

K2's EXCEL TABLES AND DATA MODELS – EFFICIENTLY MANAGING, ANALYZING, AND REPORTING YOUR DATA

CONTINUING PROFESSIONAL EDUCATION SERIES



To access demonstration files used in this seminar, please visit www.k2e.com, log-in in the Attendee Login section, and use the following code to access the materials:

ETD

K2 ENTERPRISES
1250 SW RAILROAD AVENUE, SUITE 240A ♦ HAMMOND, LA 70403

Copyright © April 6, 2016 by K2 Enterprises

All rights reserved. No part of this publication may be reproduced or transmitted in any form without the express written consent of K2 Enterprises, 1250 SW Railroad Avenue, Suite 240A, Hammond, LA 70403, telephone 985.542.9390. Requests to reproduce or transmit may be e-mailed to caron@k2e.com. Further information may be obtained on the web at www.k2e.com.

The use of trade names and trademarks in these materials are not intended to convey endorsement of these vendors or products. All trade names and trademarks used in these materials are the property of their respective owners. Any abbreviations used herein are solely for the reader's convenience and are not intended to compromise any trademarks.

The statements in these materials about specific companies or specific hardware and software products should in no way be misconstrued as statements of fact but rather are opinions of the authors and K2 Enterprises. We take great care in providing opinions that we think will be useful in helping our participants make informed decisions. K2 Enterprises reserves the right to independently evaluate the merits of the companies and of the hardware and software products that are referenced in our seminars and conference presentations.

K2 Enterprises makes no representations or warranty with respect to the contents of these materials and disclaims any implied warranties of merchantability or fitness for any particular purpose. The contents of these materials are subject to change without notice.

MISSION STATEMENT: K2's goal is to produce and deliver the highest quality technology seminars and conferences available to business professionals. We work cooperatively with professional organizations, such as state CPA societies and associations of Chartered Accountants, and vendors of technology products. K2 also provides consulting services and advice on technology. We make every effort to maintain a high level of integrity, family values, and friendship among all involved.

CONTRIBUTORS: The shareholders, associates and staff of K2 Enterprises contributed in the production of these materials.

Photograph of K2 Mountain by Adrian Burgess

Table of Contents

INTRODUCTION	IV
THREE RULES OF THIS SEMINAR	IV
COURSE DEVELOPMENT	IV
ABOUT K2 ENTERPRISES	V
THE K2 CURRICULUM	VI
Two-Day Seminar	VI
Full-Day Seminars	VI
Half-Day Seminars	VI
ON-SITE TRAINING	VII
WEB-BASED TRAINING	VII
SELF-STUDY COURSES	VIII
K2 INTERNET SITES	VIII
CHAPTER 1 – INTRODUCING TABLES.....	1
LEARNING OBJECTIVES	1
ADVANTAGES OF USING TABLES.....	1
<i>Quick and Easy Application of Formatting</i>	2
<i>Automatic Resizing Data Ranges</i>	3
<i>Structured Reference Formulas</i>	5
<i>Self-Replicating Formulas</i>	6
<i>Summary Rows</i>	7
CREATING TABLES	8
Data Considerations before Creating a Table	8
<i>Creating Tables from the Ribbon</i>	9
<i>Creating Tables with a Keyboard Shortcut</i>	9
<i>Creating Tables from External Data Sources</i>	10
NAMING AND RENAMING TABLES	11
RESIZING TABLES MANUALLY	12
FORMATTING DATA WITH TABLES	13
<i>Establishing a Default Table Style</i>	13
<i>Creating a New Table Style</i>	14
<i>Converting a Table to a Range of Data</i>	16
CHAPTER 2 – SORTING AND FILTERING IN TABLES.....	17
LEARNING OBJECTIVES	17
SORTING TABLE DATA	17
<i>Basic Sorting</i>	17
<i>Multi-field Sorting</i>	18
<i>Case-sensitive Sorting</i>	19
<i>Sorting Based on Cell Formats</i>	19
FILTERING TABLE DATA	22
<i>Filtering Tables on Single Fields</i>	23
Text Filters.....	23
Dynamic Date Filters	24
Number Filters	25
Search Filters.....	26
Using Slicer Filters on Table Data (Excel 2013/2016 only)	27
<i>Filtering on Multiple Fields</i>	29
Advanced Filtering.....	29
Filtering for Top and Bottom x Entries	29
Filtering Data on Calculated Values	30
Using the Advanced Filter Function	31
CHAPTER 3 – BUILDING STRUCTURED REFERENCE FORMULAS	35

