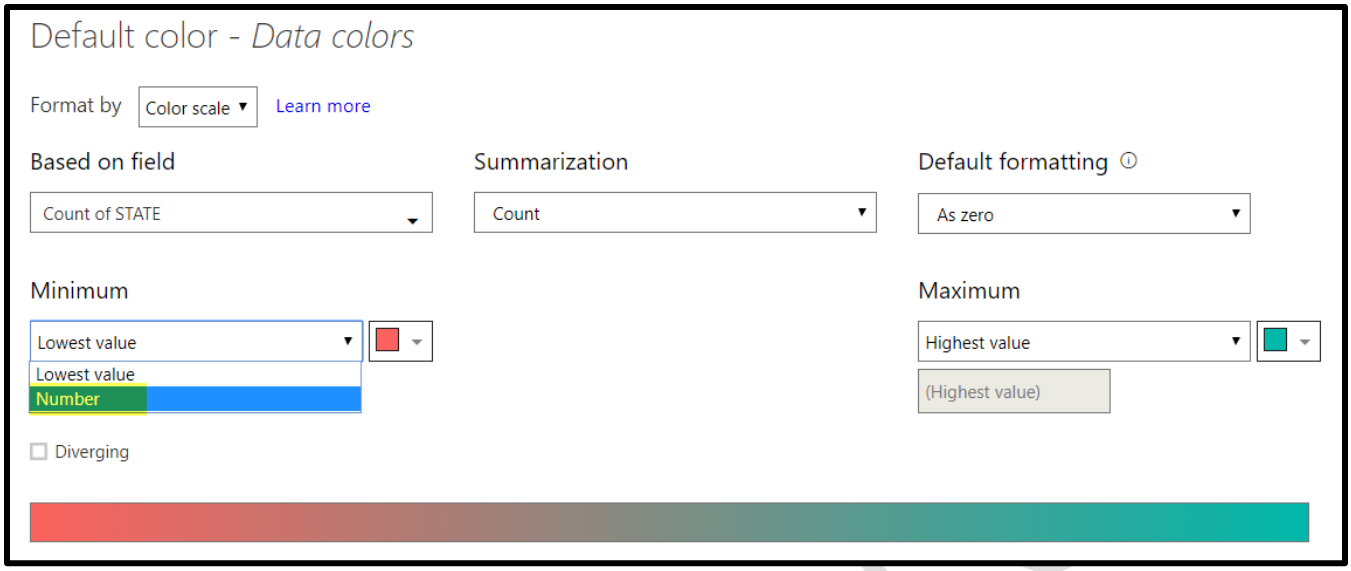


Figure 4.8–11. Selecting the "Number" option

The second coloring method, "Rules," provides you with a tool to link colors with specific number ranges. For example, the "Rules" definition in the top part of Figure 4.8–12 assigns the color orange to all states with one artist, the color blue to all states with two to ten artists, and the color purple to all states with more than ten artists. The result is shown in the lower part of Figure 4.8–12.

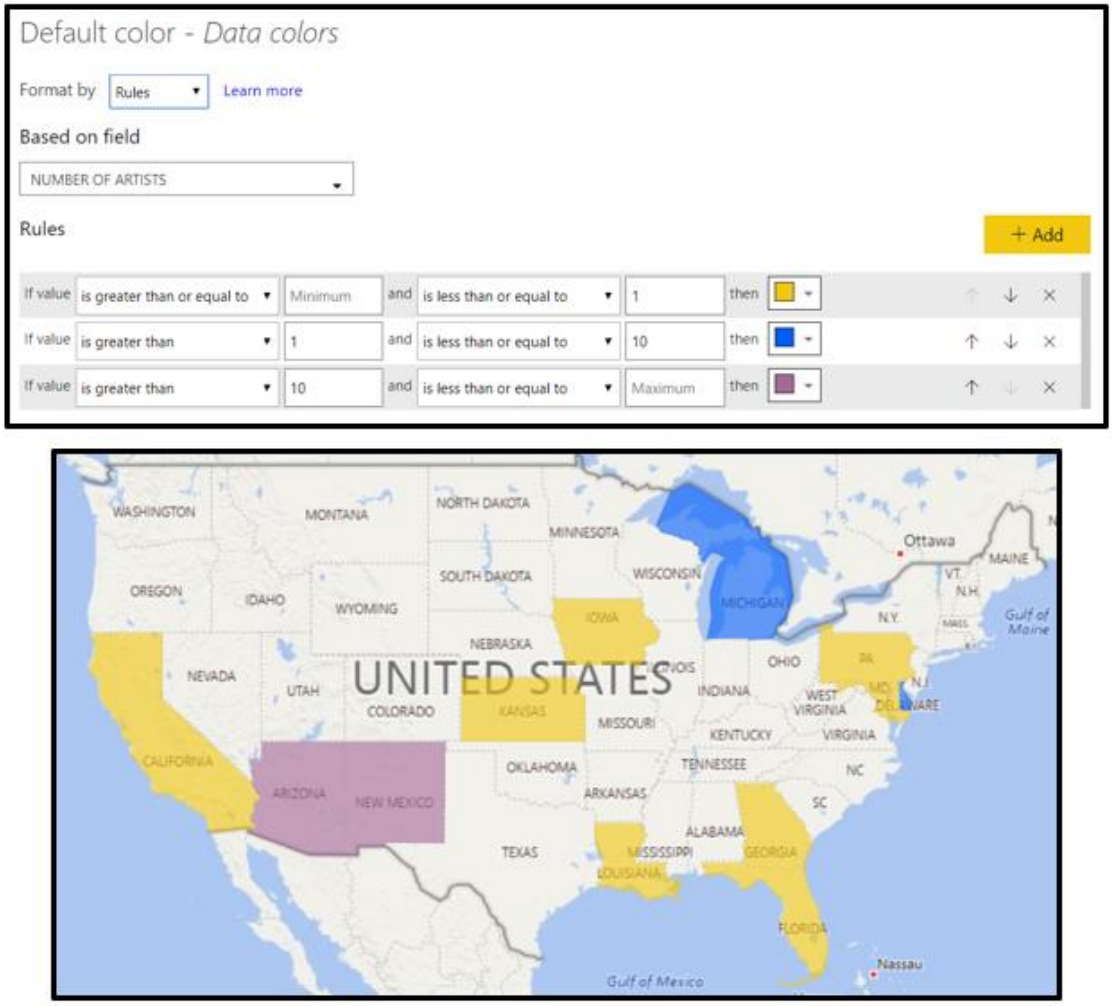


Figure 4.8–12. "Rules" definition

"Bubble maps" have similar formatting options, although there are some differences. In a bubble map, locations are represented as circles, or bubbles. You can change both the size and the color of the bubbles. As shown in Figure 4.8–13, the field that determines the size of the bubble is specified as part of the "content" definition (Fields tab). The higher the number of item types (products) an artist can create, the bigger the bubble.

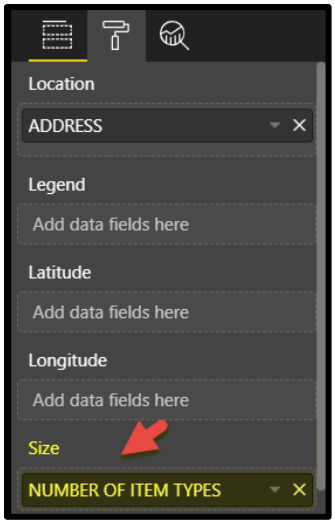


Figure 4.8–13. Size of bubble determined by the "NUMBER OF ITEM TYPES"

Color formatting for bubble maps is similar to the color formatting for filled maps discussed above. Figure 4.8–14 shows how you can use "Rules" to implement the following coloring:

RED: An artist who can craft up to five different item types (products).
BLUE: An artist who can craft six to ten different item types.
GREEN: An artist who can make more than ten different item types.

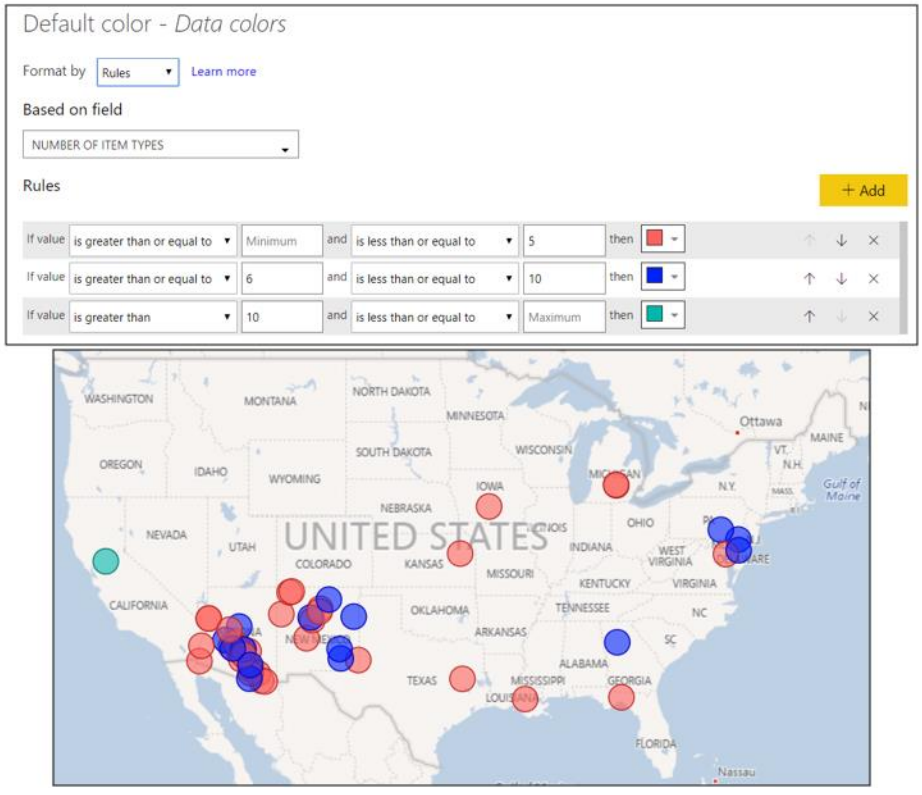


Figure 4.8–14. Coloring bubble maps using the "Rules" option

An option specific to bubble maps in the Format tab is the definition of the bubble size. The top part of Figure 4.8–15 shows how to increase the size of the bubbles. The lower part shows the resulting map for the Phoenix, Arizona area.

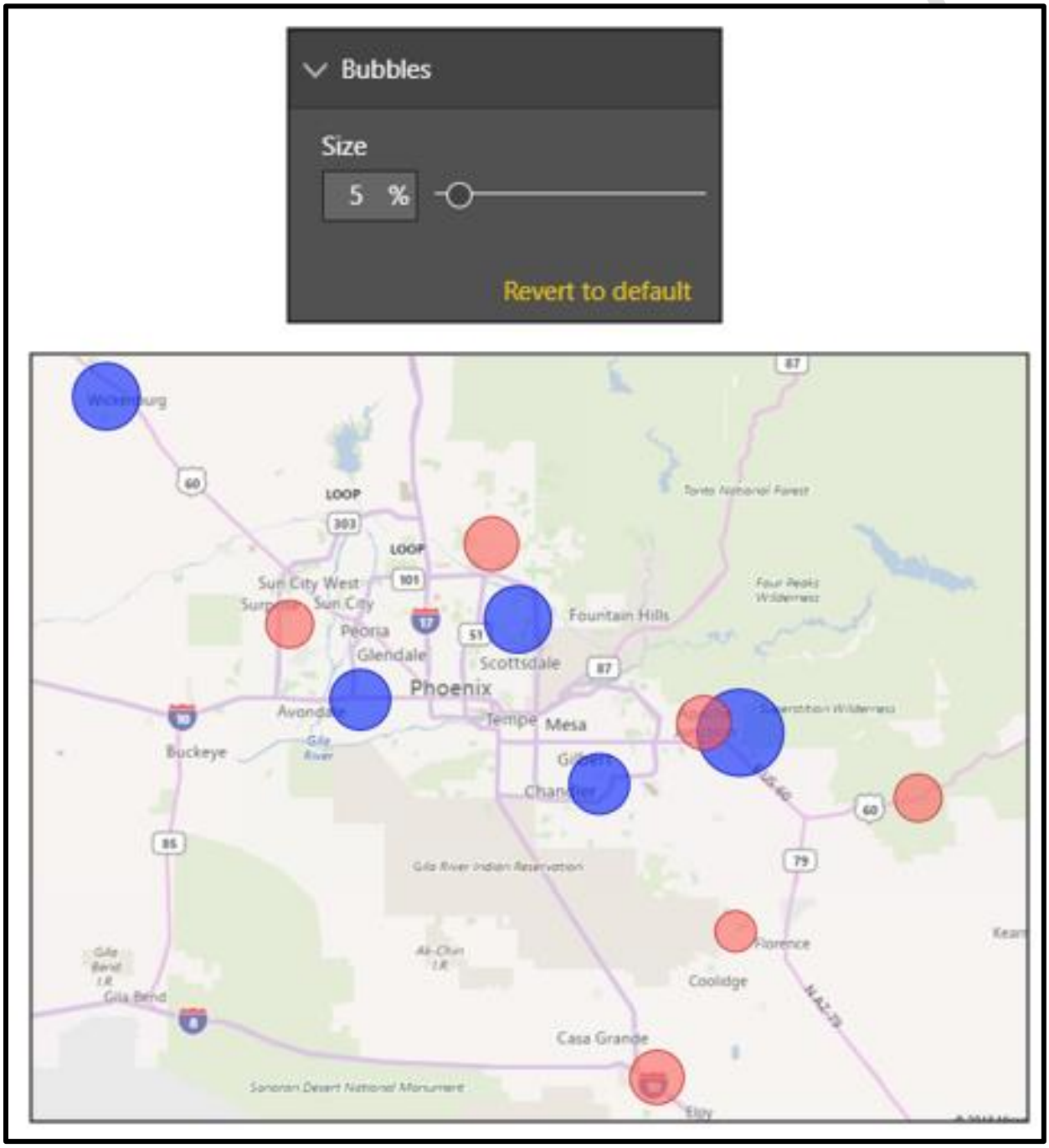


Figure 4.8–15. Increased bubble size

CHAPTER 5
INFORMATION MODELING

Learning Objectives

- ➔ Learn about the importance of the information modeling layer for analytics.
- ➔ Learn about the role of columns and measures and how to define them.
- ➔ Become acquainted with the Data Analysis Expressions (DAX) language.

In the previous chapter, you learned how to build powerful interactive dashboards with little effort. The ability to build these dashboards, however, depends on the information you have available for analysis. Defining such information in the "information modeling layer" using Power BI's Data Analysis eXpressions (DAX) language is the subject of this section.[14] Figure 5.0–1 reiterates where "information model building" or "information modeling" is situated in the data process chain.

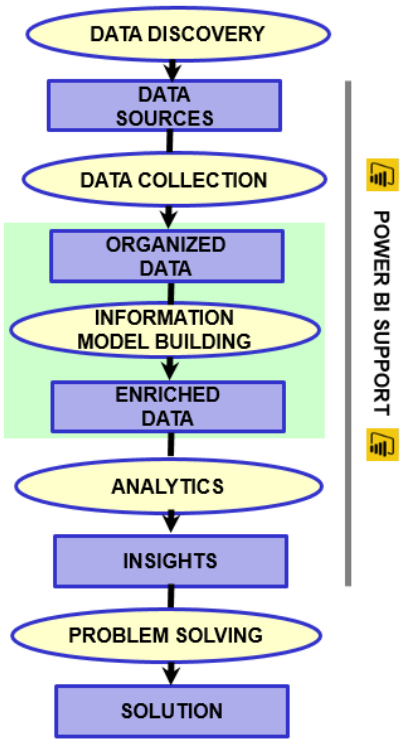


Figure 5.0–1
Information Modeling in the Data Process Chain

Information modeling aims to define information for analytical purposes, starting from the raw data collected.

[14] Excel and DAX functions have much overlap between them—about 70% of Excel functions are available in DAX as well—although DAX adds several powerful new functions, such as Calculate(), that enable advanced calculations and querying.

5.1 WHERE TO FIND THE DATA SETS

We will use two data sets in this section:

1) **ABILITY.PBIX**
   This is the data set that we used in chapter 4. We will use it again in this chapter to discuss the data set's information model, which consists of one column definition and two measure definitions, and to introduce the DAX syntax. You can use the ABILITY.PBIX data set from the previous chapter. We will only look at the data set's information model; we won't add anything.

2) **ORDERS.PBIX**
   We will use this data set to develop a more advanced information model that starts from specific information needs. Our focus will be on the development of measures. You can find the ORDERS.PBIX data set in your DropBox folder.

5.2 UNDERSTANDING THE DATA

5.2.1 The Problems to Be Solved

Because we discussed the problems to be addressed by the ABILITY.PBIX data set in the previous chapter, we will now focus on ORDERS.PBIX. Three main questions will be answered:

1. When did orders take place, and how many?
2. Whom did we order from?
3. What did we order?

5.2.2 Exploring the Data Set and Its Structure

Since we discussed the structure of the ABILITY.PBIX data set in the previous chapter, in this chapter we will limit ourselves to a discussion of the structure of the ORDERS.PBIX data set. This data set's data model is shown in figure 5.2–1 below.[15] Artist and ItemType are the same tables as those used with the same names in ABILITY.PBIX. The new tables are PORDER, PORDERLINE, and ITEMCATEGORY.

[15] We have deliberately opted not to use the recommended "star schema" here. For example, we will discuss the implementation and use of a separate date table in a more advanced module.

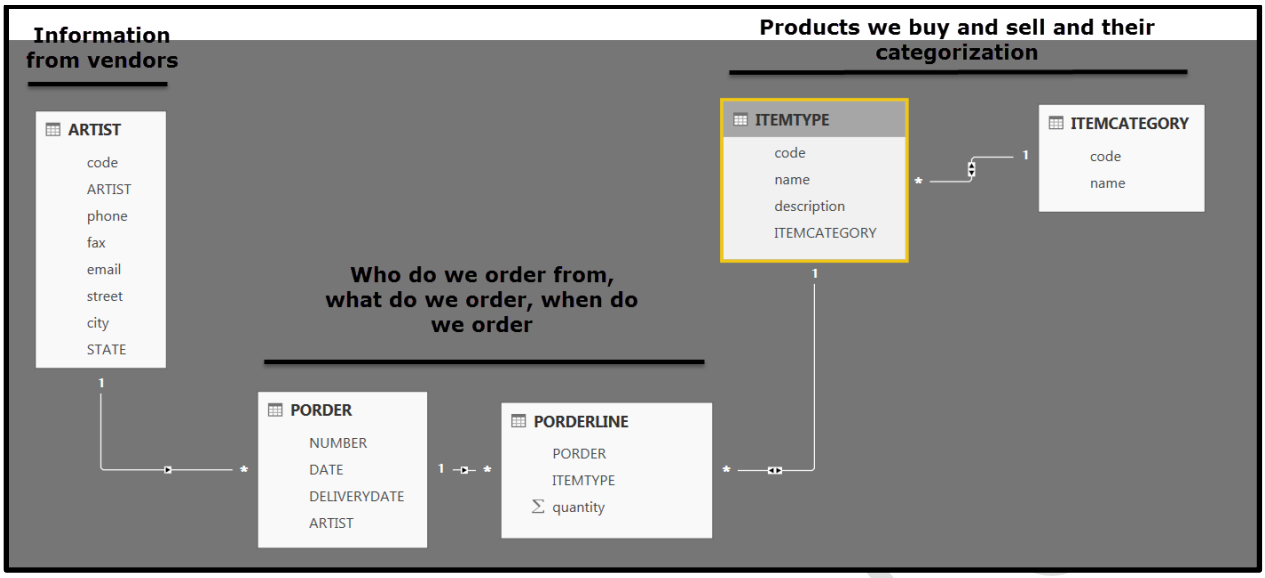


Figure 5.2–1
Data Model for ORDERS.PBIX

Together, POrder[16] and POrderLine describe from whom KaDo ordered, when, what, and how many. ItemCategory further categorizes KaDo's products (kachina dolls). We will discuss these three tables in more detail in the next section.

5.2.3 The Data Set

The ARTIST and ITEMTYPE tables contain the same information as the corresponding tables in the Ability.pbix dataset, but a few subtle differences should be noted. For example, in the ORDERS.pbix Artist table, the names of the artists are stored in the Artist field instead of the Name field, as was the case in the ABILITY.pbix data set. It is important to analyze all fields and to make sure that you understand what data are stored in the fields before you start with information modeling and/or building dashboards.

Next, let's discuss each of the three new tables in more detail: POrder, POrderline, and ItemCategory.

POrder

The POrder table describes the purchase orders that KaDo has placed and its structure is shown in figure 5.2–2. Information recorded about purchase orders includes a number that uniquely identifies an order (number), the date the order was placed (date), when the goods should be delivered (deliverydate),

[16] The "P" in POrder and POrderLine refers to "Purchase."

and from whom the items were ordered (artist). Figure 5.2–3 below shows the (data) description for one specific order (i.e., an instance).

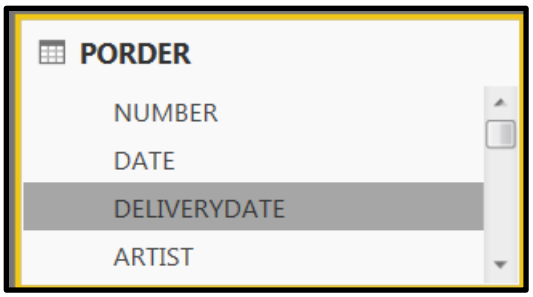


Figure 5.2–2
Structure of the POrder Table

NUMBER	DATE	DELIVERYDATE	ARTIST
1	2/24/2017	4/27/2017	AMRAS

Figure 5.2–3
Description of a Specific Order (Instance)

POrderLine

Figure 5.2–4 shows the structure of the POrderLine table. This table links orders (PORDER) with item types (ITEMTYPE) and describes *what* products have been ordered. It further describes *how many* (quantity) items have been ordered. Figure 5.2–5 shows one specific instance of POrderLine.

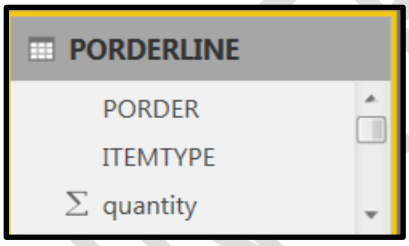


Figure 5.2–4
Structure of the POrderLine Table

PORDER	ITEMTYPE	quantity
1	CHU	3

Figure 5.2–5
Description of a Specific Order Line (Instance)

ItemCategory

Figure 5.2–6 shows the structure of the ItemCategory table. This table describes broader product categories using a code for unique identification and the name of the category ("description"). Figure 5.2–7 shows one specific instance of ItemCategory.

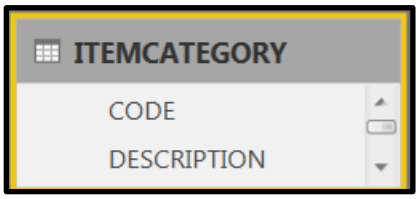


Figure 5.2–6
Structure of the ItemCategory Table

CODE	DESCRIPTION
ANI	animal

Figure 5.2–7
Data for a Specific Item Category

5.3 TOOLS FOR DEFINING THE INFORMATION MODEL: COLUMNS AND MEASURES

Information modeling in Power BI is done by means of two mechanisms: **columns** and **measures**.

- A **column** is an integral part of a table, and a value is calculated for each row in that table.
- **Measures**, in contrast, are not an integral part of a table: they define aggregates that can be used as part of dashboards/visualizations and thus for analytical purposes.

These mechanisms are defined using the "New column" and "New measure" commands, which can be accessed from a number of different menus in Power BI. For example, figure 5.3–1 shows where to find these commands in the Data View Home ribbon. The actual definition of these commands is discussed in more detail below.

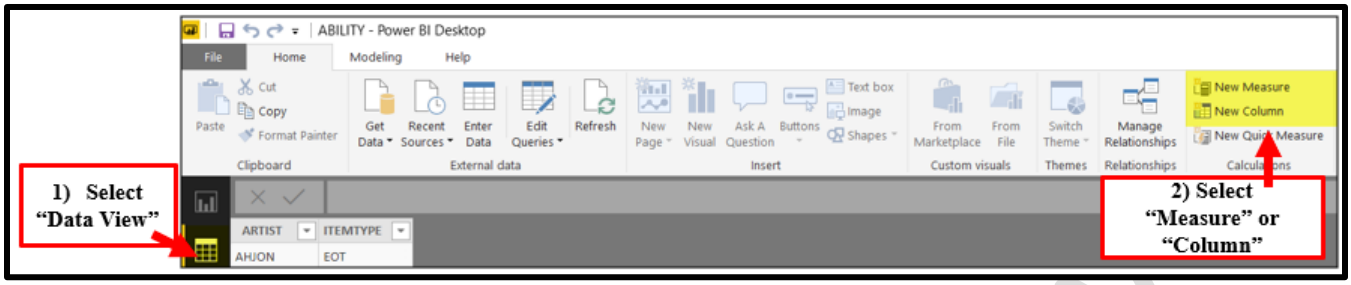


Figure 5.3–1
Defining Columns and Measures

1) First select the Data view.
2) Then select either "New measure" or "New column" (see ribbon).