LEARNING OBJECTIVES	35
INTRODUCING STRUCTURED REFERENCE FORMULAS	35
COMPONENTS OF A STRUCTURED REFERENCE FORMULA	36
<i>Table Name</i>	36
<i>Column Specifiers</i>	36
<i>Special Item Specifiers</i>	36
<i>Table Specifier</i>	37
SAMPLE STRUCTURED REFERENCE FORMULAS	38
SPECIAL SITUATIONS FOR STRUCTURED REFERENCE FORMULAS	39
<i>Fully Qualified and Unqualified Structured References</i>	39
<i>Formula Rewrite When Converting Tables to Ranges</i>	40
<i>Absolute Versus Relative Referencing</i>	40
<i>Table Column Headers and Structured Reference Formulas</i>	40
Table Column Headers as Text Strings	40
Special Characters in Column Headers	42
CHAPTER 4 – USING TABLES AS DATA SOURCES	43
LEARNING OBJECTIVES	43
BUILDING CHARTS FROM TABLES	43
<i>Filtering Charts by Filtering Tables</i>	46
<i>Multiple Charts from the Same Table</i>	47
USING TABLES AS SOURCES FOR PIVOT TABLES	47
<i>Building a PivotTable from Multiple Tables</i>	48
Consolidating Multiple Tables into One	49
Using the Data Models and the Power Pivot Add-in to Consolidate Table Data for PivotTables	51
USING TABLES AS SOURCES FOR PIVOT CHARTS	51
CHAPTER 5 – ADVANCED TABLE TECHNIQUES	55
LEARNING OBJECTIVES	55
CREATING FORMULAS THAT REFER TO TABLE DATA	55
<i>Building VLOOKUPS on Table Data</i>	56
VLOOKUP Querying Table Data	58
VLOOKUP inside a Table	59
REFERENCING TABLES ACROSS MULTIPLE WORKSHEETS AND WORKBOOKS	60
<i>Linked Workbooks and Structured References</i>	61
SELECTING ROWS AND COLUMNS IN TABLES	61
CUSTOM VIEWS UNAVAILABLE WHEN WORKING WITH TABLES	61
EXCEL DISABLES SHARED WORKBOOK FUNCTIONALITY WHEN WORKING WITH TABLES	61
DISABLING SPECIFIC TABLE FEATURES	61
USING EXCEL'S FUZZY LOOKUP ADD-IN	62
<i>Download the Add-in</i>	62
<i>Creating a Fuzzy Lookup</i>	63
<i>Fuzzy Lookup Summary</i>	66
CHAPTER 6 – INTRODUCING EXCEL'S DATA MODEL FEATURE	67
LEARNING OBJECTIVES	67
WHAT IS A DATA MODEL?	67
<i>Advantages of Working with Data Models</i>	68
CREATING SIMPLE DATA MODELS FROM EXCEL TABLES	70
<i>Data Model Relationships</i>	75
<i>Simple Relationships in Data Models</i>	76
A SUMMARY EXAMPLE OF SIMPLE DATA MODELS	77
CHAPTER 7 – ADVANCED DATA MODEL TOOLS AND TECHNIQUES	81
LEARNING OBJECTIVES	81
CREATING DATA MODELS FROM EXTERNAL DATA SOURCES	81