An area will pop up in which you can define a formula using the DAX language.

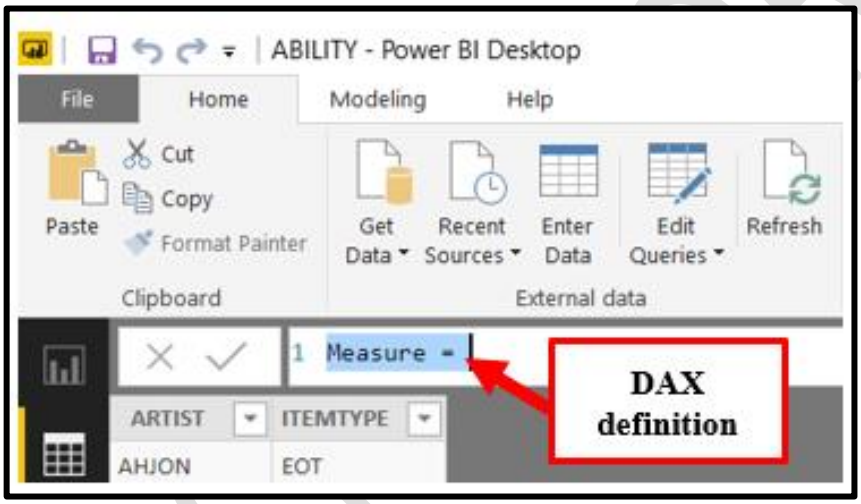


Figure 5.3–2
Entering a DAX Formula

Side Note

When you create a new "column" or "measure," that item will be added to the active table (i.e., the table that is currently selected).

5.4 UNDERSTANDING AN INFORMATION MODEL

Side Note

The purpose of this section is to **understand** an existing information model. We will use the model defined as part of the ABILITY.PBIX data set for this purpose. In the next section, you will learn how to **develop** an information model.

An information model enriches raw data—that is, it calculates information that is useful for analysis. For the data set used in chapter 4, ABILITY.PBIX, the information model consists of three elements, which we extensively used in our dashboards in that chapter:

- One column definition: "Address"
- Two measure definitions: "Number of item types" and "Number of artists."

The shaded areas (yellow) in figure 5.4–1 show how the information model is part of the data model.[17]

 indicates a column definition.
 indicates a measure definition.

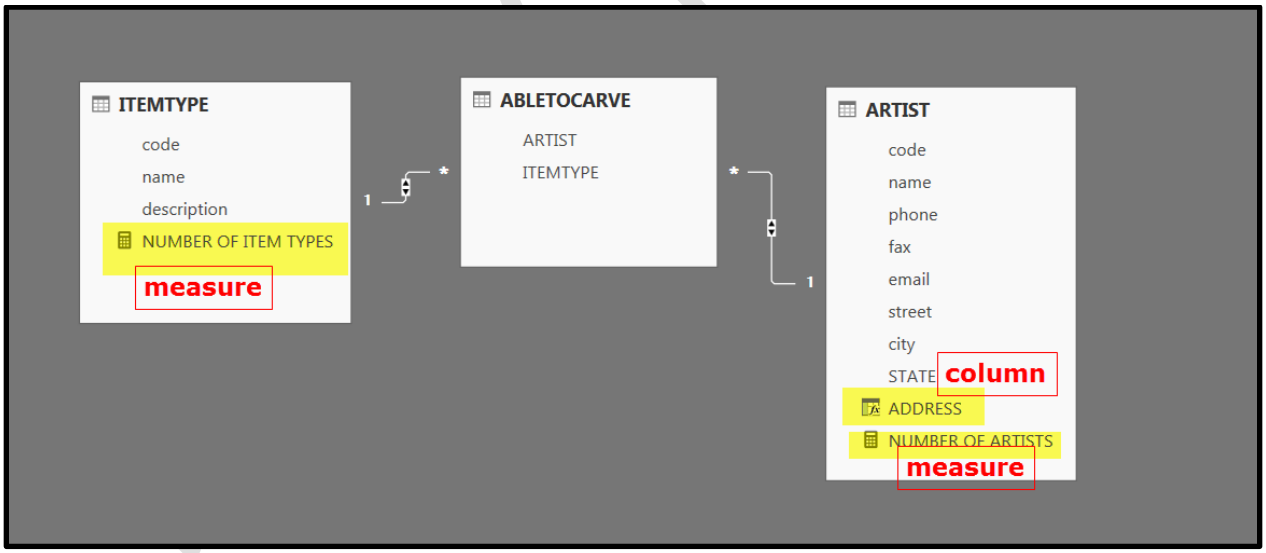


Figure 5.4–1
Information Model Definition as Part of the ABILITY.PBIX Data Model

[17] Although all columns and measures shown in Figure 5.4–1 are capitalized to make them stand out, the DAX language is not case sensitive.

Both columns and measures are defined using the DAX language; their definitions are discussed in more detail below. To see the actual definitions for a column or a measure in Power BI, just click on it.

ADDRESS

PURPOSE
To have accurate geographical information that can be used in map charts (bubble maps).

APPEARANCE
As shown in figure 5.4–2 below, the address appears as an additional column in the Artist table.

DAX DEFINITION

```
ADDRESS = ARTIST[street] & "," & ARTIST[city] & "," & ARTIST[STATE]
```

DAX SYNTAX

ADDRESS	Name of Column
ARTIST[street] ARTIST[city] ARTISTS[STATE]	Fields from the Artist table
&	DAX operator: concatenation

DAX FUNCTIONS AND OPERATORS

&	The concatenation operator joins two or more text strings into one string. The same operator is also available in Excel. Power BI also has a CONCATENATE() function, which allows you to join two strings; i.e., it has only two arguments. In the example above, & is used to create one address field (column), starting from the values of three different fields—street, city, and state—and separating them by commas.

code	ARTIST	phone	fax	email	street	city	STATE	ADDRESS
AHJON	Ahote Jones	480-982-5525	480-982-5526	ajones@yahoo.com	555 Idaho Rd.	Apache Junction	AZ	555 Idaho Rd.,Apache Junction,AZ
AMRAS	Amanda Rash	928-927-1470		amanda@hotmail.com	48865 Riviera Pl	Ehrenberg	AZ	48865 Riviera Pl,Ehrenberg,AZ
ANFER	Andrea Fernandez	520-399-0134	520-399-0139	fernandeza@hotmail.com	603 W. Camino Sorpresa	Sahuarita	AZ	603 W. Camino Sorpresa,Sahuarita,AZ
ANPRI	Anthony Pricci	520-623-4677	520-623-1111		1001 E 17th St.	Tuscon	AZ	1001 E 17th St.,Tuscon,AZ
BIANG	Bill Angelo	404-656-1230		billa@comcast.com	311 Delmont Dr NE	Atlanta	GA	311 Delmont Dr NE,Atlanta,GA
BLMEN	Blaize Menzoni	520-806-7970		bmenz@hotmail.com	65 Rosalies Ct	Tubac	AZ	65 Rosalies Ct,Tubac,AZ

Figure 5.4–2
The "Address" Column as Part of the Artist Table

NUMBER OF ITEM TYPES

PURPOSE
Determines the number of items KaDo has in its product line: how many different types of products (kachina dolls) does the company buy and offer? We used this measure extensively for analysis purposes in chapter 4.

APPEARANCE
Measures are part of the data model but are not part of a table. Their purpose is to be used as part of the visualizations for analysis purposes.

DAX DEFINITION

```
NUMBER OF ITEM TYPES = COUNTA(ITEMTYPE[code])
```

DAX SYNTAX

NUMBER OF ITEM TYPES	Name of measure
COUNTA()	DAX function that counts rows in a table
ITEMTYPE([code]):	Field used by the COUNTA() function

DAX FUNCTIONS AND OPERATORS

COUNTA()	The COUNTA() function counts the number of non-empty cells in a column. It counts every non-blank row independently of the nature of that row's values, including number, text, and date, among others.

NUMBER OF ARTISTS

PURPOSE
Determines how many different artists KaDo works with (i.e., buys items from).

DAX DEFINITION

```
NUMBER OF ARTISTS = COUNTA(ARTIST[code])
```

APPEARANCE
DAX SYNTAX
DAX FUNCTIONS AND OPERATORS

Similar to the definition of the "Number of item types" measure above.

5.5 DEVELOPING AN INFORMATION MODEL

In this section, we will develop an information model for the ORDERS.PBIX data set (the data enrichment step). The framework presented in table 5.5.1 is a useful tool for starting the development process. Moving from left to right, the table has three sections: analysis, measure, and subject of analysis.

ANALYSIS
What type of analysis will we use the data set for?
We'll start with the general categories—comparison and trend analysis (column 1). Where applicable, we'll identify the more specific type of analysis we would like to perform, such as rank (column 2).

What information will we use for this analysis?
What information we will use to rank, to determine relative shares, and other factors (column 3)?

In essence, we need to determine the types of analysis we will conduct (category and type) and the information we need to conduct the analysis (the base). Analysis answers the *what* question: What kind of analysis would we like to do using what information?

MEASURE
We'll formally define our information model at this point: What measures is our information model composed of?

We added a measure number (#) (column 4) for reference purposes, which will make our discussion below easier. Column 5 shows the names of the actual measures that will be part of the information model; these measures are the tools to be used to perform the analysis (columns 1–3).

Measures help to answer the *how* question: How will we implement the analysis using a tool like Power BI?

SUBJECT OF ANALYSIS

What objects do the analysis and measure apply to?

The third section in table 5.5–1 (columns 6–10) lists all objects, represented as tables in Power BI. Shading is used to indicate the objects for which an analysis—and thus the measures developed for the analysis—are relevant.

Table 5.5–1

Measure/Analysis Framework

ANALYSIS			MEASURE		SUBJECT OF ANALYSIS				
CATEGORY	TYPE	BASE	#	NAME	ARTIST	ITEM TYPE	ITEM CATEGORY	PORDER	PORDERLINE
COMPARISON		Number of Orders	1	Number Of Orders					
COMPARISON		Quantity Ordered	2	Quantity Ordered					
COMPARISON	RANK	Quantity Ordered	3.1	Artist Rank					
			3.2	Item Type Rank					
			3.3	Item Category Rank					
COMPARISON	% (RELATIVE SHARE)	Quantity Ordered	4	Share					
COMPARISON	COMPLEXITY	Average Order Size	5	Average Quantity Ordered					
TREND	GROWTH	Quantity Ordered	6	Growth					

Side Note

The following are a few observations resulting from the specifications in table 5.5–1.

- The specification of measures is driven by analysis requirements, and not the other way around.
- The same information (base) can be used for different types of analysis: for example, "Quantity ordered."
- The same measure can be used to analyze more than one object and thus can have multiple applications. For example, comparisons based on quantity ordered can be done for artists, item types, and item categories.
- Some measures are applicable to a number of objects but require specific definitions for each of them: for example, artist rank (measure #3.1), item type rank (measure #3.2), and item category rank (measure #3.3).
- The framework can be used as a documentation tool: Why was a measure developed?

Next, we will discuss the definitions of the six measures in more detail and introduce a few additional DAX functions.

Measure #1
Number of Orders

PURPOSE
Determines the total number of orders that KaDo has placed thus far. (To be used for comparison.)

DAX DEFINITION

TABLE: PORDER
NUMBER OF ORDERS = COUNT(PORDER[NUMBER])

DAX SYNTAX

NUMBER OF ORDERS	Name of measure
COUNT()	DAX function that counts the different values (instances) in a column
PORDER[NUMBER]	Field from the POrder table used by the COUNT() function

DAX FUNCTIONS AND OPERATORS

COUNT()	The COUNT() function is used for numeric fields and counts every non-empty cell in such a field.

USES
Table 5.5–1 shows that the "Number of orders" measure can be applied to three different objects: Artist, Item type, and Item category. The dashboard in figure 5.5–1 further illustrates this "one measure, multiple uses" principle.

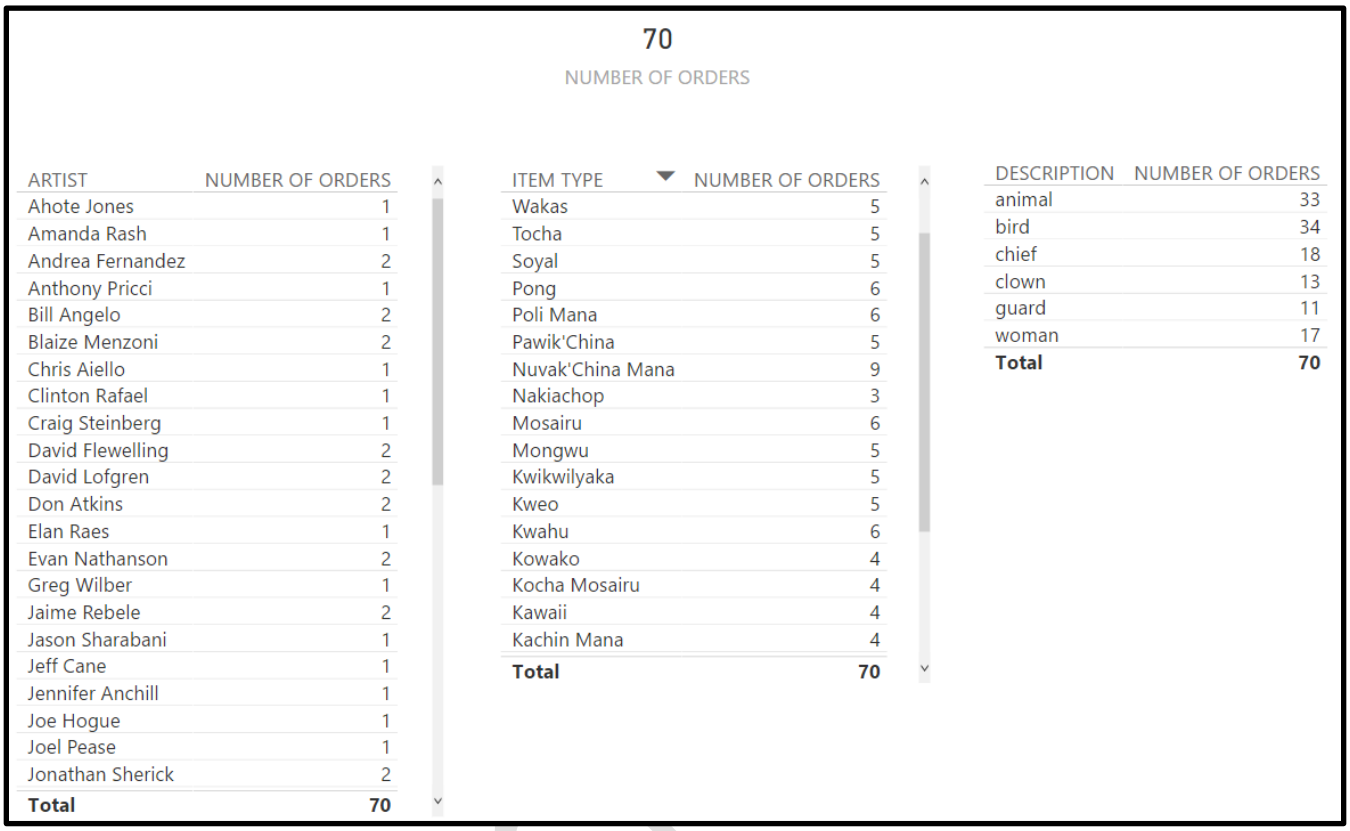


Figure 5.5–1
Illustration of the "One Measure, Multiple Uses" Principle

- A card is used at the top of the dashboard to show the total number of orders that have been placed thus far. In this case, the DAX formula has been applied without any filters.
- The table on the left is used to determine the number of orders per artist. In this case, the DAX formula is applied to each artist. The definition for the table in the Fields button looks as follows (figure 5.5–2):

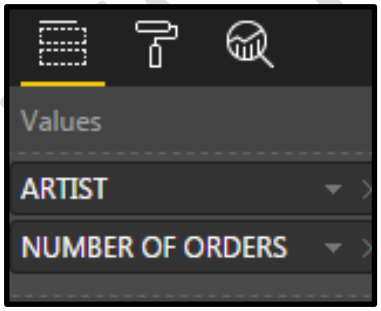


Figure 5.5–2
Definition of the "Number of Orders per Artist" Table

- The table in the middle of figure 5.5–1 shows the number of orders per item type; the definition of this table looks as follows (figure 5.5–3):

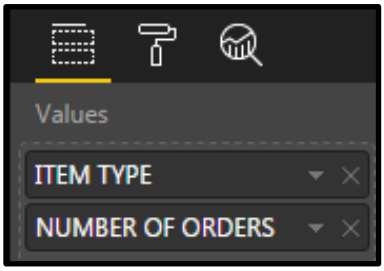


Figure 5.5–3
Definition of the "Number of Orders per Item Type" Table

- The table on the right of figure 5.5–1 shows the number of orders per item category; the definition of this table looks as follow (figure 5.5–4):

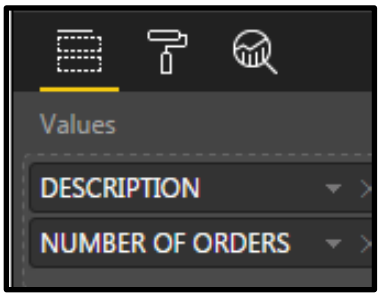


Figure 5.5–4
Definition of the "Number of Orders per Item Category" Table

Measure #2
Quantity Ordered

PURPOSE
Determines the total quantity of items that have been ordered. (To be used for comparison.)

DAX DEFINITION

TABLE: PORDERLINE
QUANTITY ORDERED = SUM(PORDERLINE[quantity])

DAX SYNTAX

QUANTITY ORDERED	Name of measure
SUM()	DAX function that sums all values in a column
PORDERLINE[quantity]	Field from the POrderLine table used by the SUM() function

DAX FUNCTIONS AND OPERATORS

SUM()	The SUM() function sums all the values in a column with a numeric data type

USES
Again, as indicated in table 5.5–1 and shown in figure 5.5–5,[18] this measure can be applied to different objects: Artist, Item type, and Item category.

[18] We have done additional formatting for the dashboard shown in figure 5.5–5.

Side Note

For comparison purposes, we put "Quantity ordered" in descending order for all three tables. You can do this by clicking on the down arrow next to the table column in your canvas. Figure 5.5–6 below shows how this can be done for the "Quantity ordered" column in the "Quantity ordered per artist" table.

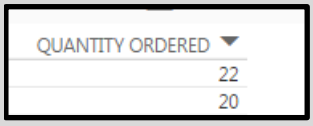
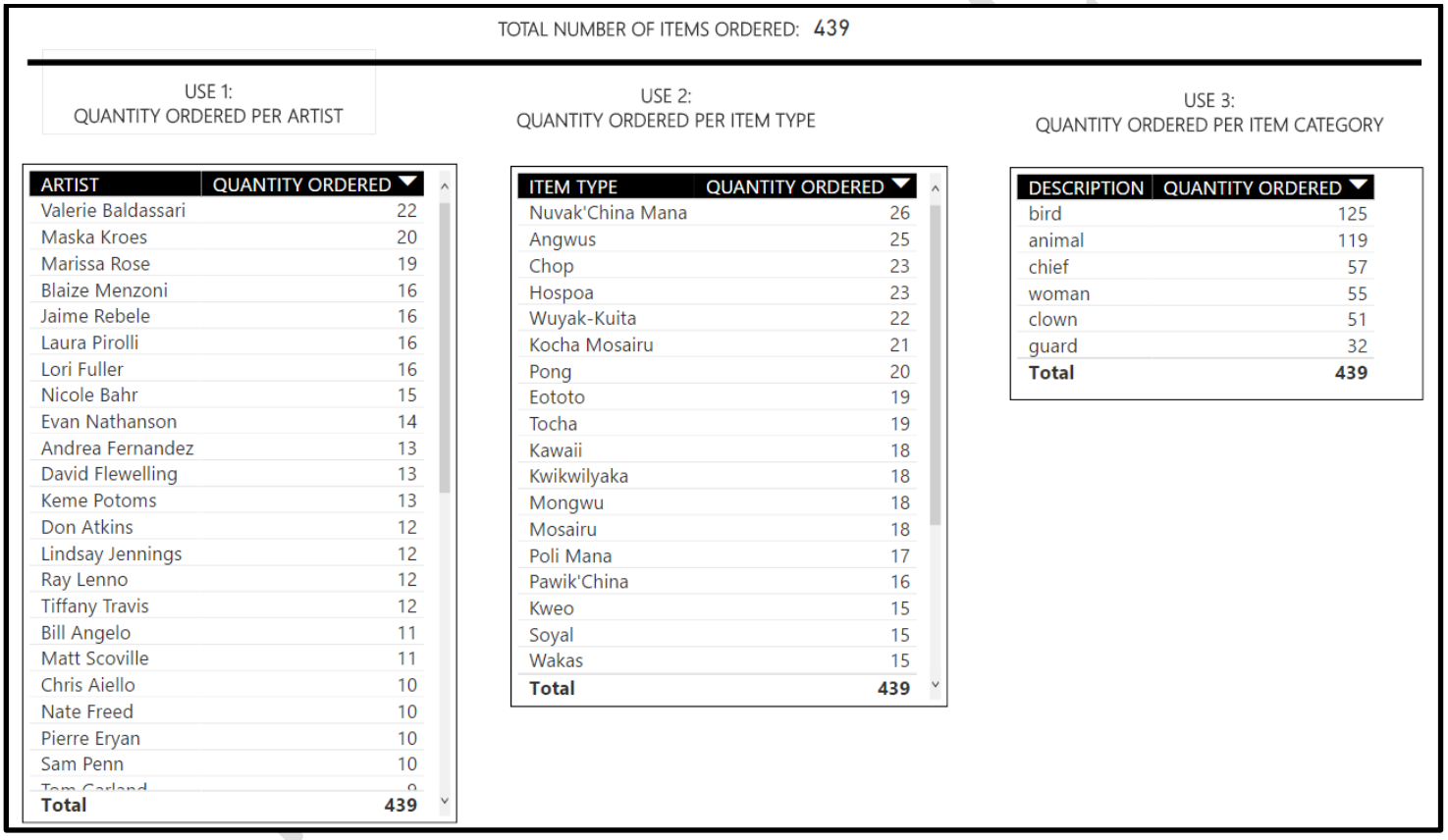


Figure 5.5–6
Selecting "Descending" Order



ARTIST	QUANTITY ORDERED
Valerie Baldassari	22
Maska Kroes	20
Marissa Rose	19
Blaize Menzoni	16
Jaime Rebele	16
Laura Pirolli	16
Lori Fuller	16
Nicole Bahr	15
Evan Nathanson	14
Andrea Fernandez	13
David Flewelling	13
Keme Potoms	13
Don Atkins	12
Lindsay Jennings	12
Ray Lenno	12
Tiffany Travis	12
Bill Angelo	11
Matt Scoville	11
Chris Aiello	10
Nate Freed	10
Pierre Eryan	10
Sam Penn	10
Tom Garland	9
Total	**439**

ITEM TYPE	QUANTITY ORDERED
Nuvak'China Mana	26
Angwus	25
Chop	23
Hospoa	23
Wuyak-Kuita	22
Kocha Mosairu	21
Pong	20
Eototo	19
Tocha	19
Kawaii	18
Kwikwilyaka	18
Mongwu	18
Mosairu	18
Poli Mana	17
Pawik'China	16
Kweo	15
Soyal	15
Wakas	15
Total	**439**

DESCRIPTION	QUANTITY ORDERED
bird	125
animal	119
chief	57
woman	55
clown	51
guard	32
Total	**439**

Figure 5.5–5
Quantity Ordered per Artist, Item Type, and Item Category (Formatted)

Measure #3: Rank
Artist Rank, Item Type Rank, Item Category Rank

PURPOSE
Determines the rank of an artist, item type, or item category based on total quantity ordered.

DAX DEFINITIONS

TABLE: ARTIST

```
ARTIST RANK = RANKX (ALL (ARTIST), PORDERLINE[QUANTITY ORDERED] )
```

TABLE: ITEMTYPE

```
ITEM TYPE RANK = RANKX (ALL (ITEMTYPE), PORDERLINE[QUANTITY ORDERED] )
```

TABLE: ITEMCATEGORY

```
ITEM CATEGORY RANK = RANKX (ALL (ITEMCATEGORY), PORDERLINE[QUANTITY ORDERED] )
```

Given that the table names are being used as part of the formula definition, we need to develop three different measures, one for each object type: Artist, ItemType, and ItemCategory. Given the similarity among the three formulas, we will only discuss artist rank in detail below.

DAX SYNTAX

ARTIST RANK	Name of measure
RANKX()	DAX function that ranks instances in a table (artist in this case) based on a specified expression (quantity ordered in this case)
ALL()	DAX function that eliminates all existing filters
PORDERLINE[quantity]	Field from the POrderLine table used by the RANKX() function

DAX FUNCTIONS AND OPERATORS

RANKX()	RANKX(Table,Expression). Determines the rank of a row (instance) in a table compared to all other rows (instances) in the same table based on the value determined by "expression."
ALL()	The ALL() function considers all the rows in a table, ignoring all filters.

USES

Specific measures are defined for three different objects: Artist (#3.1), Item type (#3.2), and Item category (#3.3). Each of these objects can be used to rank the object's instances based on quantity ordered and thus can generate an additional value for each instance—for example, the rank of a specific artist.

FURTHER DISCUSSION AND ILLUSTRATION

RANKX() and ALL() are powerful DAX functions but are also somewhat more complex in nature. Next, we will clarify the mechanics of these two functions by means of a three-step exercise. As part of this exercise, we will make use of the ALLSELECTED() DAX function, which can be described as follows:

ALLSELECTED()	The ALLSELECTED() function considers all the rows (instances) that are currently selected.

Step 1

Create a new dashboard and add a table to the canvas. Add two columns to the table (its content): (1) ARTIST (the artist's name), and (2) ARTIST RANK (the measure you defined above). Figure 5.5–7 shows the table's content definition.

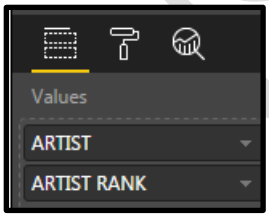


Figure 5.5–7
Definition of Artist Table with Name and Rankings

The resulting table looks as follows (figure 5.5–8); it shows the name (column 1) and the ranking (column 2) for all artists.

ARTIST	ARTIST RANK
Ahote Jones	42
Amanda Rash	28
Andrea Fernandez	10
Anthony Pricci	38
Bill Angelo	17
Blaize Menzoni	4
Brad Saylor	46
Chris Aiello	19
Clinton Rafael	41
Craig Steinberg	28
David Flewelling	10
David Lofgren	24
Don Atkins	13
Elan Raes	35
Evan Nathanson	9
Greg Wilber	24
Jaime Rebele	4
Jason Sharabani	43
Jeff Cane	28
Jennifer Anchill	24
Joe Hogue	28
Joel Pease	43
John White	46
Jonathan Sherick	35
Katie Carpenter	24
Keme Potoms	10
Laura Pirolli	4
Leslie Hitchens	28
Lindsay Jennings	13
Lori Fuller	4

Figure 5.5–8
Artist Table with Name and Rankings

Step 2

Add a slicer to your dashboard that will show the names of all artists. The content definition for the slicer looks as follows (figure 5.5–9):

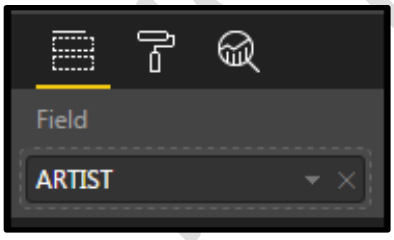
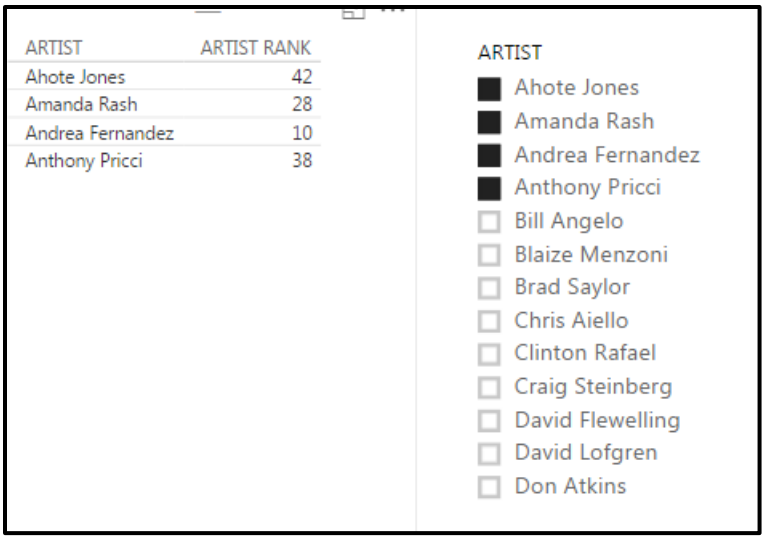


Figure 5.5–9
Content Definition of Artist Slicer

Select the first four artists in your slicer. Figure 5.5–10 shows how your canvas should look. Notice that the overall ranking of the selected artists is shown. All() means that the **ranking** is applied to "all" artists (instances), independently of whether or not they are selected.



5.5–10
Canvas Showing Overall Ranking of Selected Artists

Step 3

Create a new function, SELECTED ARTIST RANK, which we will define as follows:

TABLE: ARTIST
SELECTED ARTIST RANK = RANKX (**ALLSELECTED**(ARTIST), PORDERLINE[QUANTITY ORDERED])

Change the content definition for the table on your canvas as follows (figure 5.5–11):

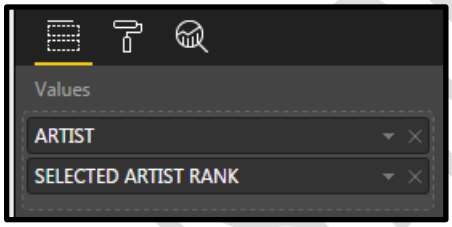


Figure 5.5–11
Revised Artist Table with Name and Rankings

Again, check the first four artists in your slicer. Your canvas should now look as follows:

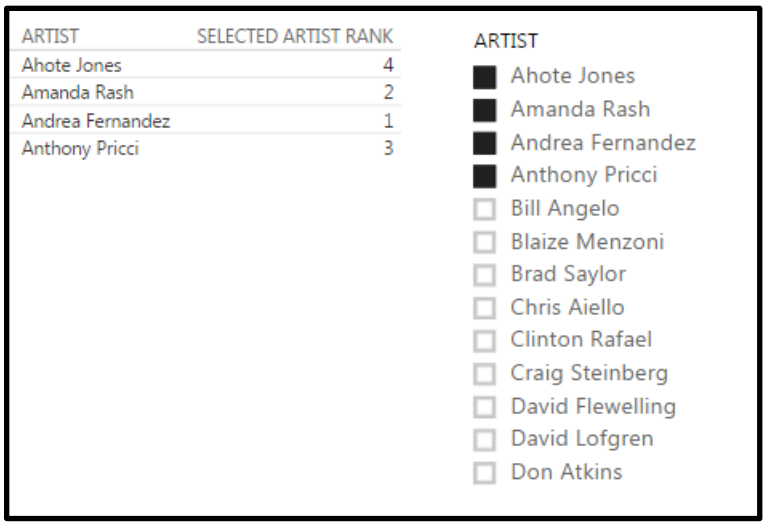


Figure 5.5–12
Canvas Showing Ranking of Selected Artists

The selected artists are now ranked against one another (i.e., the ranking is relative). This ranking is the result of using the ALLSELECTED() function instead of the ALL() function.

Measure #4
Share

PURPOSE
Determines the relative importance (%) of an object (instance), such as an artist, based on quantity ordered.

DAX DEFINITIONS[19]

TABLE: PORDERLINE

```
TOTAL QUANTITY ORDERED = CALCULATE(SUM(PORDERLINE[quantity]),ALL(PORDERLINE))

SHARE = [QUANTITY ORDERED]/[TOTAL QUANTITY ORDERED]
```

We can consider Share to be a "measure hierarchy," meaning it uses other measures in its definition.

DAX SYNTAX

TOTAL QUANTITY ORDERED	Name of measure
SHARE	Name of measure
QUANTITY ORDERED	Name of measure
CALCULATE()	DAX function that enables the definition of one or more filters for expressions
SUM()	DAX function that sums all values in a column
ALL()	DAX function that eliminates all existing filters
PORDERLINE[quantity]	Field from the POrderLine table used by the SUM() function
PORDERLINE	Table used by the ALL() function, which defines a filter as part of the CALCULATE() function

[19] We could have combined the two formulas into one, but by defining TOTAL QUANTITY ORDERED separately, we can reuse the formula in other queries; doing so also makes the explanation of the CALCULATES() function easier.

DAX FUNCTIONS AND OPERATORS

CALCULATE()	Calculate(formula,expression).[20] Enables the user to add filters to a formula and therefore to control Power BI's navigation logic.

USES

Table 5.5–1 shows that the Share measure can be applied to three different objects: Artist, Item type, and Item category.

FURTHER DISCUSSION AND ILLUSTRATION

CALCULATE()[21] is a powerful and useful DAX function that is somewhat complex in nature. Next, using the Share example, we will discuss how this function is used to control Power BI's navigation logic. For the Share measure, we used both "Total quantity ordered" and "Quantity ordered." Total quantity ordered uses the ALL() function as a filter within the CALCULATE() function to make sure that the "total" quantity is being calculated, independently of any active filters (i.e., independently of the context). In contrast, "Quantity ordered" varies depending on the active filter(s) that are in place. Below, we will create a dashboard that illustrates how to use the CALCULATE() function to control context.

Next, create a new dashboard and add a table. The content definition for the table is shown in figure 5.5–13, while the resulting table is shown in figure 5.5–14.

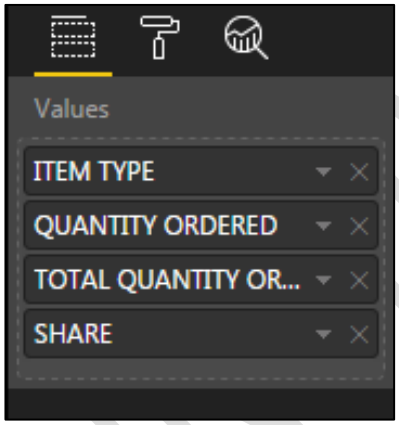


Figure 5.5–13
Context Independency: Content Definition

[20] The actual syntax for CALCULATE() is very complex and powerful; among other things, you can add multiple expressions.
[21] The Calculate() function is not supported by Excel.

ITEM TYPE	QUANTITY ORDERED	TOTAL QUANTITY ORDERED	SHARE
Nuvak'China Mana	26	439	0.06
Angwus	25	439	0.06
Chop	23	439	0.05
Hospoa	23	439	0.05
Wuyak-Kuita	22	439	0.05
Kocha Mosairu	21	439	0.05
Pong	20	439	0.05
Eototo	19	439	0.04
Tocha	19	439	0.04
Kawaii	18	439	0.04
Kwikwilyaka	18	439	0.04
Mongwu	18	439	0.04
Mosairu	18	439	0.04
Poli Mana	17	439	0.04
Pawik'China	16	439	0.04
Kweo	15	439	0.03
Soyal	15	439	0.03
Wakas	15	439	0.03
Kwahu	14	439	0.03
Wupamo	13	439	0.03
Chusona	12	439	0.03
Kachin Mana	12	439	0.03
Heee	10	439	0.02
Hon	10	439	0.02
Kowako	10	439	0.02
Nakiachop	10	439	0.02
Total	**439**	**439**	**1.00**

Figure 5.5–14
Context Independency: Table

The second column uses the "Quantity ordered" measure and is **context dependent**: the quantity ordered varies by item type; it is the total quantity ordered for a specific item type. The third column uses the "Total quantity ordered" measure and is **context independent**: the total number of items ordered (439 in this case) is shown for each specific item type. This situation is the result of using the ALL() function—which deletes all active filters—as part of the CALCULATE() function. As shown by the fourth column, relative proportions (the Share column) can then be determined by dividing the second column (which is context dependent) by the third column (which is context independent). The Share measure is **context dependent**, given that the "Quantity ordered" function is context dependent. The

example above is for item type. As shown in table 5.5–1, Share can also be applied to Artist and Item category.

Measure #5
Average Quantity Ordered

PURPOSE

To understand whether artists (vendors), on average, receive small or large orders, determined in terms of average order size. Quantity ordered is used to determine the latter.

DAX DEFINITION

TABLE: PORDER
AVERAGE QUANTITY PER ORDER = [QUANTITY ORDERED]/[NUMBER OF ORDERS]

DAX FUNCTIONS AND OPERATORS
No new DAX functions or operators for this measure; both "Quantity ordered" and "Number of orders" were defined above. This situation is another example of a "measure hierarchy."

USES
This measure is relevant for Artist only.[22]

FURTHER DISCUSSION AND ILLUSTRATION
For the definition of the "average quantity per order" measure, we have reused measure #1 (Number of orders) and measure #2 (Quantity ordered), both of which are context dependent. Next we will define a dashboard with a table that shows the average order size for each artist (vendor).

Figure 5.5–15 shows the content definition for the table. It has two columns: Artist (name) and "Average quantity per order," which is used as an indicator for average order complexity.

[22] In ORDERSOLUTIONS.pbix, we define "average quantity per order" as a measure in the PORDER table, although this information is really only relevant for "artist," and we thus could have defined "average quantity per order" as a measure in the ARTIST table instead.

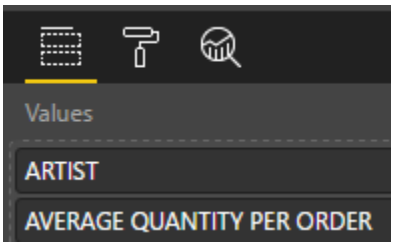


Figure 5.5–15
Applying Average Quantity per Order: Content Definition

The resulting table, with some basic formatting applied, is shown in figure 5.5–16.

ARTIST	AVERAGE QUANTITY PER ORDER
Tiffany Travis	12
Valerie Baldassari	11
Chris Aiello	10
Maska Kroes	10
Pierre Eryan	10
Marissa Rose	10
Blaize Menzoni	8
Greg Wilber	8
Jaime Rebele	8
Jennifer Anchill	8
Laura Pirolli	8
Lori Fuller	8
Amanda Rash	7
Craig Steinberg	7
Evan Nathanson	7
Jeff Cane	7

Figure 5.5–16
Applying Average Quantity Ordered: Table

Measure #6
Growth

PURPOSE
To understand growth, in terms of quantity ordered, from one month to another.

DAX DEFINITIONS

TABLE: PORDER

```
FEBRUARY QUANTITIES = CALCULATE(PORDERLINE[QUANTITY ORDERED],MONTH(PORDER[DATE])=2)

MARCH QUANTITIES = CALCULATE(PORDERLINE[QUANTITY ORDERED],MONTH(PORDER[DATE])=3)

GROWTH = [MARCH QUANTITIES] - [FEBRUARY QUANTITIES]
```

DAX SYNTAX

FEBRUARY QUANTITIES	Name of measure (measure hierarchy)
MARCH QUANTITIES	Name of measure (measure hierarchy)
GROWTH	Name of measure (measure hierarchy)
PORDERLINE[QUANTITY ORDERED]	Name of measure defined as part of the PORDERLINE table
CALCULATE()	DAX function that enables the definition of one or more filters for expressions
MONTH()	DAX function that returns the number of the month for a given date
PORDER [DATE]	Field from the POrder table used by the MONTH() function

DAX FUNCTIONS AND OPERATORS

This measure does have one new function: MONTH(). All the other functions were defined earlier in this chapter.

MONTH()	Returns the number of the month for a given date (1–12).

USES

As shown in table 5.5–1, the Growth measure applies to three different objects: Artist, Item type, and Item category.

FURTHER DISCUSSION AND ILLUSTRATION

"Growth" is a measure hierarchy with different levels. Growth reuses the "February quantities" and "March quantities" measures, while the latter two reuse the "Quantity ordered" measure. Further, CALCULATE() is used to hardwire context—February and March. The growth measure really compares quantity ordered in March with quantity ordered in February.

Figures 5.5–17 through 5.5–20 show how to apply growth to all three objects: Artist, Item type, and Item category. In all three cases, the Table visualization is used. Some basic formatting was applied for the dashboard definition shown in figure 5.5–20.

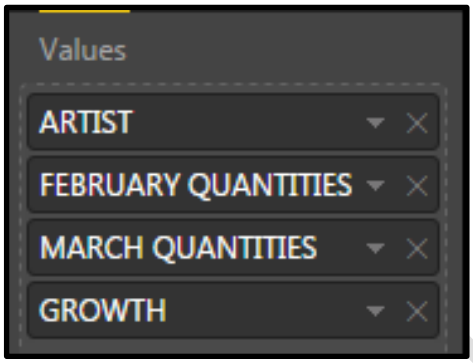


Figure 5.5–17
ARTIST GROWTH: Content Definition

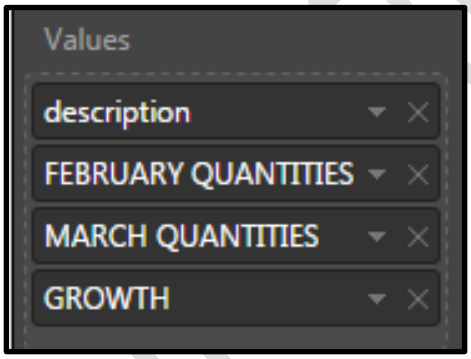


Figure 5.5–18
ITEM TYPE GROWTH: Content Definition

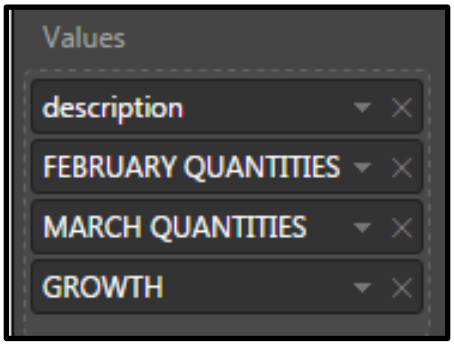


Figure 5.5–19
ITEM CATEGORY GROWTH: Content Definition

ARTIST

ARTIST	FEBRUARY QUANTITIES	MARCH QUANTITIES	GROWTH
Ahote Jones		3	3
Amanda Rash	7		-7
Andrea Fernandez		5	5
Anthony Pricci	5		-5
Bill Angelo	7		-7
Blaize Menzoni		10	10
Chris Aiello		10	10
Craig Steinberg	7		-7
David Flewelling	6		-6
David Lofgren	6		-6
Don Atkins	10		-10
Evan Nathanson	11		-11
Greg Wilber	8		-8
Jaime Rebele	8		-8
Jeff Cane		7	7
Jennifer Anchill	8		-8
Joe Hogue	7		-7
Jonathan Sherick		3	3
Katie Carpenter	6		-6
Keme Potoms	3		-3
Laura Pirolli	10		-10
Leslie Hitchens		3	3
Lindsay Jennings	8		-8
Lori Fuller	6		-6
Marissa Rose	9		-9
Mark Boaman	7		-7
Maska Kroes	9		-9
Matt Scoville	8		-8
Nate Freed	8		-8
Nicole Bahr	10		-10
Pierre Eryan	10		-10
Ray Lenno	9		-9
Sam Penn	5		-5
Steve Forrest	5		-5
Tom Garland	6		-6

ITEM TYPE

description	FEBRUARY QUANTITIES	MARCH QUANTITIES	GROWTH
Antelope Kachina	10		-10
Bear Kachina	10		-10
Broad-Faced Kachina	7	3	-4
Buffalo Kachina	10		-10
Butterfly Girl	6	4	-2
Chicken Kachina	7	3	-4
Cow Kachina	7	3	-4
Crow Kachina	10		-10
Duck	4	6	2
Eagle	8	2	-6
Great Horned Owl	10		-10
Horse Kachina	5	5	0
Hummingbird Kachina	10		-10
Kachina Chief	10		-10
Long-Billed Kachina	10		-10
Mocking Kachina	10		-10
Mountain Sheep Kachina	10		-10
Return Kachina	10		-10
Road Runner Kachina	10		-10
Silent Warrior	10		-10
Snake Dancer	10		-10

ITEM CATEGORY

DESCRIPTION	FEBRUARY QUANTITIES	MARCH QUANTITIES	GROWTH
animal	62	8	-54
bird	59	11	-48
chief	40		-40
clown	25	5	-20
guard	17	3	-14
woman	16	14	-2

Figure 5.5–20[23]
The "Growth" Measure Applied to Artist, Item Type, and Item Category

[23] March was obviously a slow month for KaDo from a purchasing perspective, as indicated by the many nulls in the "March quantities" column (item types with no orders) and the negative numbers in the Growth column.

CHAPTER 6
DATA COLLECTION

Learning Objectives

- ➔ Learn how to extract data using data connectors.
- ➔ Learn how to profile data.
- ➔ Learn how to clean data.
- ➔ Learn how to integrate data.

The previous chapter discussed how to build powerful information models, but we can only do that if we start from an organized data set. In this chapter, you will learn how to create such a data set using data collection tools. Data collection aims to extract, profile, clean, and integrate data in order to create an organized data set. Figure 6.1 reiterates where data collection is situated in the data process chain.

Further Reading

For a more in-depth discussion of the importance of the data collection phase in the data process chain, see:

Press, Gil, "Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Surveys Says," *Forbes*, March 23, 2016. Available at https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#ef746d66f637.

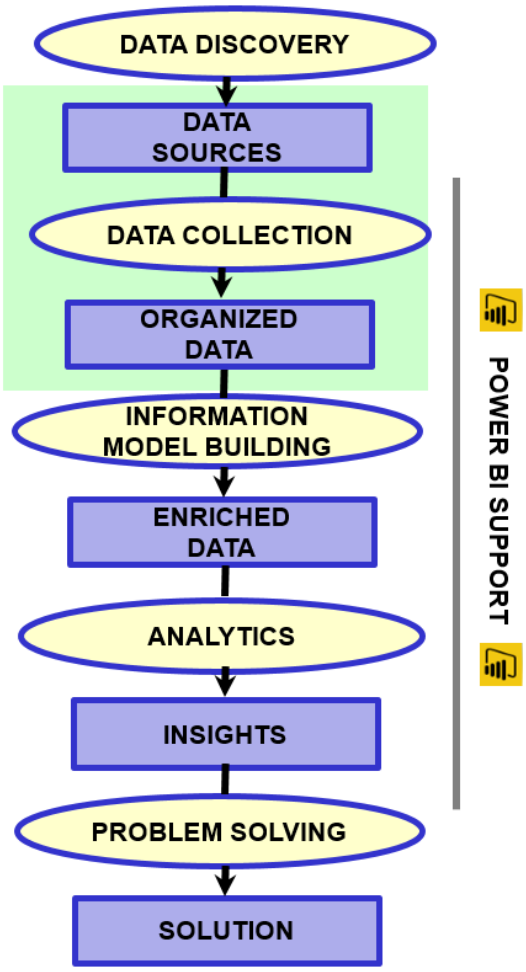


Figure 6.0–1
Data Collection in the Data Process Chain

Data Collection aims at extracting, profiling, cleaning, and integrating data in order to create an "organized" data set.

6.1 WHERE TO FIND THE DATA SETS

The purpose of this chapter is to learn how to develop a clean, well-structured, and integrated data set starting from a number of different data sources. We will use KaDo's sales system for illustrative purposes; we will assume that the company keeps its sales-related data in three separate systems (data sources), as follows.

1) ITEMDATA.TXT

KaDo has a legacy system that keeps track of the individual products (dolls) the company buys and sells. The legacy system is able to generate a TXT file[24] with item-related information; this text file is essentially a data dump.

[24] This is actually a Comma-Separated Values (or CSV) file.

2) CUSTOMERDATA.ACCDB
KaDo keeps all its customer information in this Access database.

3) SALESDATA.XLSX
KaDo uses an Excel spreadsheet to keep track of sales. More specifically, the company has a separate worksheet for each week: PERIOD1, PERIOD2, PERIOD3, and PERIOD4.

You can find all three data sources, ITEMDATA, CUSTOMERDATA, and SALESDATA, in your DropBox folder.

For this section, we will start from an empty project. Launch Power BI to create a new project and name it **SALES.pbix**. To assign a name to the project, first click on File (main menu) and then on Save As.

6.2 UNDERSTANDING THE DATA

6.2.1 The Problem to Be solved

The organization of the data set involves multiple steps, including data extraction, data profiling and cleaning, and the integration of three data sources.

6.2.2 Exploring the Data Set and Its Structure

Our starting point is three data sources—INTEMDATA, CUSTOMERDATA, and SALESDATA—which we will transform and integrate into one data set: SALES.PBIX. To do this, it is important to first explore the structure of the different data sources and learn what the integrated data model should look like.

6.2.3 The Data Set

Figure 6.2–1 shows the integration of the three data sources—SALESDATA, ITEMDATA, and CUSTOMERDATA—into one data set: SALES.PBIX. The top part of figure 6.2–1 shows the three data sources (the input), two of which contain more than one table. For example, SALESDATA.XLSX contains four different worksheets and thus tables: PERIOD1, PERIOD2, PERIOD3, and PERIOD4 (indicated in yellow). The middle part of figure 6.2–1 shows the data transformation process:

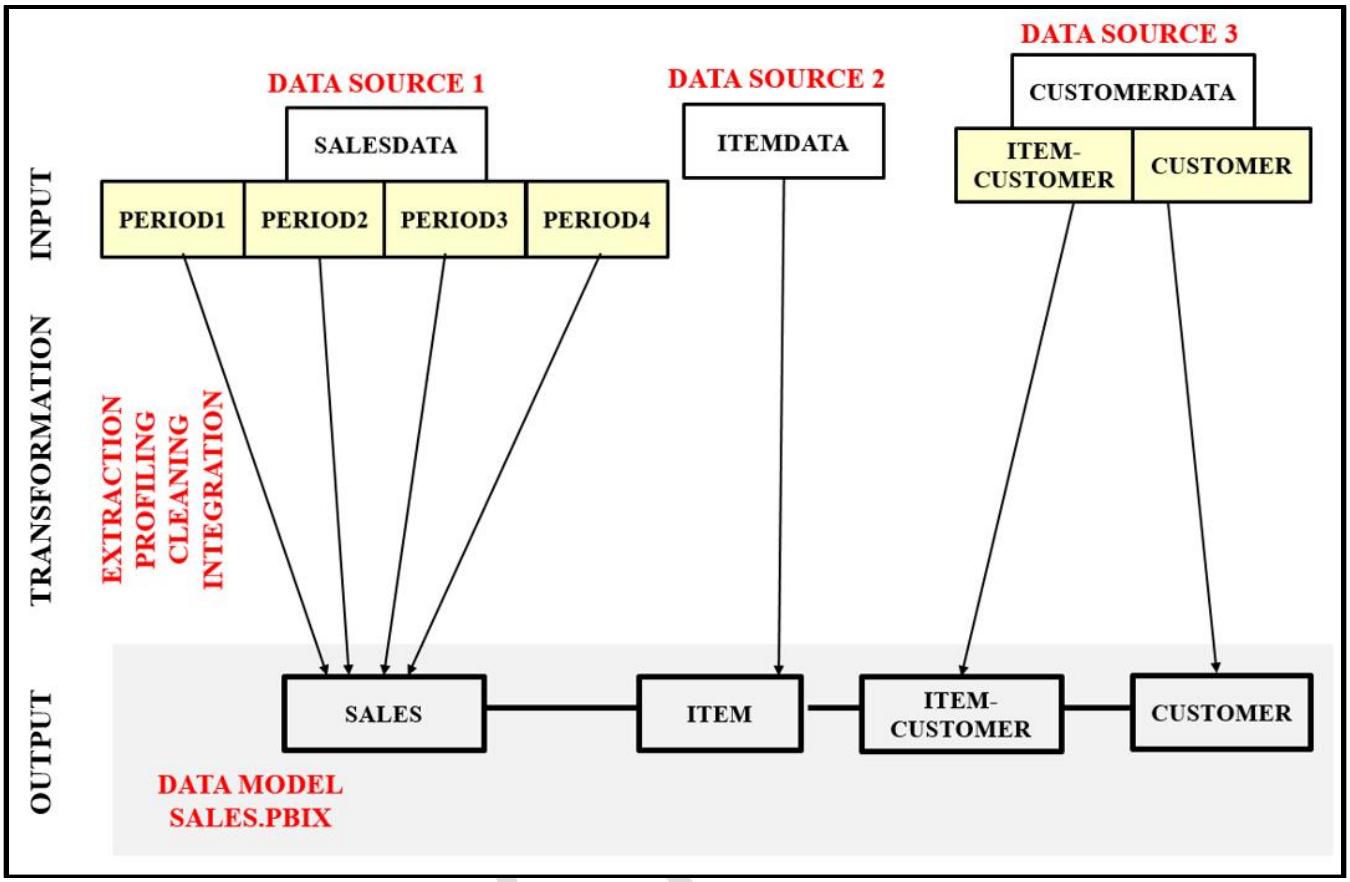


Figure 6.2–1
Creating an Integrated Data Set

extracting, profiling, cleaning, and integrating the data. The lower part of figure 6.2–1 shows the data model for SALES.PBIX (the output), including the relationships among the different tables.[25]

6.3 DATA EXTRACTION

6.3.1 Data Connectors

Power BI provides a continuously growing number of powerful data connectors, each of which has its own unique and easy-to-use interface, which makes it possible to load data into Power BI with just a few clicks. In this chapter, we will use the text/CSV, Access database, and Excel connectors, but many other connectors are available that are useful as well.

[25] The data could have been organized differently. For simplicity, we have made "item" the focus of our data model.

6.3.2 Where to Find the Data Connectors

As shown in figure 6.3–1, click on the "Get data" button to go the "Data connectors" window.

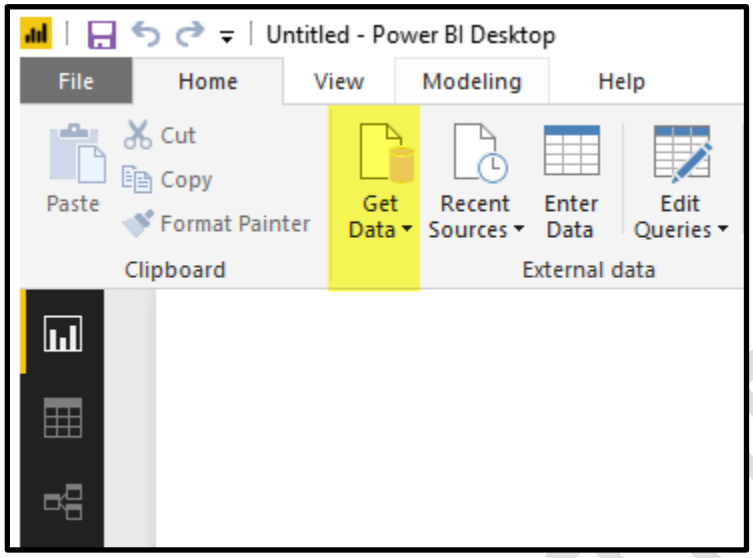


Figure 6.3–2
Finding the Data Connectors

Click on the "Get data" button.

Once you click the "Get data" button, the dialog box in figure 6.3–2 will open. As shown by the left side of the dialog box, connectors are further categorized into seven groups: All, File, Database, Power BI, Azure, Online services, and Other. The right side of the dialog box shows the actual connectors you can choose from.[26]

Useful Tip
Recent Resources

The "Recent resources" button enables you to reuse data sources that you have accessed before without having to re-enter your login information.

[26] Note: data connectors that have "(Beta)" at the end are in the testing phase.

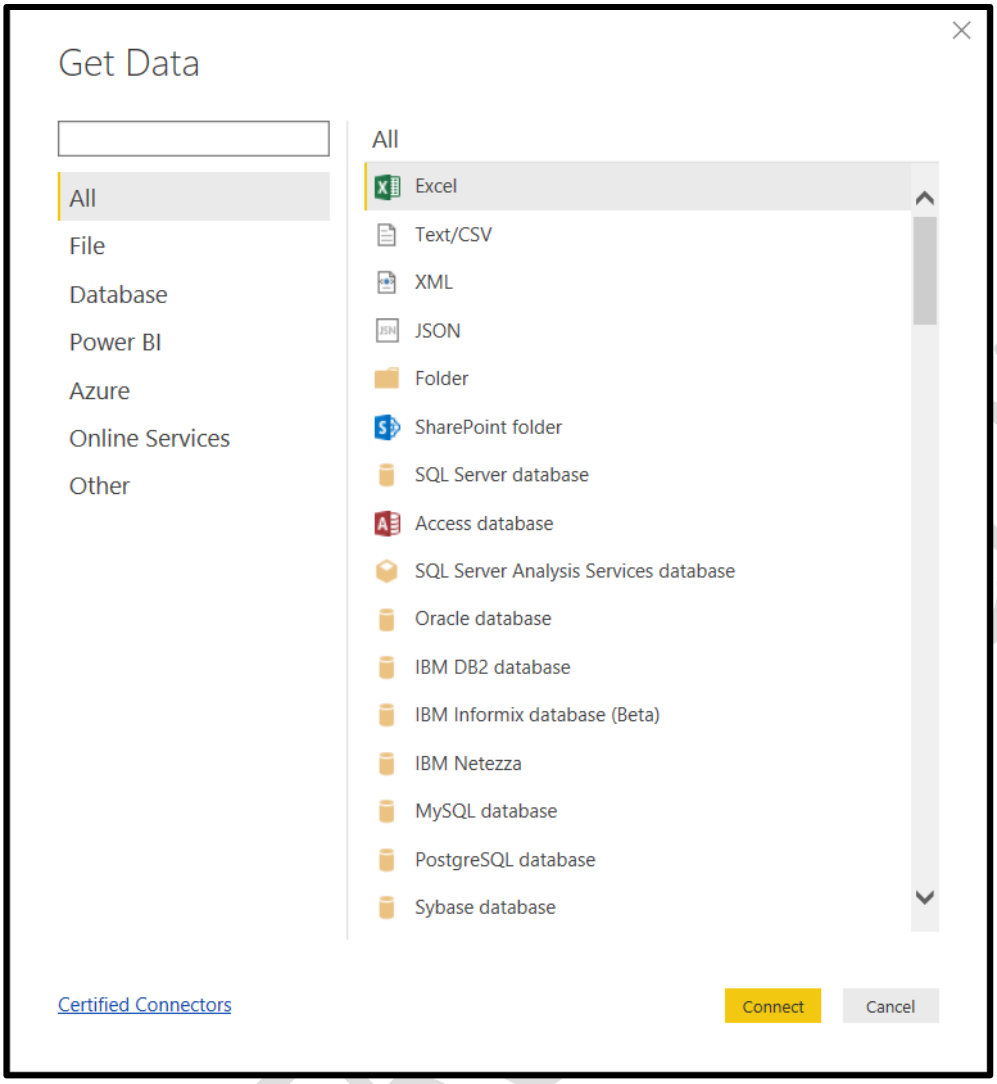


Figure 6.3–2
The "Get Data" Dialog Box

6.3.3 Working with Specific Data Connectors

The following is the general procedure for working with data connectors:

1. Select a data connector from the dialog box (figure 6.3–2) and click Connect.
2. Follow the directions specific to the data connector. Many of the interfaces give you control over how the data will be loaded.
3. Choose Edit or Load. Edit will open the query editor (to be discussed in detail below) and will allow you to further transform the data. Load will load the data into Power BI's data model, ready for information modeling and analysis. You can always go back to Edit later to further transform the data, for example by modifying and/or adding steps.

Next, we will apply this general procedure to three different file types: a text/CSV file, a Microsoft Access database, and a Microsoft Excel workbook.[27]

EXTRACTING DATA FROM A TEXT/CSV FILE: ITEMDATA.TXT

CSV, which stands for Comma-Separated Values, is a common file format. CSV files are text files; every line represents a record, and data elements (fields) are separated by delimiters such as commas. When you click on the CSV data connector, the dialog box shown in figure 6.3–3 appears. Select the ITEMDATA.TXT file and click the Open button. The dialog box shown in figure 6.3–4 then appears; this step gives you some control over how the data will be loaded.

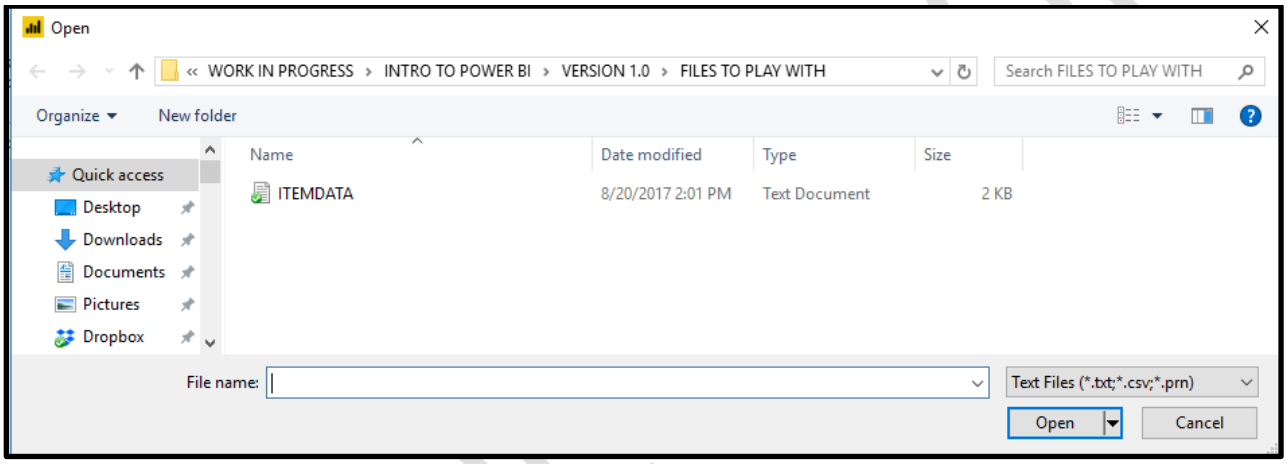


Figure 6.3–3
The "Text/CSV File" Dialog Box

[27] You can find the files required for the exercises below in your DropBox folder: ITEMDATA.TXT (text/CSV), CUSTOMERDATA.ACCDB (Access database), and SALESDATA.XLSX (Excel spreadsheet).

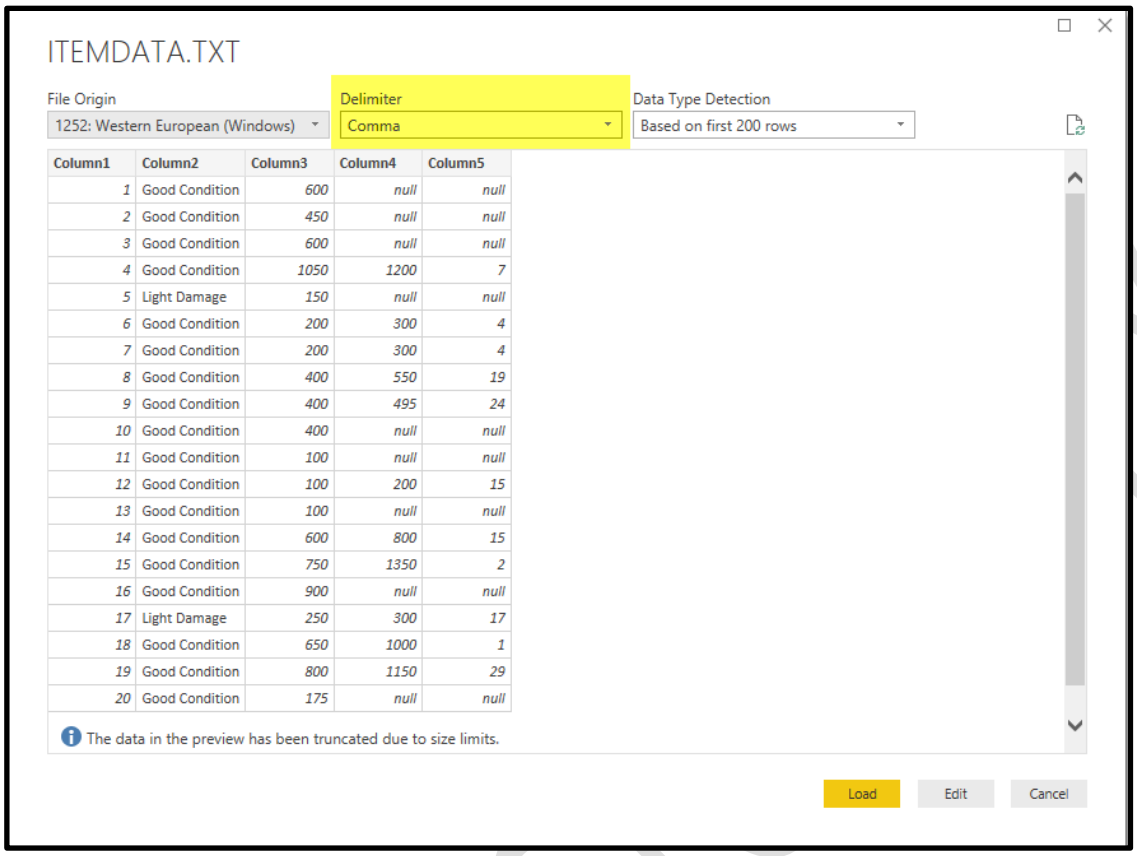


Figure 6.3–4
The "Text/CSV File" Control Dialog Box

Controlling How Data from a TEXT/CSV File are Loaded

First, Power BI lets you choose the delimiter (i.e., the character that separates the different data fields). Keep "Comma" for the ITEMDATA.TXT file.

Second, in "Data type detection," Power BI will try to automatically detect the data types for each of the fields. Again, you are given some control over the process: don't detect the types, use more data, etc. Use the suggested option for the ITEMDATA.TXT file: "Based on first 200 rows."

Next, click on Load. This will load the data into Power BI's data model; an ITEMDATA table is then created. Later in this chapter we will use the query editor to transform the data structure of the ITEMDATA table.

EXTRACTING DATA FROM A MICROSOFT ACCESS DATABASE: CUSTOMERDATA.ACCDB

Tremendous amounts of business data are stored in databases such as Microsoft Access, DB2, MySQL, and SQL Server. In this book we will extract data from a Microsoft Access database, but data extraction processes for other databases follow a similar pattern. Again, click on the "Get data" button, choose Database, select "Access database," and click Connect: see figure 6.3–5.

Side Note

You will need Microsoft Access on your machine to extract data from CUSTOMERDATA.ACCDB. If you don't have Microsoft Access on your machine or if you have issues with extracting data from the Access file (you get an error message), then follow the instructions in Appendix 1.

Next, select the CUSTOMERDATA.accdb file and click Open. (See figure 6.3–6.)

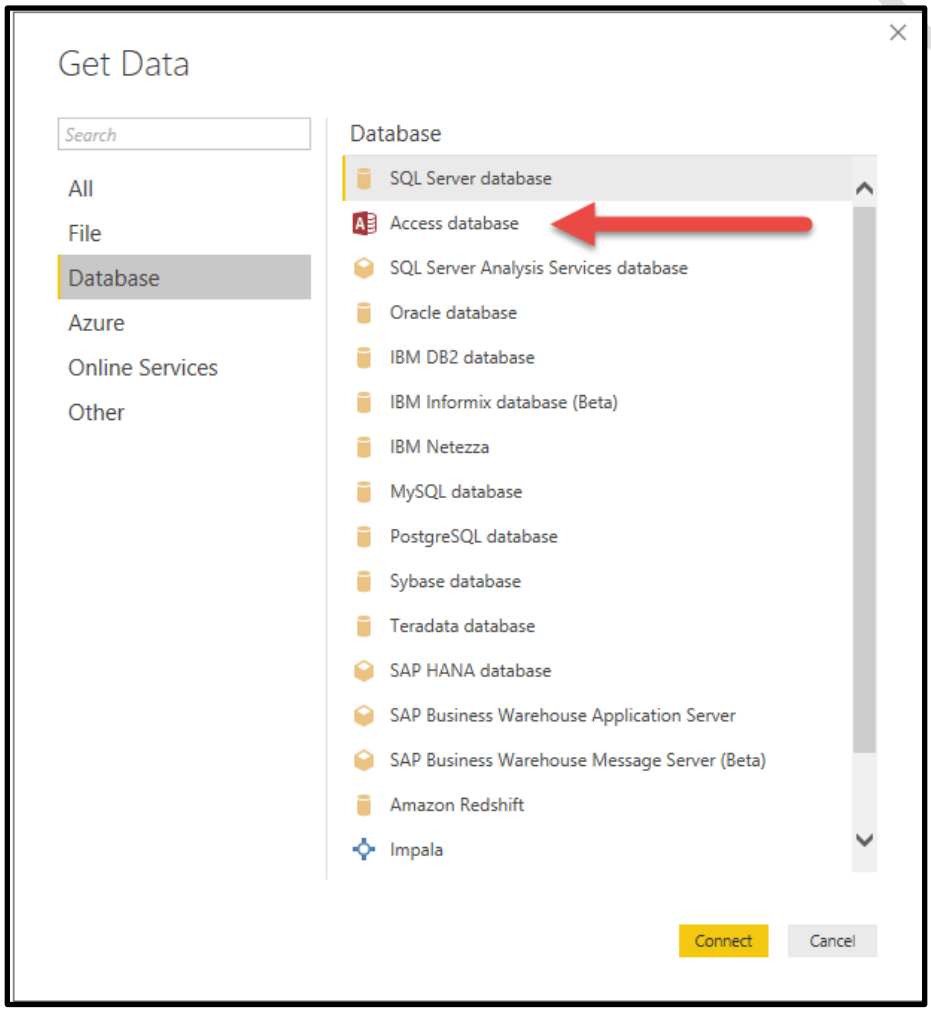


Figure 6.3–5
The "Get Data" Dialog Box; Selecting the Access Database

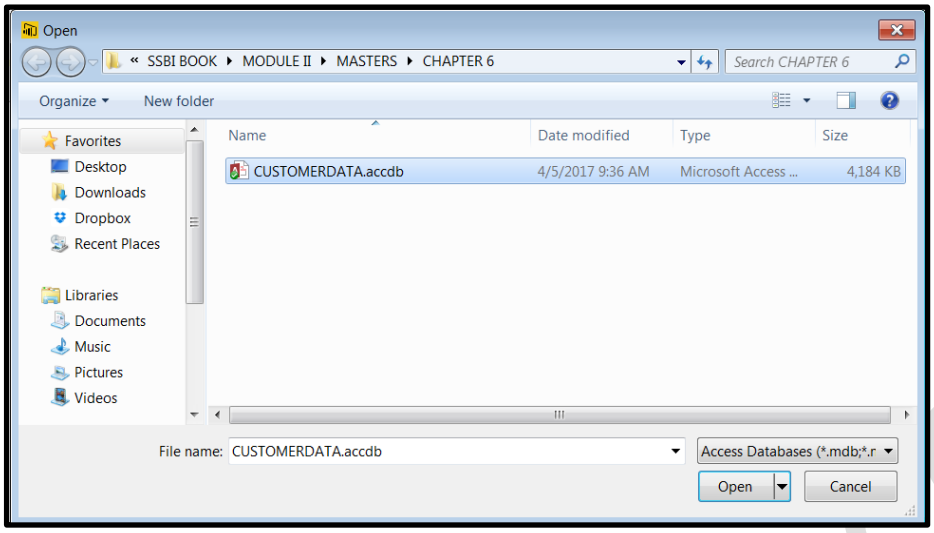


Figure 6.3–6
Selecting the CUSTOMERDATA.accdb Database

The navigator window then opens (see figure 6.3–7), which shows you all tables (CUSTOMERDATA and ITEM-CUSTOMER) and queries (NON-DELAWARE CUSTOMERS) in the database. You can download any combination of tables and queries into Power BI. For our purposes, we need the CUSTOMERDATA and ITEM-CUSTOMER tables. If you click on a table, then the right side of the screen will give you a preview of the data. Again, you could edit the data before you load them. We will load both tables now (click the Load button) and then Edit them later (see the discussion of Power Query below).

Useful Tip
Data Source Transformations

You can filter, clean, and transform data using database software. For example, you could use a query, called NON-DELAWARE CUSTOMERS, to select customers who live outside of Delaware. Then load the query (as opposed to the table) into Power BI.

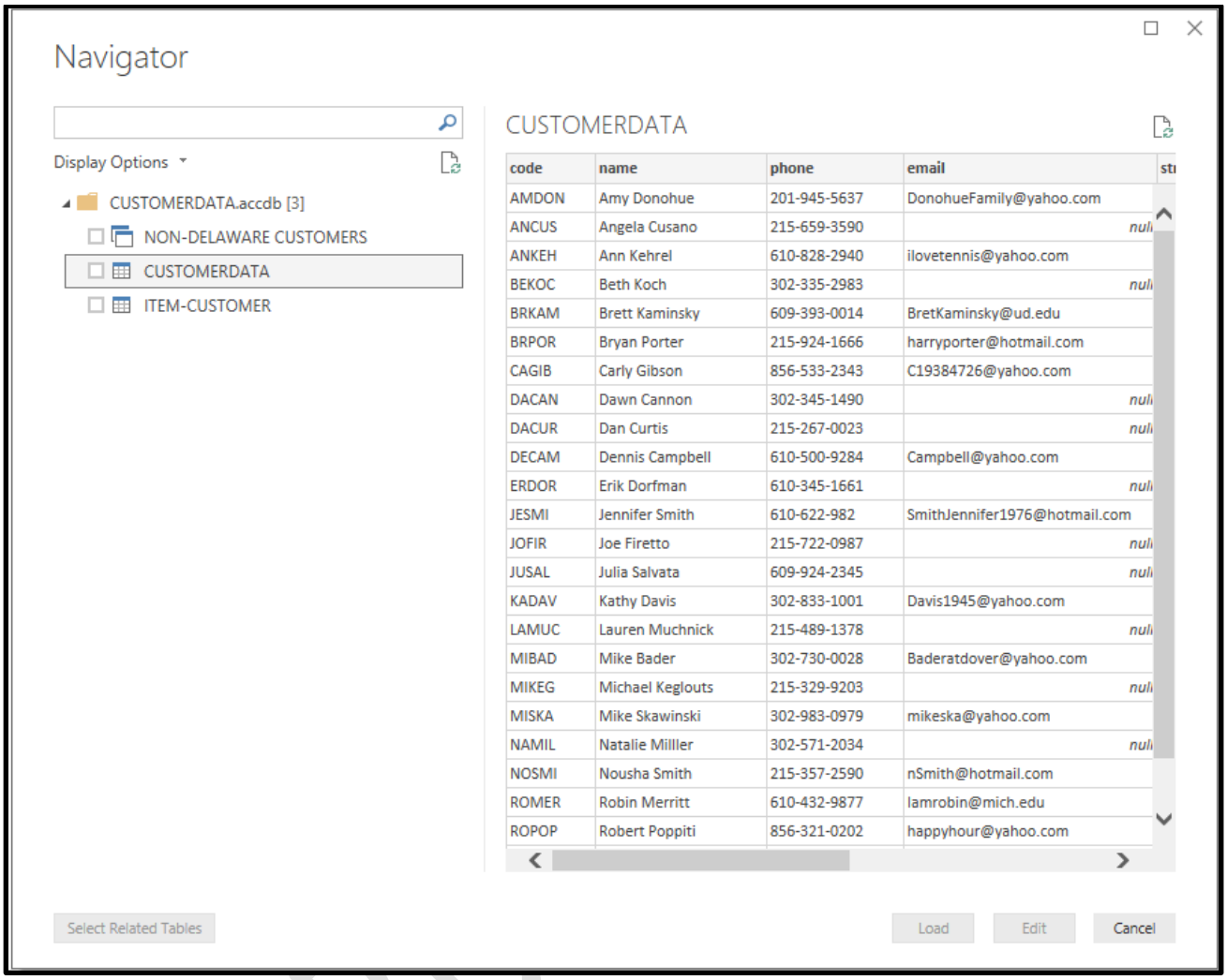


Figure 6.3–7
The "Microsoft Access" Control Dialog Box

EXTRACTING DATA FROM AN EXCEL SPREADSHEET: SALESDATA.XLSX

Most businesses also have tremendous amounts of data stored in spreadsheets. For example, KaDo has an Excel workbook that keeps track of sales. Worksheets are created on a weekly basis. Figure 6.3–8 shows the worksheet for period 4 (week 4). The first column shows what items—their IDs (number)—were sold that week. The remainder of the columns show the dates in the week items were sold; a 1 indicates that a specific item was sold on that day. This format makes it easy to answer questions such as how many items were sold on a specific day and how many items were sold in a week. This might not be the best format for other questions, however, such as monthly sales.

	A	B	C	D	E	F	G
1	number	5/15/2017	5/16/2017	5/17/2017	5/18/2017	5/19/2017	
2	23	1	0	0	0	0	
3	24	1	0	0	0	0	
4	25	1	0	0	0	0	
5	27	0	1	0	0	0	
6	28	0	1	0	0	0	
7	29	0	0	1	0	0	
8	30	0	0	1	0	0	
9	31	0	0	0	1	0	
10	35	0	0	0	0	1	

Figure 6.3–8
The "Period 4" Sales Worksheet

To extract the data from the SALESDATA workbook and load it into Power BI, first click on the "Get data" button, select Excel, and then click Connect. (See figure 6.3–9.)

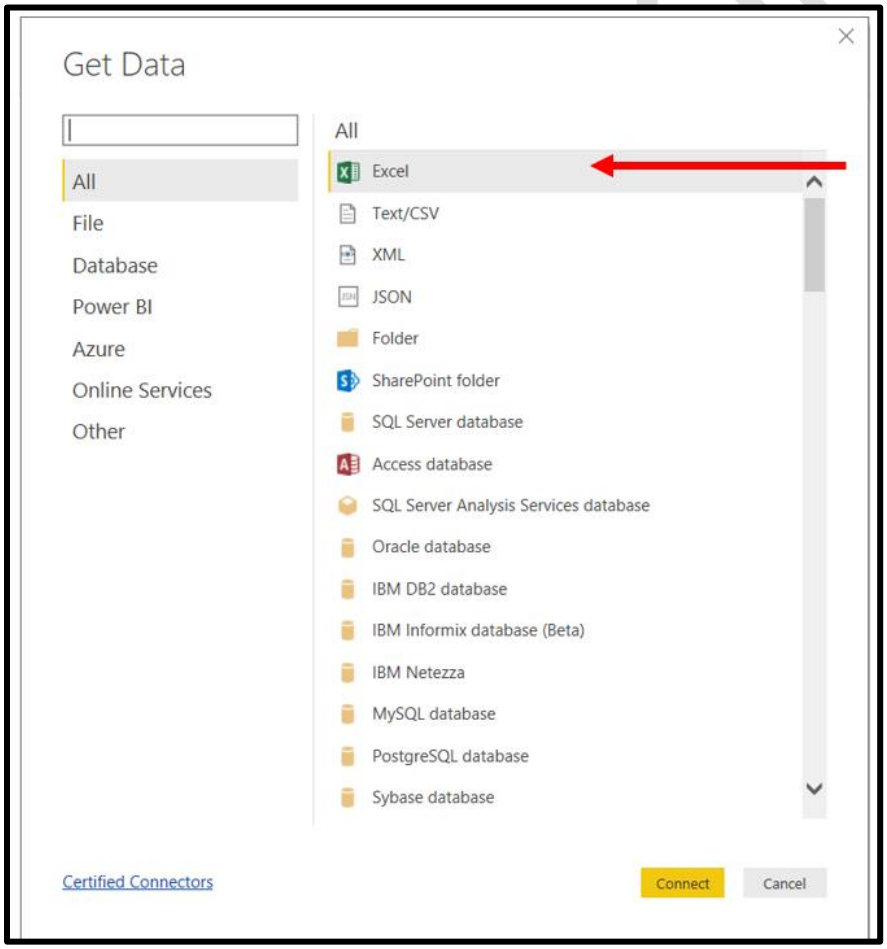


Figure 6.3–9
The "Get Data" Dialog Box; Select "Excel"

Then, select the SALESDATA Excel workbook (see figure 6.3–10).

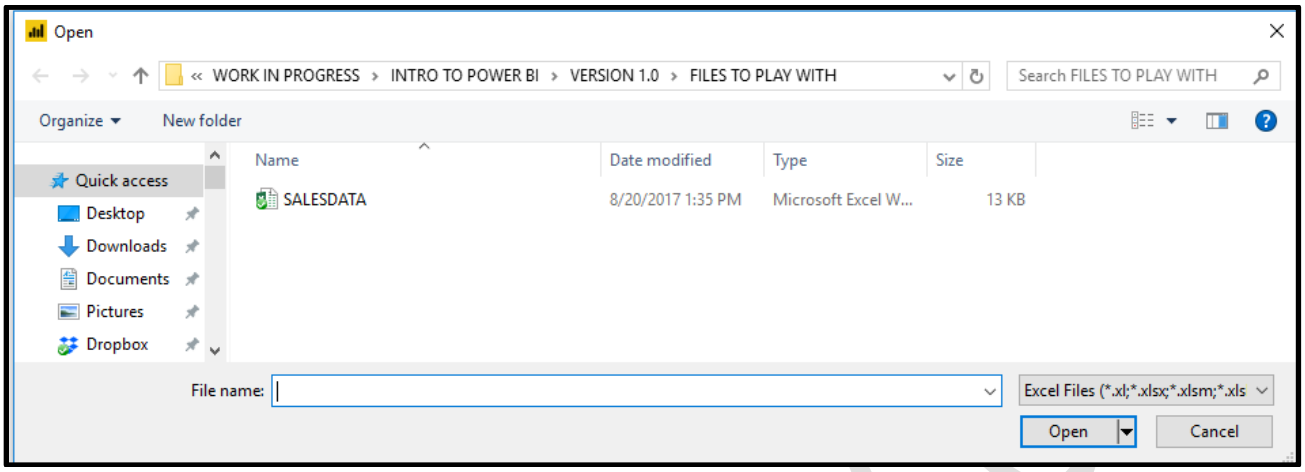


Figure 6.3–10
Select the SALESDATA Excel Workbook

Using the navigator, Power BI allows you to import worksheets, named tables, and named ranges. For this chapter, you will import all four worksheets: PERIOD1, PERIOD2, PERIOD3, and PERIOD4. You can import these worksheets simultaneously. Click all four boxes (as shown in figure 6.3–11) and then click Load.

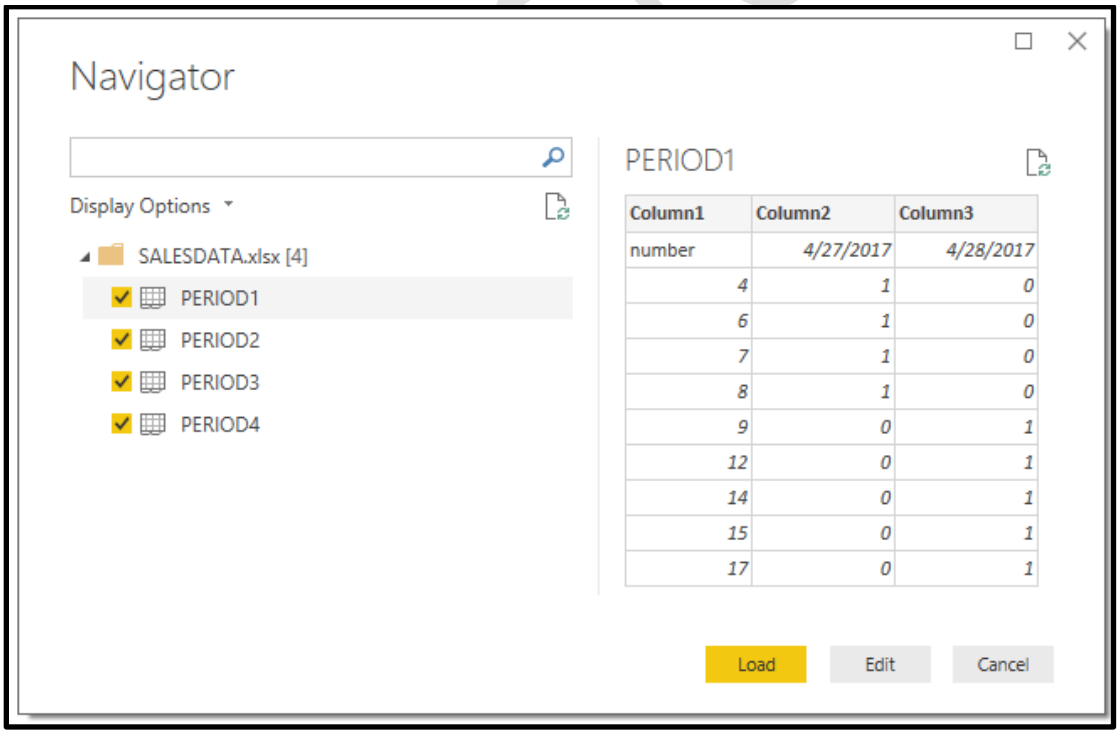


Figure 6.3–11
The "Excel" Control Dialog Box

6.4 DATA TRANSFORMATION AND THE QUERY EDITOR

Data transformation refers to the profiling, filtering, cleaning, and integrating of data, all of which prepares the data for analysis purposes. Power BI provides a powerful set of tools for data transformation, the following of which are just a few examples: splitting and merging columns, correcting data, merging tables, and joining tables. We will illustrate some of Power BI's data transformation capabilities by using these tools to profile, clean, filter, and integrate the data files loaded above. Click on Data view in SALES.PBIX, and the Fields panel should look as follows (figure 6.4–1).

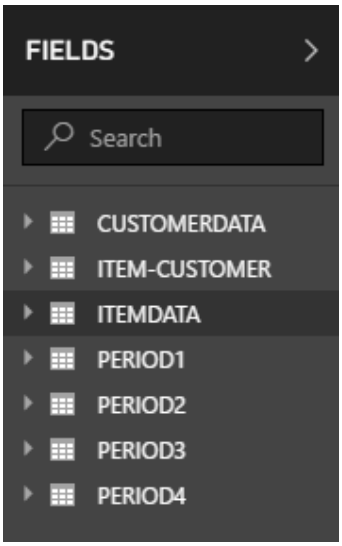


Figure 6.4–1
SALES.PBIX—Data View

6.4.1 The Query Editor

In Power BI, the transformation of data is done by means of the query editor. Click on "Edit queries" (see figure 6.4–2) to open the query editor.

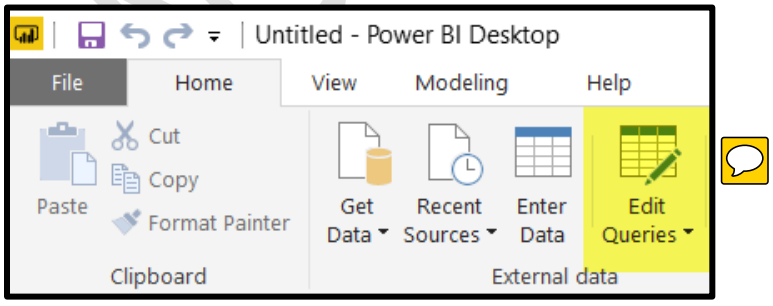


Figure 6.4–2
Opening the Query Editor

Figure 6.4–3 shows the query editor. We have divided it into six parts, each of which is briefly discussed next.

I. Main menu.
II. Ribbon. Each item in the main menu has a different ribbon. The ribbons provide a wide variety of transformation operations.
III. List of queries (the Query panel). Each table has at least one query.
IV. A query is a series of steps/instructions that loads and transforms data. Area IV shows all the steps a query consists of (the APPLIED STEPS). Steps can be edited, added, or re-ordered at any time. The (data) source is always the first step in a query.
V. The code underlying a step defined in terms of the "M" language.[28]
VI. What the data set looks like after the "active" step has been executed.[29]

Note: queries are executed before the data is re-loaded into the "data engine."

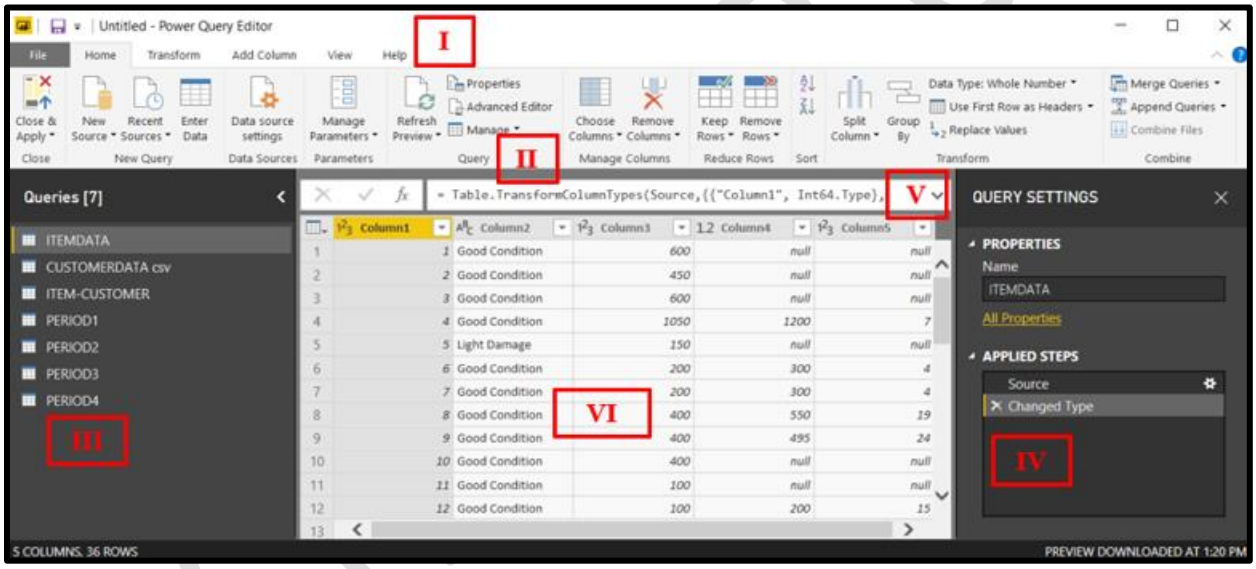


Figure 6.4–3
Anatomy of the Query Editor

Next, we will use the power query editor to transform and integrate the data from our three data sources: ITEMDATA, CUSTOMERDATA, and SALESDATA.

[28] M is a powerful language underlying Power BI transformations; it is an advanced topic that will not be covered here.
[29] A sample is shown, rather than the full data set.

6.5 TRANSFORMING "ITEMDATA"

Click on the ITEMDATA query in the Query panel (left side). The screen below (figure 6.5–1) will show up as part of the "Query settings" panel (right side). Two steps—"Source" and "Changed type"—were applied automatically when you used the data connector to extract the data from the ITEMDATA.TXT file.

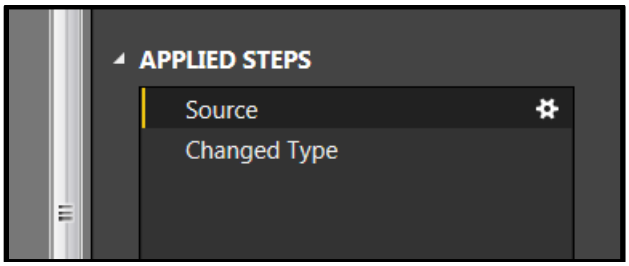


Figure 6.5–1
Automatically Generated Applied Steps

"Source" tells you where the data were extracted from. You can fine-tune these definitions to your specific needs at any time. For example, if we click on the wheel next to Source, the window below (figure 6.5–2) will appear; this window allows you to change the source, the type of file, and other items.

The "Changed type" step shows that Power BI selected "data types" when the data were extracted.

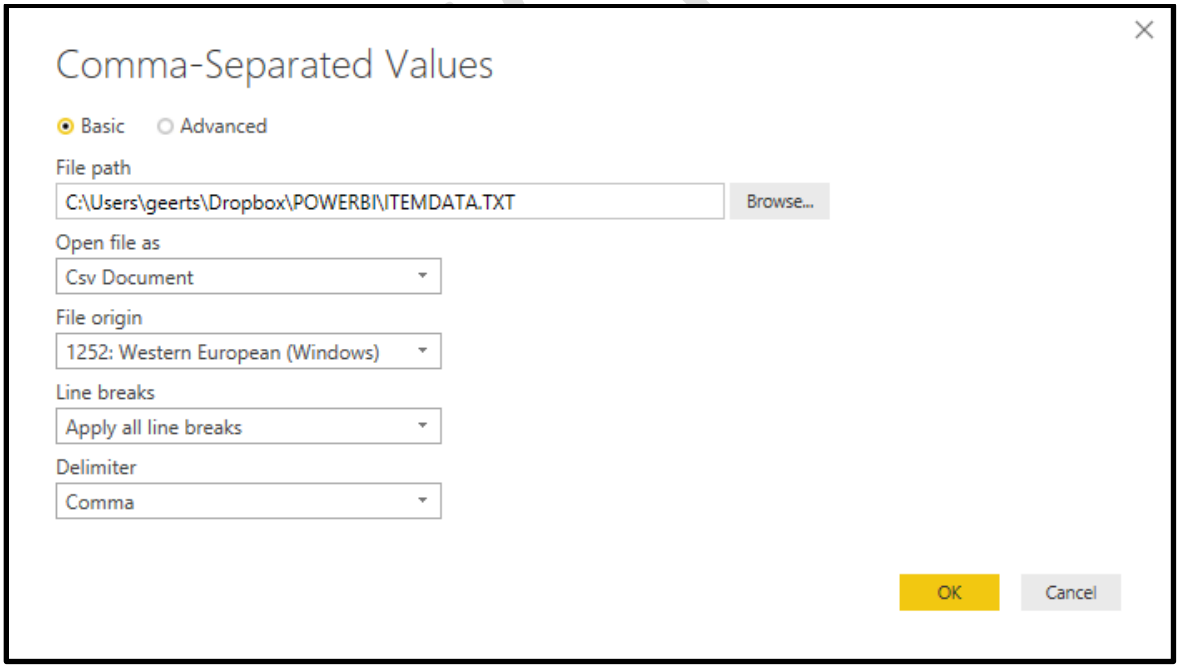


Figure 6.5–2
Changing the Data Source

Transformation typically starts with looking for quality issues; this is known as data profiling. Three issues with the data in the ITEMDATA table need to be addressed. **First**, the data in the CSV/TXT file did not provide any column/field names, so the query editor therefore names them column1, column2, etc. The second column in table 6.5–1 below shows the actual names for each of the five columns.

CURRENT COLUMN NAMES	NEW (ACTUAL) COLUMN NAMES
Column1	TAGNUMBER
Column2	DESCRIPTION
Column3	BUYPRICE
Column4	SELLPRICE
Column5	SHIPMENT

Table 6.5–1
Field Names

Follow the steps below to rename a column:

1. Select the column
2. Right-click
3. Select Rename (see figure 6.5–3 below)

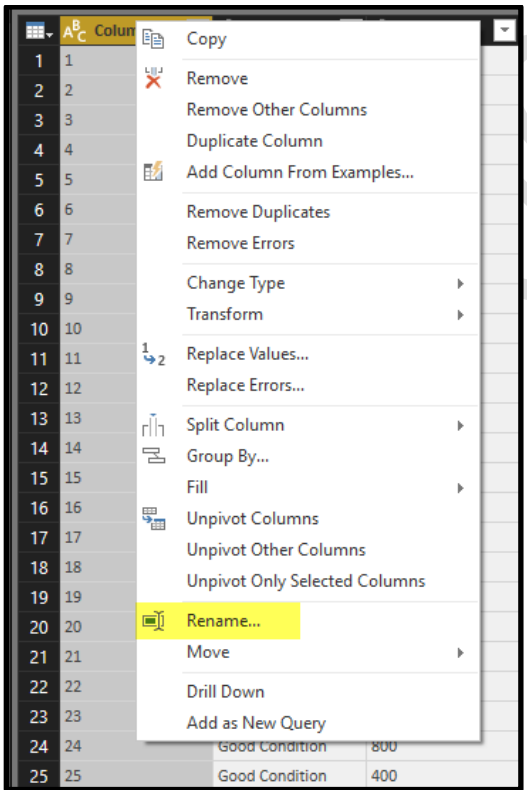


Figure 6.5–3
Renaming a Column

The revised table structure will look like this:

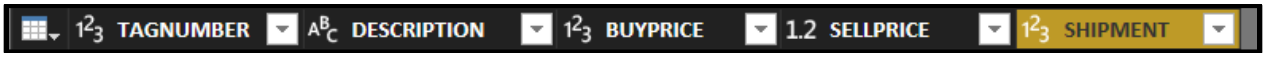


Figure 6.5–4
Revised Table Structure

The "Applied steps" panel now shows the extra step (see figure 6.5–5). The "Renamed columns" step can be deleted or modified at any time

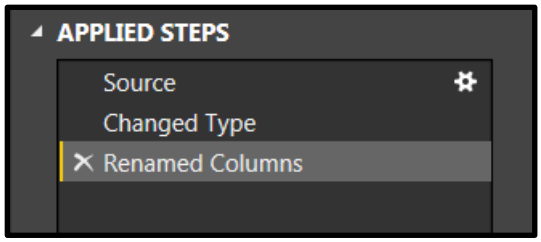


Figure 6.5–5
"Renamed Columns" Added as an Applied Step

Second, the extracted data set contains a few duplicates (i.e., it exhibits redundancy); in other words, the same fact is recorded more than once. Redundancy is a quality issue that might result in wrongly aggregated data. Addressing redundancy typically involves two steps: (1) **detecting** and (2) **correcting**. In this case, we assume that redundancy is detected by means of eyeballing, using visual detection. As shown in figure 6.5–6, sort the data in ascending order of "tag number" (click on the arrow pointing down next to the field name).

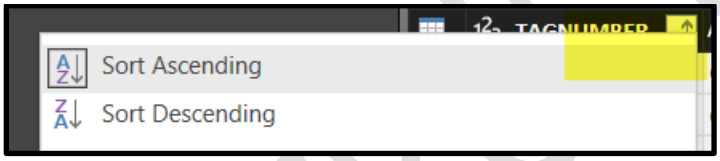


Figure 6.5–6
Sorting Records in Ascending Order of "Tag Number"

Figure 6.5–7 below shows that the data for the item with tag number 22 is recorded twice.

22	22	Good Condition	175	null	null
23	22	Good Condition	175	null	null

Figure 6.5–7
Data Redundancy

Useful Tip
Redundancy Detection Algorithms

For large sets of data, redundancy detection algorithms are typically used, instead of visual detection.

Once redundancy is detected, Power BI provides tools to eliminate the redundancy (i.e., correction) with the "Remove duplicates" option. Figure 6.5–8 shows where to find the "Remove duplicates" option in the main menu of the query editor. Figure 6.5–9 shows a more detailed screen.

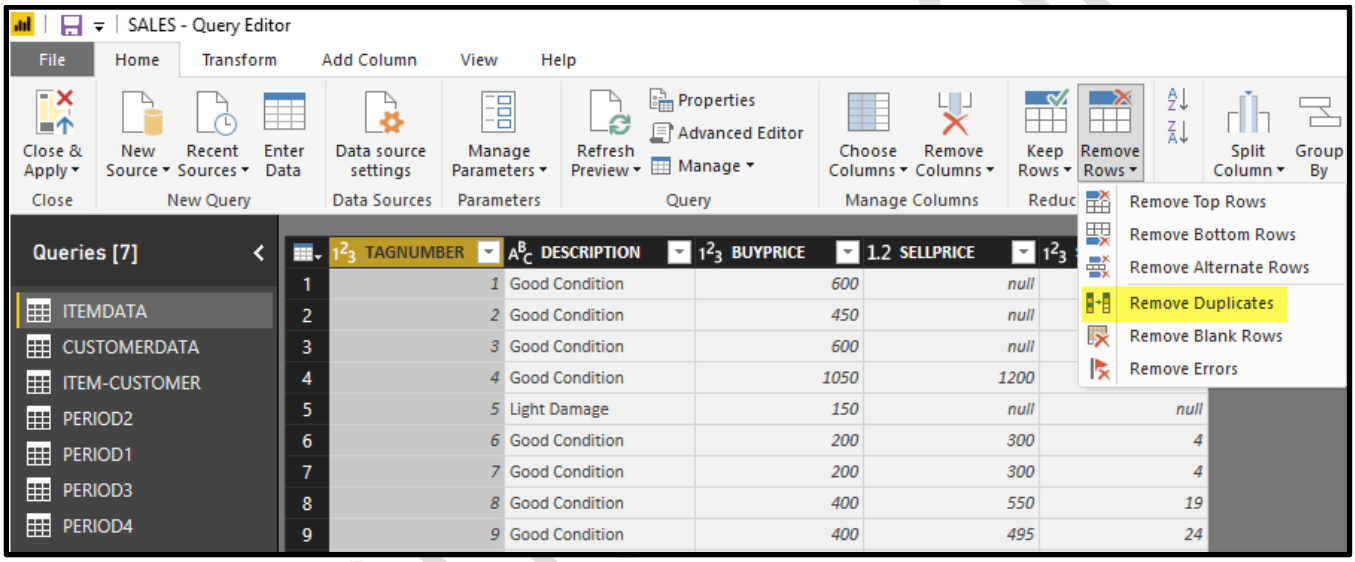


Figure 6.5–8
Remove Duplicates: Main Screen

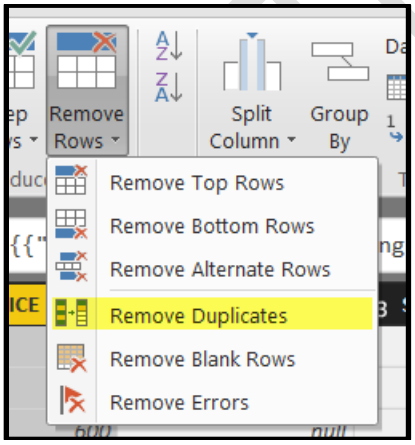


Figure 6.5–9
Remove Duplicates: Detailed Screen

Click on the "Remove duplicates" option, and the redundancy will be eliminated. An extra step will also be added to the "Applied steps" panel, as shown in figure 6.5–10.

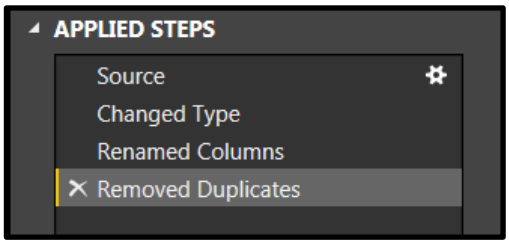


Figure 6.5–10
"Removed Duplicates" Added to the Applied Steps Pane

Third, data sets often contain logical errors. The same detection-correction process applies to logical errors. The quality of data can be assessed by running a number of tests (i.e., detection). A logical rule for the ITEMDATA data set is that *Only items that have been sold can have a "sell" price*. In Power BI, testing can be done by means of DAX formulas.

Close the query editor, save the changes you made (i.e., apply changes), and go to the Data view. Add a column[30] called LOGICAL TEST to the ITEMDATA table (Data view) using the following DAX formula:

DAX DEFINITION OF "LOGICAL TEST" ATTRIBUTE

```
LOGICAL TEST = IF(OR(ISBLANK(ITEMDATA[SELLPRICE]) &&
NOT(ISBLANK(ITEMDATA[SHIPMENT])),ISBLANK(ITEMDATA[SHIPMENT]) &&
NOT(ISBLANK(ITEMDATA[SELLPRICE]))),-1)
```

DAX SYNTAX

LOGICAL TEST	Name of column
IF()	Conditional test, which is done by a logical expression; the value generated will differ depending on whether the expression is TRUE or FALSE
OR()	Tests whether at least one of two expressions is TRUE
&&	AND operator
NOT()	Is TRUE when its expression is FALSE
ISBLANK()	Is TRUE when a field is empty
ITEMDATA[SHIPMENT]	Field from the ITEMDATA table of which the content is tested by the ISBLANK() function
ITEMDATA[SELLPRICE]	Field from the ITEMDATA table of which the content is tested by the ISBLANK() function

[30] Apply what we learned in chapter 5.

DAX FUNCTIONS AND OPERATORS

IF()	The IF() function has three arguments: IF(conditional expression, TRUE value, FALSE value) In our example, the conditional expression is: OR(ISBLANK(ITEMDATA[SELLPRICE]) && NOT(ISBLANK(ITEMDATA[SHIPMENT])),ISBLANK(ITEMDATA[SHIPMENT]) && NOT(ISBLANK(ITEMDATA[SELLPRICE]))) IF the conditional expression is TRUE, then the value –1 is generated. IF the conditional expression is FALSE, then a blank value is generated—the default, since the third argument, which is optional, is not specified.

OR()	The OR() function takes two arguments; it generates a TRUE value if at least one of the arguments is true. In the formula above, the OR() function is used to test if "sell price" is empty and "shipment" is not empty, OR the other way around

&&	Creates an AND condition between two logical expressions; the combination is TRUE only if both individual expressions are true

NOT()	Changes TRUE to FALSE or FALSE to TRUE

ISBLANK()	Returns TRUE if a field is blank

Once created, the LOGICAL TEST column shows that one error has occurred (figure 6.5–11):

TAGNUMBER	DESCRIPTION	BUYPRICE	SELLPRICE	SHIPMENT	LOGICAL TEST
33	Good Condition	900	400		-1

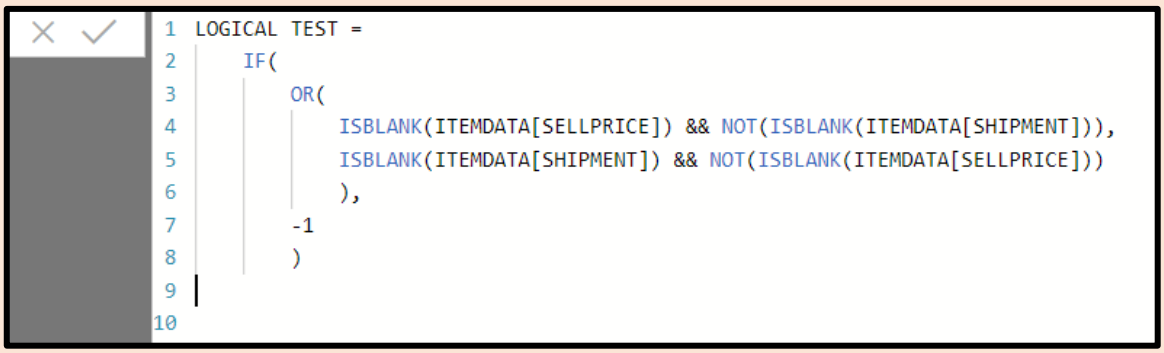
Figure 6.5–11
Use of a Column to Determine a Logical Error

Upon further analysis of our data, it turns out that the item has not been shipped yet, and the "sell price" was entered by error. This error can be corrected in the underlying data source.[31]

[31] This error was not corrected in SALESSOLUTIONS.PBIX.

Useful Tip
Structuring your DAX code

To avoid errors in formulas, it often helps to structure the code of the formula. Look at the restructured LOGICAL TEST formula below. The logic is much easier to understand.



```
LOGICAL TEST =
    IF(
        OR(
            ISBLANK(ITEMDATA[SELLPRICE]) && NOT(ISBLANK(ITEMDATA[SHIPMENT])),
            ISBLANK(ITEMDATA[SHIPMENT]) && NOT(ISBLANK(ITEMDATA[SELLPRICE]))
            ),
        -1
        )

```

Use “Shift” + “Enter” to move code to the next line.

6.6 TRANSFORMING "CUSTOMERDATA"

Open the query editor again and click on the CUSTOMERDATA query in the Query panel (left side). The screen below (figure 6.6–1) will show up as part of the "Query settings" panel (right side). These two steps were applied automatically when you used the data connector to extract data from the CUSTOMERDATA.ACCDB file.

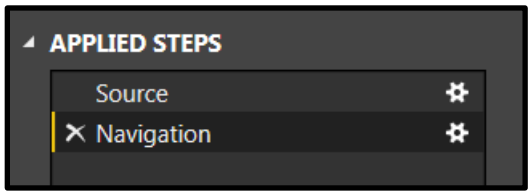


Figure 6.6–1
Automatically Generated Applied Steps

"Source" defines where the data were extracted from. "Navigation" refers to the specific tables that were extracted from the database. Again, you can fine-tune the data extraction at this point by clicking on the wheel next to Navigation.

We will follow the same detection-correction process in this case, similarly to what we did above for ITEMDATA. Let's visually inspect the data in the City field. Order this field in ascending order and take a look at possible inconsistencies, such as the same city being spelled differently (this is the detection step). If you look carefully at the city names, you will find the following two inconsistencies (figure 6.6–2):

Inconsistency 1	Inconsistency 2
Philadelphia	Wilmington
Philadelphia	Wilmington
Philadelphia	Wilmington
Philadelphia	Wimington
Philadelphia	
Philladelphia	

Figure 6.6–2
Inconsistencies Detected in CUSTOMERDATA ("City" Field); Inconsistent Spellings for "Philadelphia" and "Wilmington"

Not correcting these errors will distort the analysis. How do we correct them? As shown in figure 6.6–3, Power BI provides powerful find-and-replace tools. When you click on "Replace values" ("Transform Tab"), the window shown in figure 6.6–4 will appear. Use this window to replace "Philladelphia" with "Philadelphia" and "Wimington" with "Wilmington." The replacement will apply to the whole column.

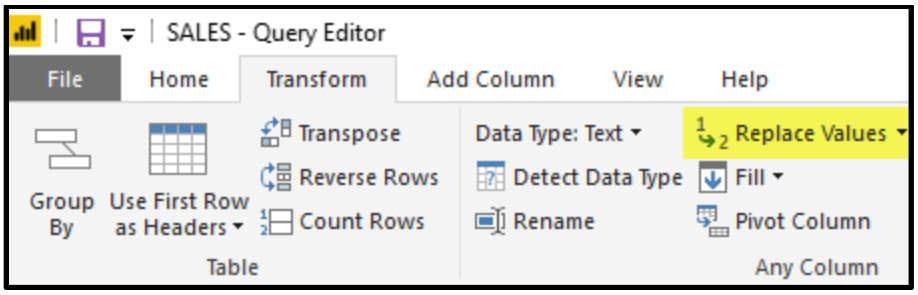


Figure 6.6–3
Power BI's Find-and-Replace Tools

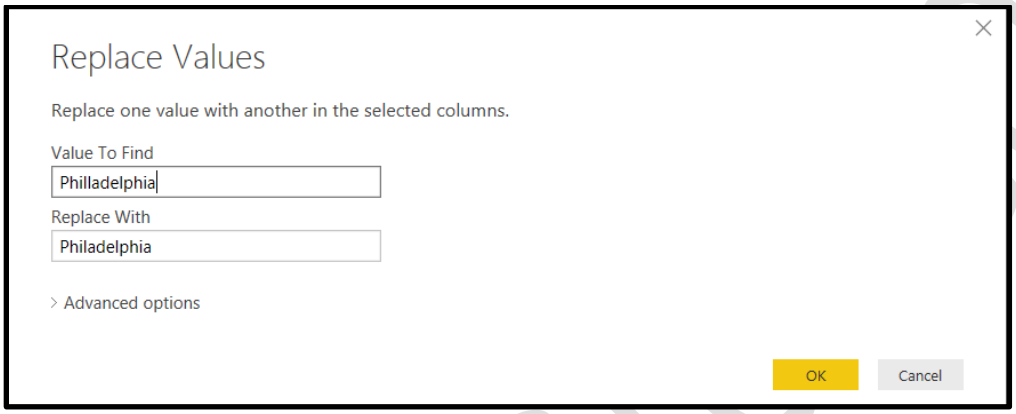


Figure 6.6–4
The "Replace Values" Window

ASSIGNMENT 6.6-A1

The ITEM-CUSTOMER table does not require any further transformation, except that the field names (column headers) need to be changed as follows (figure 6.6–5):

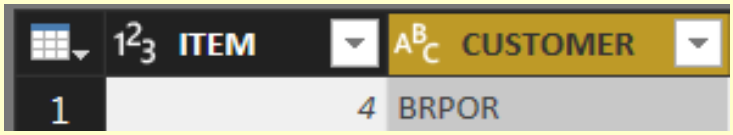


Figure 6.6-5
Changing Field Names of the ITEM-CUSTOMER Table

TRANSFORMING "SALESDATA"

Sales data are scattered across four tables: PERIOD1, PERIOD2, PERIOD3, and PERIOD4. While a question such as "How many items did we sell on May 16, 2017" is easy to answer, other questions, such as the total number of items sold, is a much bigger challenge. The integration of the four tables requires more advanced transformations.

All four tables have the same format. Figure 6.7–1 shows the structure for the PERIOD4 table—the active table in the Queries panel (left). The "Query settings" panel shows the three transformation steps that were applied during the extraction process.

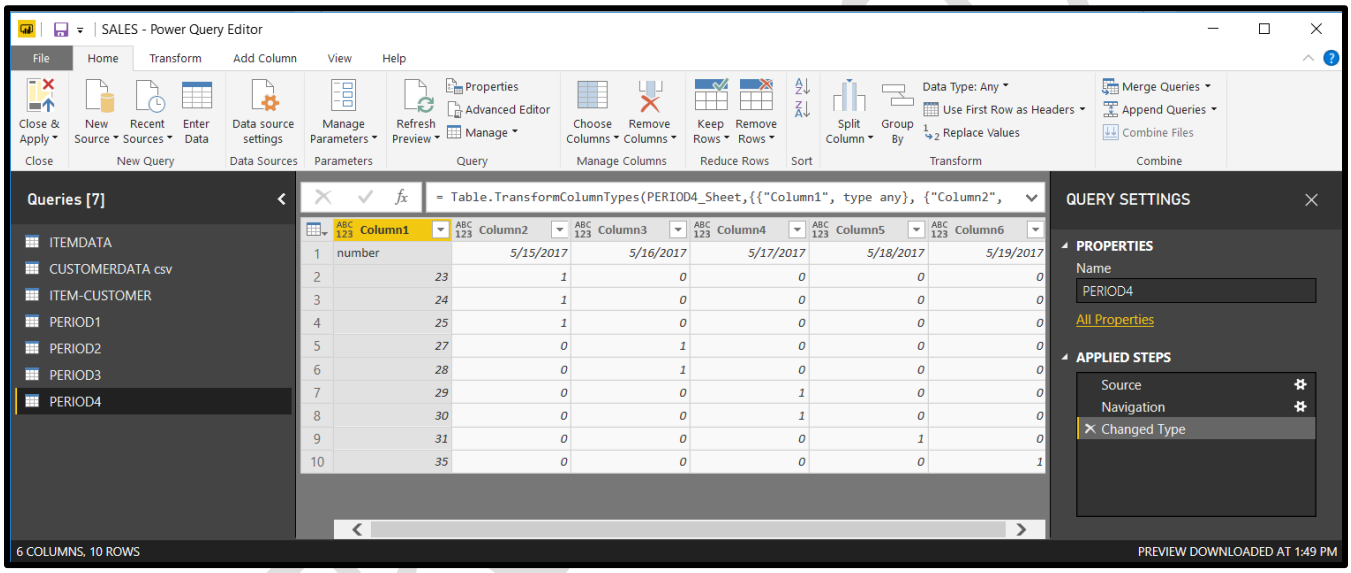


Figure 6.7–1
Structure and Transformation of the PERIOD4 Table

Step 1. Source: the data source (the Excel file) is selected.
Step 2. Navigation: each of the four worksheets selected is converted into a table.
Step 3. Changed type: Power BI has changed the data types during the extraction process.

All six columns were actually converted to the "Any" data type (figure 6.7–2).

Figure 6.7–2
Field with "Any" Data Type

A field with the "Any" data type mixes text and numbers.

The first issue we need to address is related to the table's headers. During the extraction process, the headers were put in the first row (figure 6.7–3).

Column1	Column2	Column3	Column4	Column5	Column6
number	5/15/2017	5/16/2017	5/17/2017	5/18/2017	5/19/2017
23	1	0	0	0	0
24	1	0	0	0	0

Figure 6.7–3
Column Headers Imported as First Row

Power BI's query editor has a Transform option that promotes the values in the first row of a table to be the table's headers: "Use first row as headers." (See figure 6.7–4.)

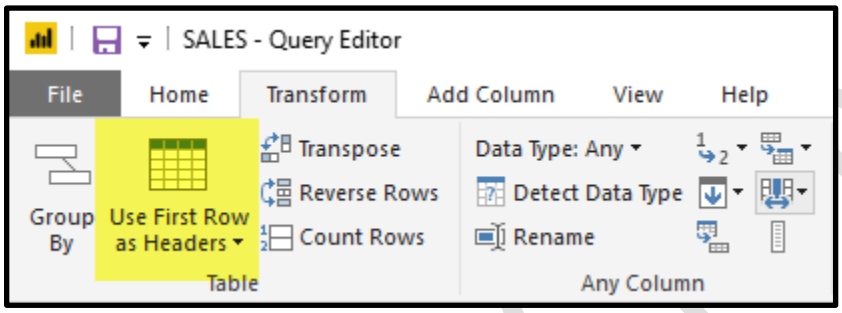


Figure 6.7–4
Using the First Row as Headers

As a result of this transformation, the PERIOD4 table should now look as follows (6.7–5):

	number	5/15/2017	5/16/2017	5/17/2017	5/18/2017	5/19/2017
1	23	1	0	0	0	0
2	24	1	0	0	0	0
3	25	1	0	0	0	0
4	27	0	1	0	0	0
5	28	0	1	0	0	0
6	29	0	0	1	0	0
7	30	0	0	1	0	0
8	31	0	0	0	1	0
9	35	0	0	0	0	1

Figure 6.7–5
PERIOD4 Table with Transformed Headers

The next issue to be addressed is the integration of the data scattered across the four worksheets by combining them, which will allow us to perform analysis across periods (weeks). This integration requires a uniform format for all four PERIOD tables. For each item, we would like to know on what day it was sold. We can accomplish this task via the "unpivoting" transformation shown in figures 6.7–6 and 6.7–7 below.[32]

Select the "number" field. Then, select "Unpivot columns" and choose "Unpivot other columns." This transformation will change the structure of the table. For each tag number, rows will be generated that add each of the headers (i.e., dates) and their corresponding value, which indicates whether or not the item was sold that day. The end result of this transformation is shown in figure 6.7–7.

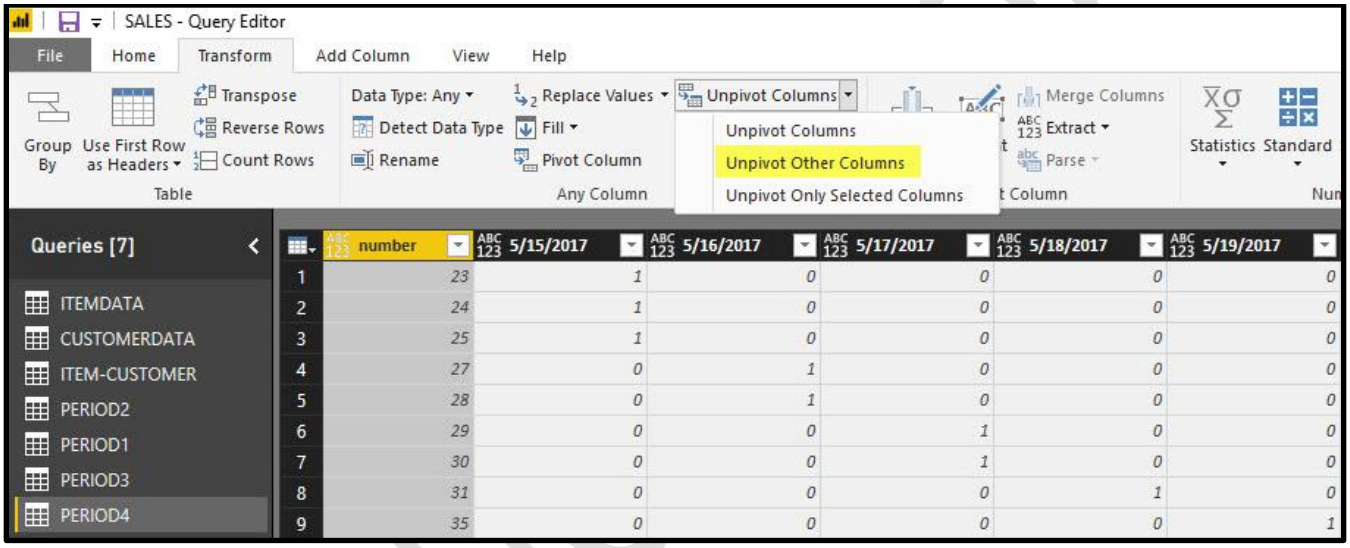


Figure 6.7–6
Unpivoting Transformation

[32] The format of the table in 6.7–6 is known as a cross-tab table, while the format of the table in 6.7–7 is known as a flat table.

	number	Attribute	Value
1	23	5/15/2017	1
2	23	5/16/2017	0
3	23	5/17/2017	0
4	23	5/18/2017	0
5	23	5/19/2017	0
6	24	5/15/2017	1
7	24	5/16/2017	0
8	24	5/17/2017	0

Figure 6.7–7
The Transformed Table

The transformed table has three columns:

- **Number**: the item's tag number.
- **Attribute**: a date
- **Value**: indicates whether an item was sold (1) or not sold (0) on that date.

A little further cleaning is needed. First, change the "data type" of the number field to "Whole number," which is the data type used for tag numbers in the ITEMDATA table. Figure 6.7–8 shows how to do a data type transformation.

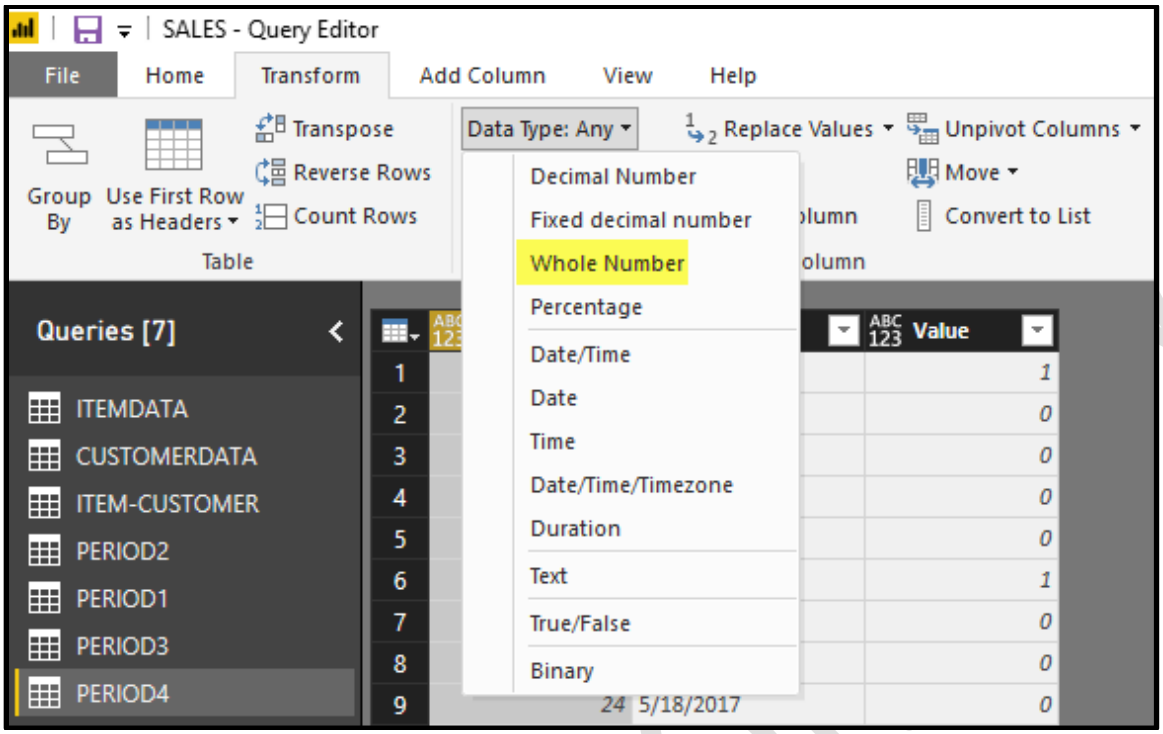


Figure 6.7–8
Data Type Transformation

Second, change the data type of the second field (Attribute) to Date, and rename its header to "DATE."

Third, change the header of the third column to "SOLD." Also, we only want to keep information for Sold items (i.e., value = 1). We can use a filter to make sure that only records with value = 1 are loaded. Figure 6.7–9 shows how to define such a filter.

Figure 6.7–10 provides an overview of the transformation steps applied to the PERIOD4 table. The resulting table (the output) is shown in figure 6.7–11.

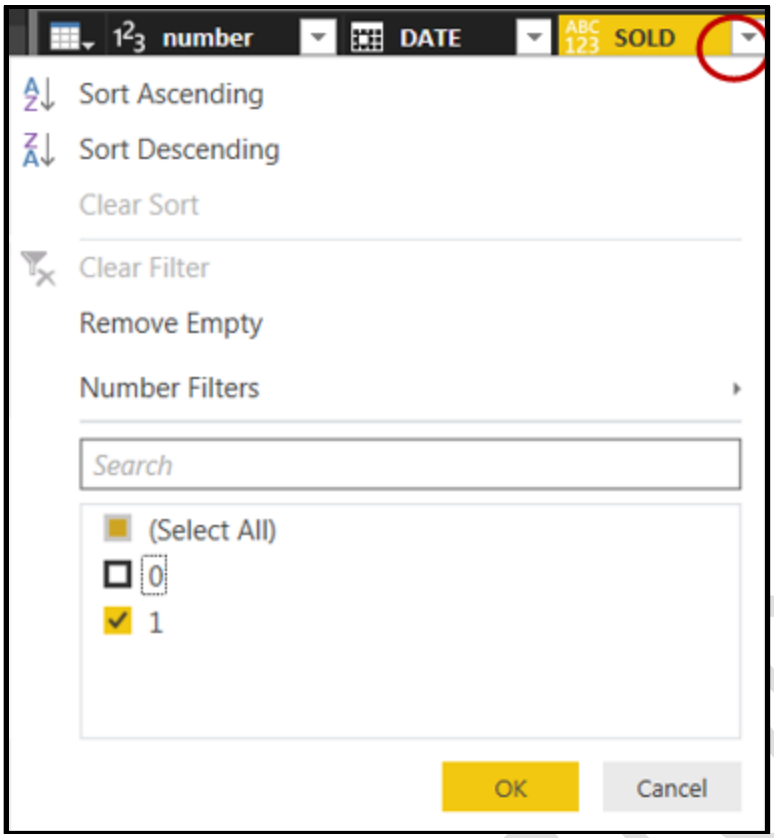


Figure 6.7–9
Adding a Filter

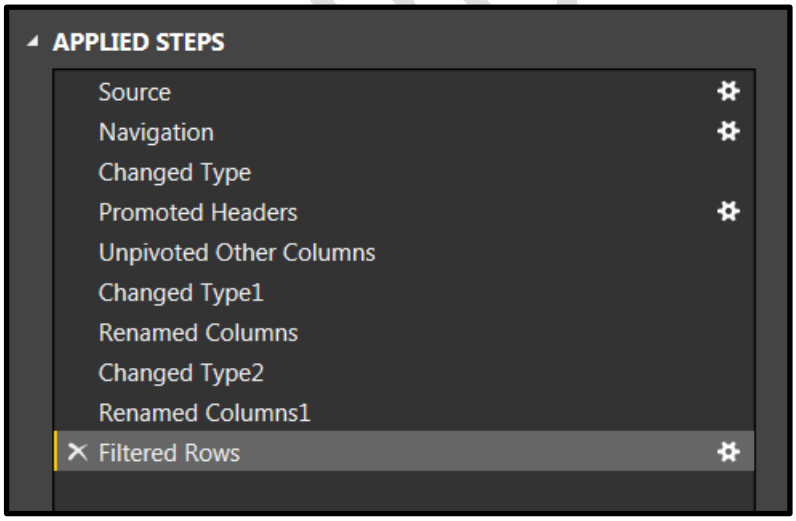


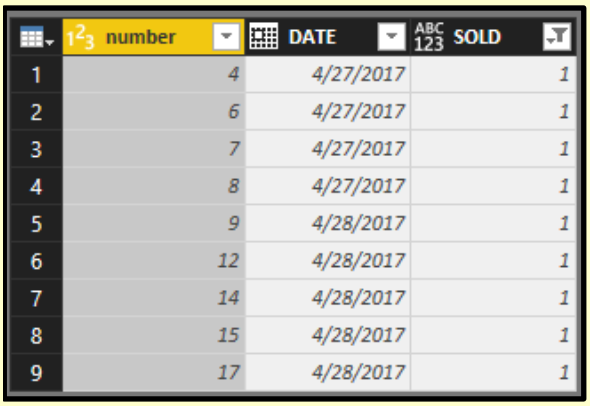
Figure 6.7–10
Overview of Transformation Steps

	number	DATE	SOLD
1	23	5/15/2017	1
2	24	5/15/2017	1
3	25	5/15/2017	1
4	27	5/16/2017	1
5	28	5/16/2017	1
6	29	5/17/2017	1
7	30	5/17/2017	1
8	31	5/18/2017	1
9	35	5/19/2017	1

Figure 6.7–11
Transformed PERIOD4 Table

ASSIGNMENT 6.7-A1

Follow the steps above to transform the PERIOD1, PERIOD2, and PERIOD3 tables. The respective results are shown in figures 6.7–12, 6.7–13, and 6.7–14.

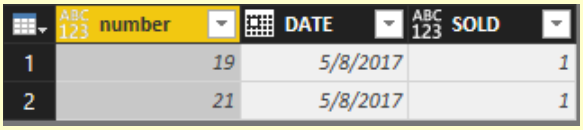


	number	DATE	SOLD
1	4	4/27/2017	1
2	6	4/27/2017	1
3	7	4/27/2017	1
4	8	4/27/2017	1
5	9	4/28/2017	1
6	12	4/28/2017	1
7	14	4/28/2017	1
8	15	4/28/2017	1
9	17	4/28/2017	1

Figure 6.7–12
Transformed PERIOD1 Table

	number	DATE	SOLD
1	18	5/1/2017	1

Figure 6.7–13
Transformed PERIOD2 Table



	number	DATE	SOLD
1	19	5/8/2017	1
2	21	5/8/2017	1

Figure 6.7–14
Transformed PERIOD3 Table

Make sure that all four tables have the same data structure: number of columns, name of columns (case sensitive), and data type of columns.

The next step is to combine the four PERIOD tables together. The query editor makes this easy with the "Append query" option, as shown in figure 6.7–15.

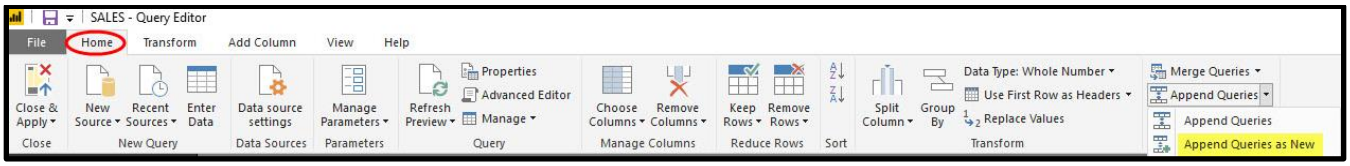


Figure 6.7–15
Append Queries: Combining Tables

Select the "Append queries as new" option. The following window will appear (figure 6.7–16).

Figure 6.7–16
Append Queries: Interface (Two Tables)

Select the option "Three or more tables." The window will then change as follows (figure 6.7–17):

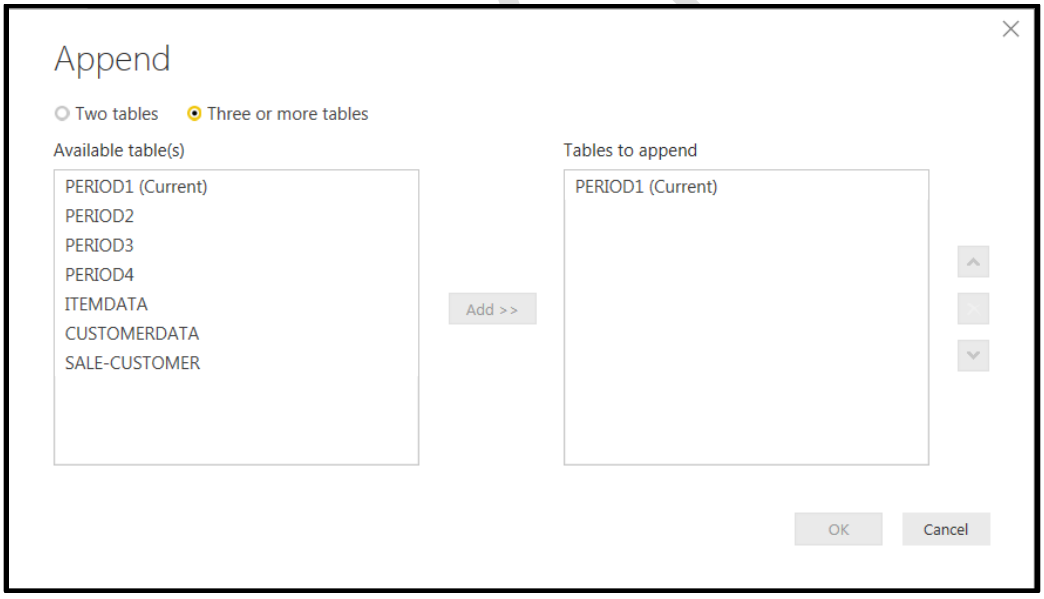


Figure 6.7–17
Append Queries: Interface (Three or More Tables)

Move the PERIOD2, PERIOD3, and PERIOD4 tables from the left to the right column (using the ADD>> button) and click OK. Figure 6.7–18 below shows the new table. Rename the new query SALES. The resulting data set (1) allows us to determine how many items we have sold thus far, (2) lets us determine how many items were sold in a specific period, and (3) makes it easy to connect the sales information with the other tables.

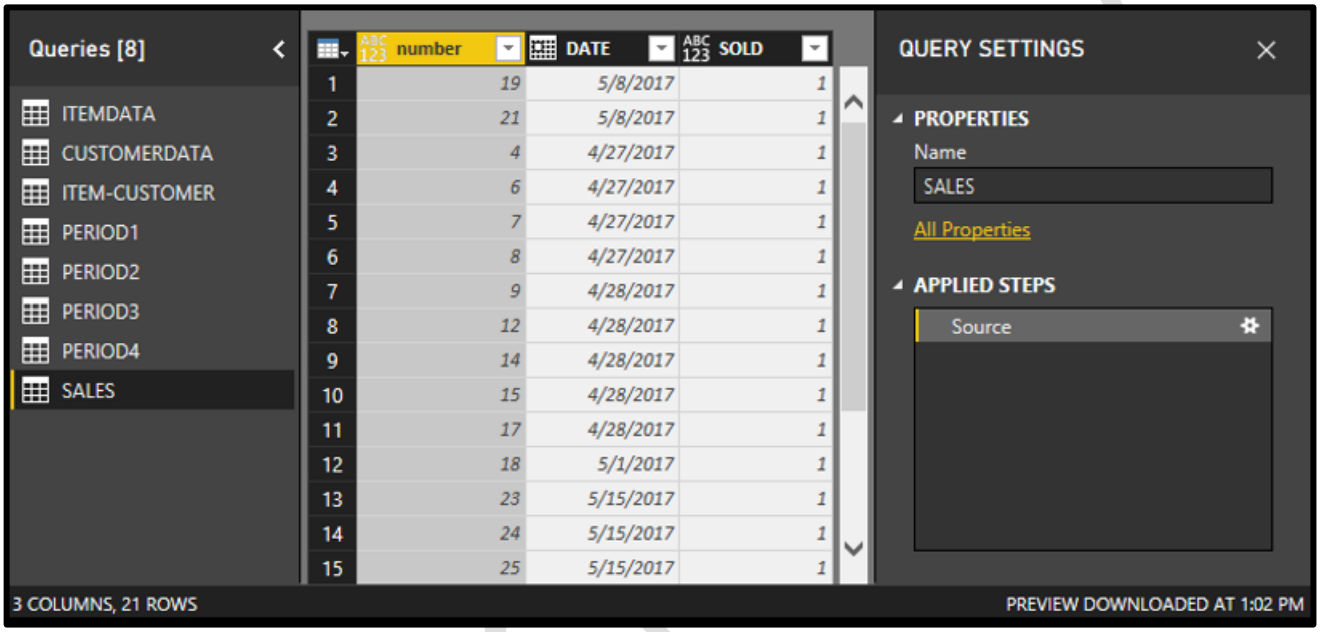


Figure 6.7–18
The Integrated SALES Table

Close the query editor, save the changes you made (i.e., apply changes), and go to the Data view.

Rename the ITEMDATA table as ITEM, and rename the CUSTOMERDATA table as CUSTOMER.

The final step consists of putting all the pieces together—i.e., creating an integrated data set—as shown in figure 6.7.19. The data model integrates:

- what is being sold (ITEM)
- when was it sold (SALES)
- to whom was it sold (ITEM-CUSTOMER, CUSTOMER).

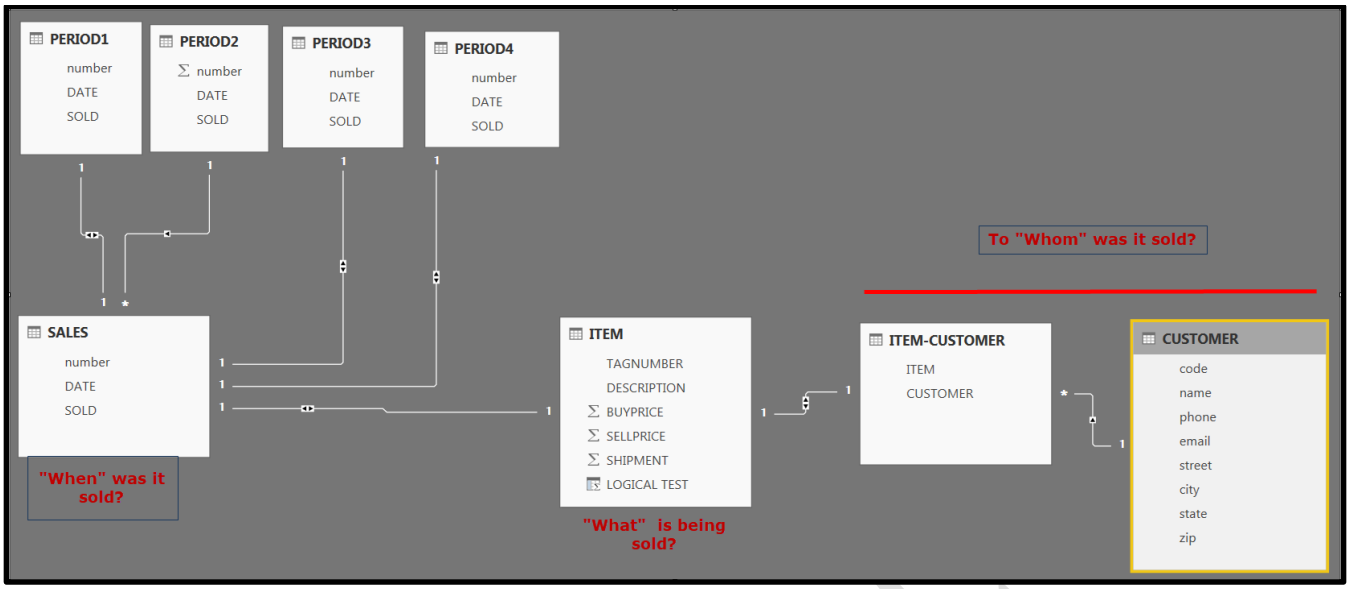


Figure 6.7–19
The Data Model

The data model further shows how the SALES table integrates the different PERIOD tables.

Power BI creates a first draft of the data model. You will find that some of the links (the lines between tables) are already there; others will need to be deleted or added, as follows.

- **Delete a link**: click on the link, right-click, and select Delete.
- **Add a Link**: a link is created by connecting two fields, by dragging one field to the other. You should only link fields that contain the same data and have the same data types.

Click on "Manage relationships" (figure 6.7–20) in the Relationship view (Home tab) to see a more detailed specification of the links.

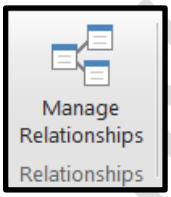


Figure 6.7–20
The "Manage Relationships" Button

A window with detailed specifications will then appear (figure 6.7–21). Among other things, this window will show you the specific fields that are being used for each link. For example, the first entry shows you that a link exists between the CUSTOMER field in the ITEM-CUSTOMER table and the Code field in the CUSTOMER table.

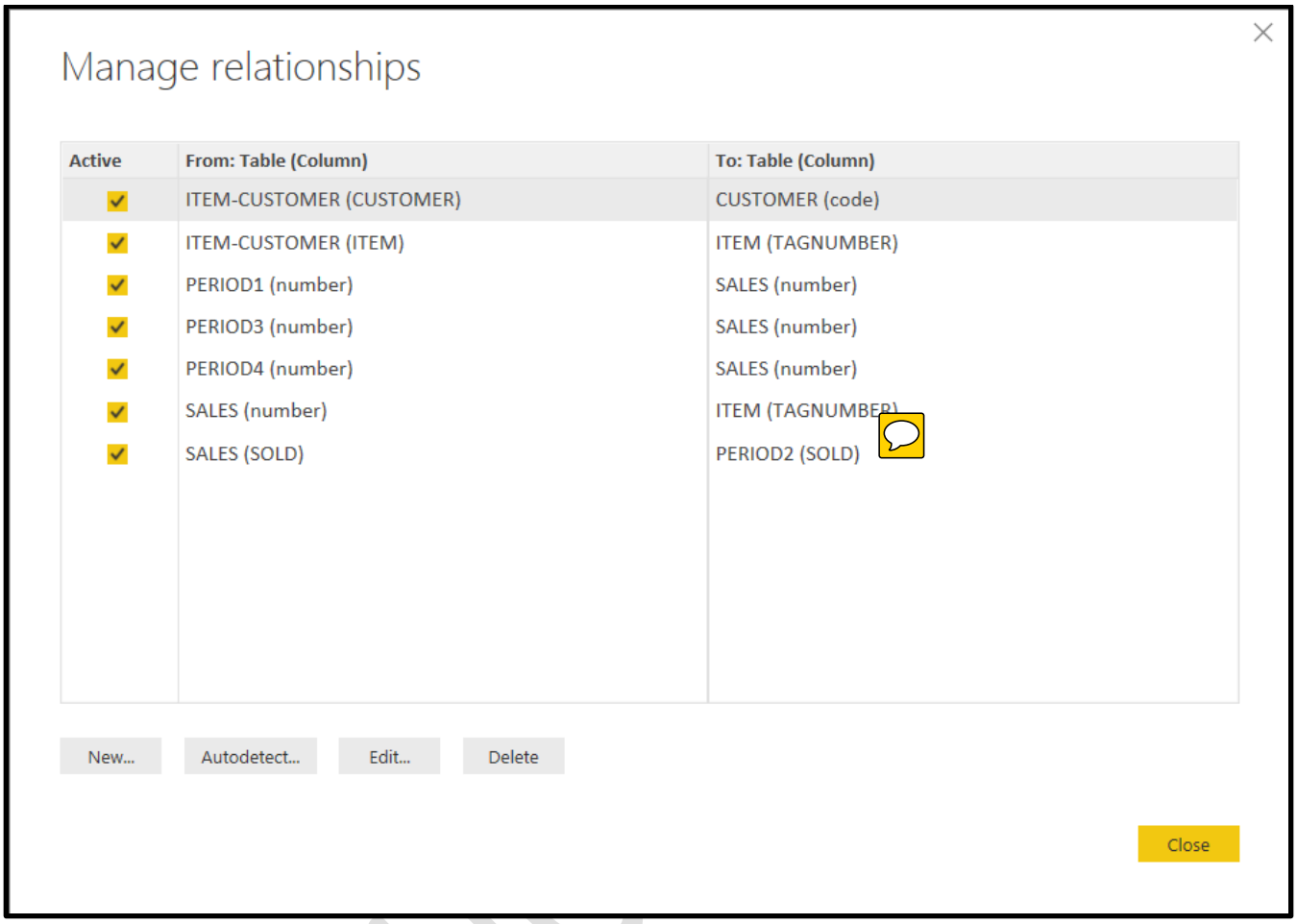


Figure 6.7–21
The "Manage Relationship" Window

Again, it should be noted that the data model shown in figure 6.2–1 (and thus 6.7–19) does not follow the recommended star schema. As mentioned above, the specification and implementation of such schemas is a more advanced topic that is beyond the scope of this book.

ASSIGNMENT 6.7-A2

Develop a few dashboards using the integrated sales data set developed above.

CHAPTER 7
CASE STUDY

Learning Objective

Apply what you have learned in this book to an actual data set.

7.1 THE DATA SET (DATA DISCOVERY)

The purpose of this chapter is to apply what you have learned in this book to an actual data set. Thousands of free data sets are available, many of which are made available by the US and other governments. For this case study we will use licensing data made available by the state of Delaware.[33]

Go to the Delaware Open Data Portal (https://data.delaware.gov/) and search for "Professional and occupational licensing," as shown in figure 7.1–1 below.

Figure 7.1–1
The Delaware Open Data Portal

Next, click on the link shown in figure 7.1.2.

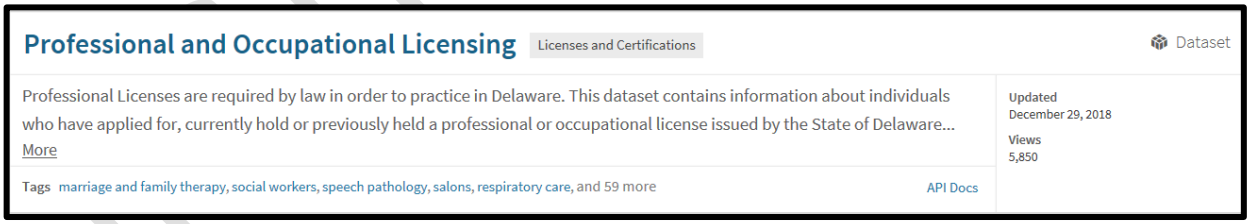


Figure 7.1–2
The Professional and Occupational Licensing Link

The following screen will then appear (figure 7.1.3):

[33] I have chosen Delaware because that's where I live and work, but many other states and countries offer similar data portals.

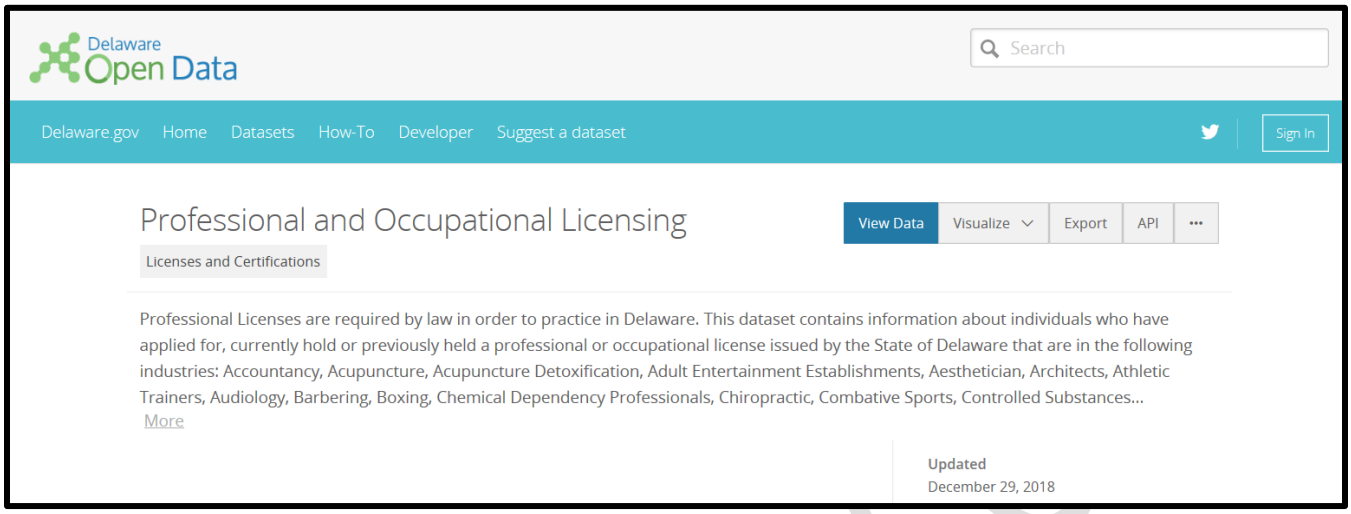


Figure 7.1–3
The Data Set Interface

As shown in figure 7.1–4, different ways are provided to access and query the data, one of which is the use of filters online (the "View data" option). We will use an "OData" connection, which will build an active link to the data set. This connection implies that by clicking the refresh command in Power BI, we will have access to the most recent data. Click on "Access data via OData."

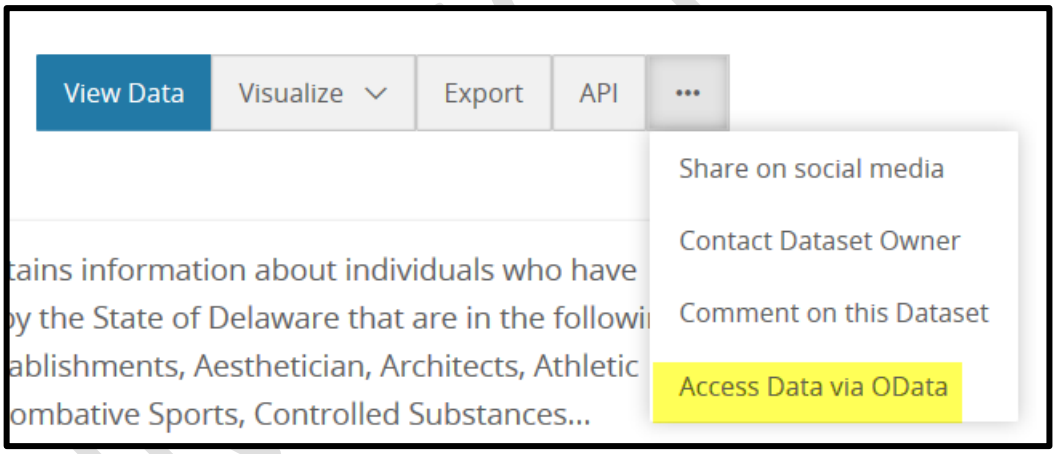


Figure 7.1–4
The Access Data via OData Option

An OData "endpoint" will be provided to you (see figure 7.1–5). We will use this endpoint to build an active link to the data set.

Figure 7.1–5
The OData Endpoint

7.2 DATA EXTRACTION AND ORGANIZATION (COLLECTION)

Next, create a new Power BI project and name it LICENSEINFO. Click on "Get data" and then select "OData feed" (see figure 7.2–1 below). Click on Connect.

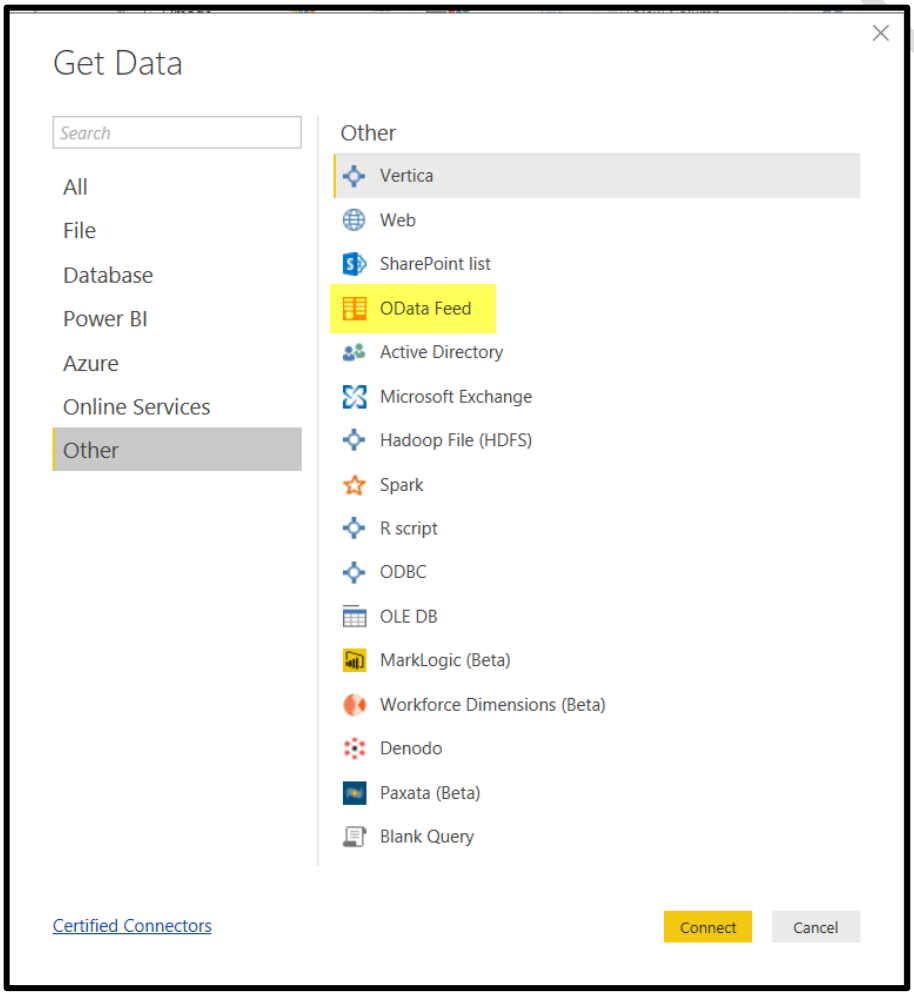


Figure 7.2–1
Selecting the "OData Feed" Connection

The following “OData feed” interface will appear (see figure 7.2–2).

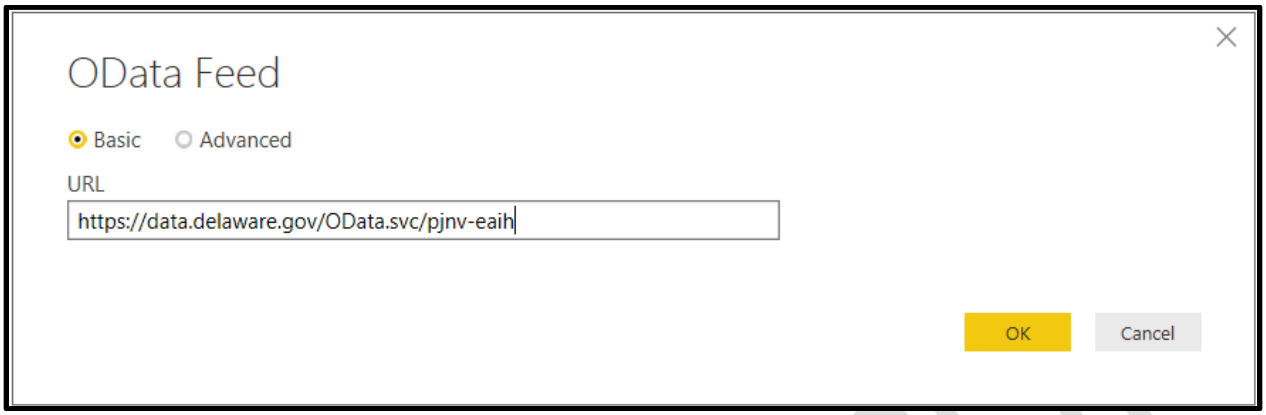


Figure 7.2–2
The OData Feed Interface

Copy the “OData endpoint” and click OK.[34] Congratulations! You have now built an active link to the data set. (Yes, it is that easy ☺!) The website provides the data dictionary shown in table 7.2–1.

[34] This is a large data set, and downloading it requires some time.

Table 7.2–1
Data Dictionary for the "Licensing" Data Set

Column Name / Field Name	Description
last_name	The last name of the individual who has applied for, currently holds, or has held a professional or occupational license issued by the State of Delaware.
first_name	The first name of the individual who has applied for, currently holds, or has held a professional or occupational license issued by the State of Delaware.
combined_name	The full name of the individual who has applied for, currently holds, or has held a professional or occupational license issued by the State of Delaware.
license_no	The license number issued by the State of Delaware for this individual and the specific professional/occupational license.
profession_id	Professional or occupational category of the license
license_type	Specific license or permit type within the professional or occupational category. May be a specific event for certain types of licenses (ie. Bingo event, boxing match).
issue_date	The date that the license or permit was initially issued. The year of 1900 was used when the original issue date was unknown.
expiration_date	The date that the license or permit expires. If this field is blank, the application may be pending, application was abandoned, or the license/permit

Table 7.2–1 (Continued)
Data Dictionary for the "Licensing" Data Set

	does not expire, such as a CPA Certificate.
disciplinary_action	Disciplinary actions exist for this license/permit. Valid values are Y=Yes, N=No.
license_status	The current status of this specific license/permit. (ie. Active, Expired, Lapsed, Pending)
Count	This column has been added strictly to be able to count records for summaries and visualizations. The value will always be "1".

The data set appears to be clean, and it would be fine to just load it. I do suggest that you do the following, however:

1. Rename some of the fields, since they will be used in the dashboard. The names I suggest are shown below (table 7.2–2).
2. Delete the last column, since it doesn't appear to be useful for any analysis.
3. For the Issued and Expiration fields, change the data type from Date/time to Date.
4. Finally, change the name of the query "Query1" to "LicensingData."

Table 7.2–2
Revised Field Names

NEW FIELD NAME	OLD FIELD NAME
ID	__id
LAST	last_name
FIRST	first_name
NAME	combined_name
NO	license_no
PROFESSION	profession_id
LICENSE	license_type
ISSUED	issue_date
EXPIRATION	expiration_date
DISCIPLINARY-ACTION	disciplinary_action
STATUS	license_status

7.3 DATA ENRICHMENT

Analysis will primarily focus on the question, "How many licenses?" This information can then be further sliced and diced by profession, licensing type, period, disciplinary action, or other traits. We will therefore define a measure that determines the number of licenses and name it Number. The DAX formula for this measure is:

DAX DEFINITIONS

TABLE: LICENSINGDATA
NUMBER = COUNT(LICENSINGDATA[ID])

7.4 DATA ANALYSIS

Using the enriched data set (the number measure), you can now start to develop dashboards for analysis purposes. Figure 7.4–1 shows one possible dashboard. Three slicers (in pink) are used for interactive analysis: profession, license type, and license status. Any combination of these three fields can be selected. In figure 7.4–1, Accountancy has been chosen as the profession, and CPA certificate as the license type.

The "Number of licenses" card (in blue) shows the total number of licenses based on the selections made in the slicers. The bottom part shows detailed information about the selected licensees (in green). The pie chart shows the number of licenses issued over time.

See the LICENSEINFOSOLUTION.PBIX file in your DropBox folder for a full specification of the dashboard and visualizations shown in figure 7.4–1.

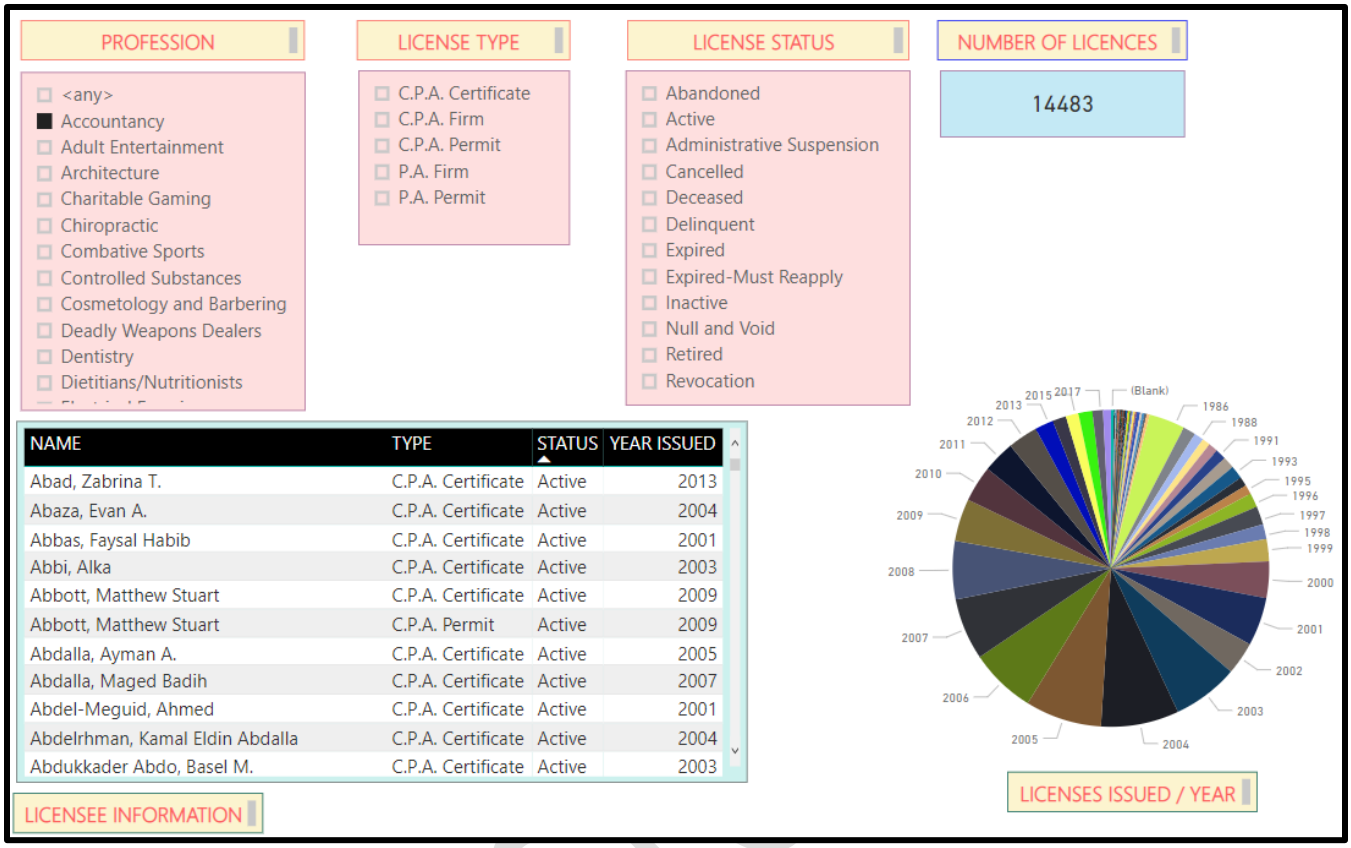


Figure 7.4–1
Dashboard for "Licensing Data" Analysis

ASSIGNMENT

Use the LICENSING.PBIX data set to create a few additional reports.

GLOSSARY

Analytical Database
An integrated data set developed for analysis purposes.

Analytics
Analyzing data to generate insights in support of problem solving and decision making.

Big Data
Massive continuous streams of complex data for which the reliability is not always clear.

Data Analysis Expressions (DAX)
A formula language developed by Microsoft that consists of a set of functions and operators that can be used for defining tests and rich information models.

Data Process Chain
The prototypical sequence of steps necessary to execute a big data project: data discovery, data collection, information modeling, analytics, and problem solving.

Dashboard
An interactive and integrated display of visualizations for reporting, monitoring, and exploring data.

Data Collection
The process of extracting, transforming, and loading data. Also known as the ETL process.

Data Connector
An easy-to-use program for connecting to a data source and transferring data.

Data Discovery
The process of exploring what data are available, as well as the data's relevance and affordability.

Data Model
A formal definition of the structure of a data set: what information is captured by the tables, what columns are part of the table, how the different tables are connected to one another, etc.

Data Profiling
The process of understanding data, assessing data quality, and providing guidance about any necessary transformations.

Extract-Transform-Load (ETL)
The process of transferring data from one or more sources (extraction); profiling, cleaning, restructuring, and integrating the data (transform); and loading the data into the analytical database (loading).

Information Modeling
Calculating the information required for analysis by means of formulas.

Instance
A specific object. For example, Vincent Van Gogh is an instance of a painter.

Measure
A calculation by a formula that can be used for analysis purposes as part of visualizations/dashboards.

Power BI
Microsoft's self-service business intelligence (SSBI) tool

Self-Service Business Intelligence (SSBI) Software
Easy-to-use software that provides powerful support for profiling, extracting, cleaning, integrating, and analyzing data. Examples of SSBI software include Power BI, Qlik, Spotfire, and Tableau.

Star Schema
A data structure that organizes data in terms of fact and dimension tables; the star schema is easy to understand and efficient to process.

Visualization
A type of image or chart that can be used to present data in a specific way; examples include bar charts, maps, tables, and slicers.

REFERENCES

Geerts, Guido L., *An Introduction to Big Data* (Author, 2017). Available at www.bigdatavillage.com/books.

Geerts, Guido L., and Kinsun Tam. "KaDo: An Advanced Enterprise Modeling, Database Design, Database Implementation, and Information Retrieval Case for the Accounting Information Systems Class." *Journal of Information Systems* 22, 2 (2008): 141–50.

Microsoft. *Power BI Documentation*, 2018. Available at https://docs.microsoft.com/en-us/power-bi/.

Microsoft. *Best Design Practices for Reports and Visuals*, 2017. Available at https://docs.microsoft.com/en-us/power-bi/power-bi-visualization-best-practices.

Press, Gil. "Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Surveys Says." *Forbes*, March 23, 2016. Available at https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#ef746d66f637.

APPENDIX 1
EXTRACTING "CUSTOMERDATA" FROM TEXT/CSV FILES

If you don't have Microsoft Access available on your computer or if you get an error message when extracting data from a Microsoft Access database, I have created two text/CSV files from which you can extract the same data. A few additional instructions are provided below. You will need two files: CUSTOMERDATA.CSV and ITEM-CUSTOMER.CSV. Both are described below.

CUSTOMERDATA.CSV

Let's start with the CUSTOMERDATA.CSV file. Open "Get data" and click on the CSV data connector (figure A1–1). The dialog box in figure A1–2 will appear. Select the CUSTOMERDATA.CSV file and click Open. The dialog box in figure A1–3 appears, which gives you control over how the data will be loaded.

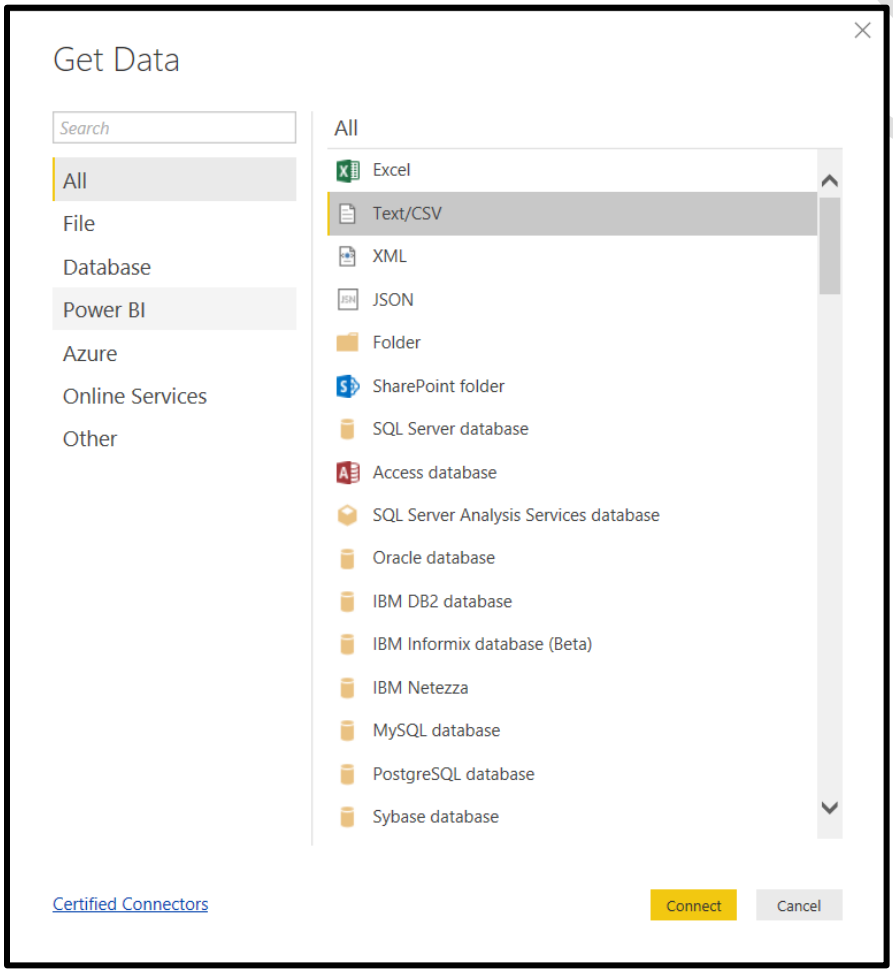


Figure A1–1
"Text/CSV File" Data Connector

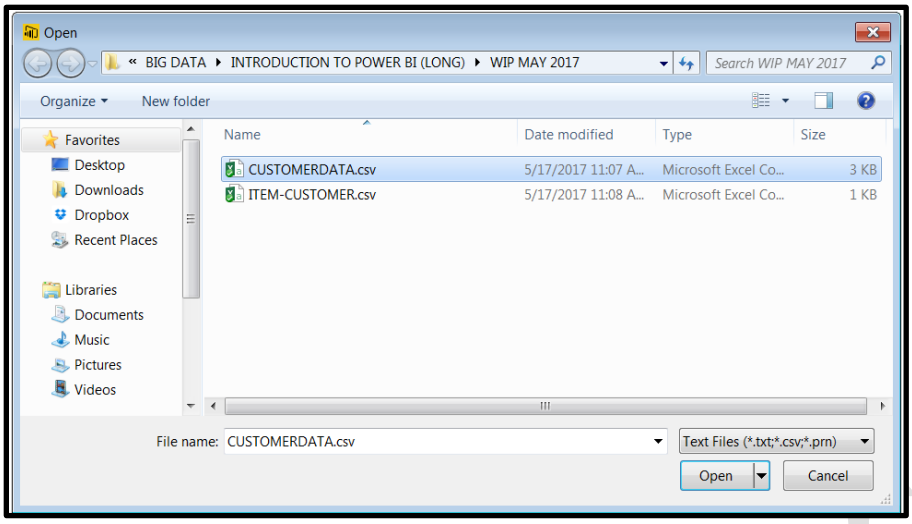


Figure A1–2
Selecting the "CustomerData.csv" File

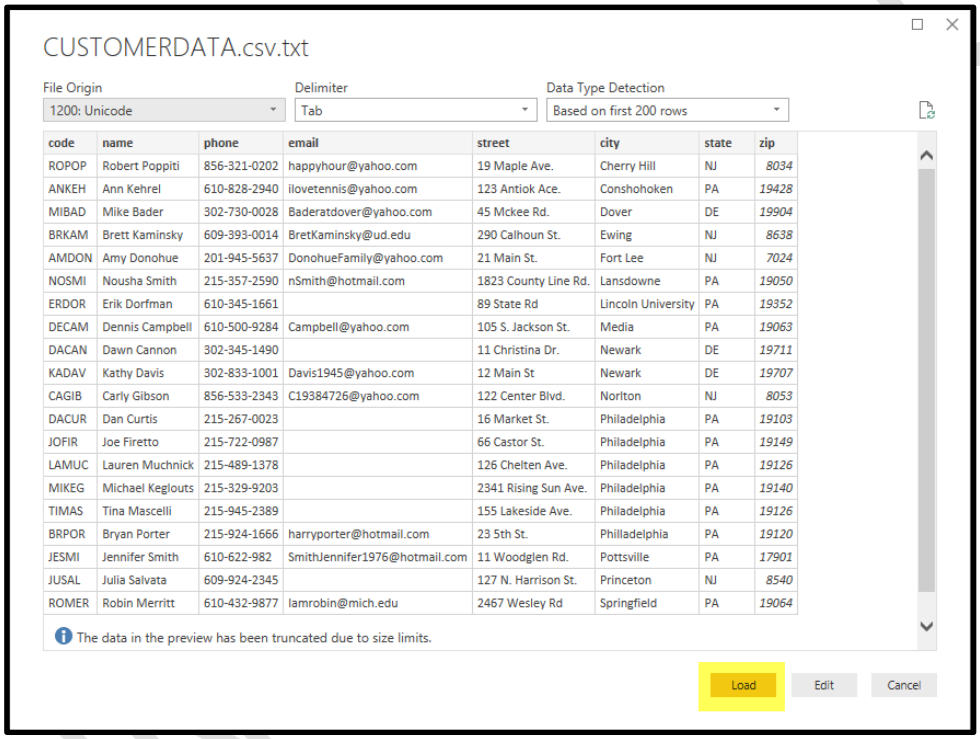


Figure A1–3
Controlling How Data from a Text/CSV File Are Loaded

Do not change any of the control options ("delimiter" and "Data type detection"), and click Load. This will load the data into Power BI's data model; a CUSTOMERDATA table is then created. This is the same table as the one created by extracting data from the Access database.

ITEM-CUSTOMER.CSV
Repeat the same steps for the ITEM-CUSTOMER.CSV file.