BUILDING AND ADDING TO DATA MODELS WITH POWER QUERY	85
<i>Using Power Pivot to Query Data into a Data Model</i>	90
WORKING WITH DATA IN A DATA MODEL.....	91
<i>Managing Data and Relationships in a Data Model</i>	91
<i>Building Formulas with Data Analysis Expressions</i>	95
Creating Calculated Columns with DAX	97
Creating Measures in a PivotTable using DAX	100
<i>Creating Hierarchies to Manipulate Reports More Quickly</i>	102
<i>Using Perspectives to Control the Amount of Data Available</i>	107
CHAPTER 8 - CREATING REPORTS AND SUMMARIES FROM DATA MODELS	111
LEARNING OBJECTIVES	111
CONSTRUCT PIVOTTABLES AND PIVOTCHARTS THAT USE DATA MODELS AS DATA SOURCES	111
<i>Creating a PivotTable Report in Power Pivot</i>	116
CREATE INTERACTIVE REPORTS USING POWER VIEW	119
<i>Organizing Your Data for Power View</i>	119
<i>Creating a Simple Power View Dashboard</i>	120
<i>Adding Visualization to a Power View Dashboard</i>	121
<i>Adding Additional Elements to the Dashboard</i>	122
<i>Filtering a Power View Dashboard</i>	123
UTILIZE DATA MODELS IN CONJUNCTION WITH 3D MAPS	125

Introduction

Three Rules of this Seminar

1. There Are No Dumb Questions.

Please ask questions if you need further explanation. We welcome your questions and will do our best to provide accurate and meaningful answers.

2. Obtain the Information For Which You Came.

If you have a specific interest in a topic, please let us know. If your topic is not already part of the course materials, we will do our best to accommodate your request. Our instructors are anxious to meet your needs, but, if you do not ask questions (rule #1), they cannot give you the information that you want.

3. Share and Share Alike.

Our instructor is here to share all possible information related to the course. We hope that you will be willing to do the same. If you have some great tips or experiences, please share them with us and with the other participants.

Course Development

The shareholders, associates, and staff of K2 Enterprises developed this course and the related course materials. Our goal is to deliver top quality educational courses that help you better use and understand the tools and concepts that are the focus of this session.

We designed our course presentation style for busy professionals. Through numerous short case studies and feature reviews, we strive to deliver concise information in a context that is easy to understand and easy to absorb.

To maximize the use of your time and the benefits of attending this course, we have worked hard to filter out trivial and obscure features. Our goal is not to cover every keystroke and menu option but to provide a fast-paced course that focuses on state-of-the-art information technology and what it can do for you and your staff.

About K2 Enterprises

Mission Statement

K2's goal is to produce and to deliver the highest quality technology seminars and conferences available to business professionals. We work cooperatively with professional organizations such as state CPA societies, associations of Chartered Accountants, and vendors of technology products. K2 also provides consulting services and advice on technology. We make every effort to maintain a high level of integrity, family values, and friendship among all involved.

K2 Enterprises is named after the second highest mountain in the world. (The summit of K2 is 28,251 feet above mean sea level; Mount Everest is 29,026 feet above mean sea level.) The shareholders and associates of K2 Enterprises are listed in the following table.

Shareholders

Randolph P. Johnston, MCS	Hutchinson, KS	<i>randy@k2e.com</i>
Val D. Steed, CPA.CITP, MA	Centerville, UT	<i>val@k2e.com</i>
Thomas G. Stephens, CPA.CITP	Woodstock, GA	<i>tommy@k2e.com</i>

Associates

John K. Keegan, CPA.CITP	Oakhurst, NJ	<i>john@k2e.com</i>
Lawrence A. McClelland, MBA, JD	Carriere, MS	<i>mac@k2e.com</i>
Steven M. Phelan, CPA.CITP	Oklahoma City, OK	<i>steven@k2e.com</i>
Brian F. Tankersley, CPA.CITP	Knoxville, TN	<i>brian@k2e.com</i>
Stephen M. Yoss, Jr. CPA	Claremont, CA	<i>steve.yoss@k2e.com</i>

For more than thirty years, our team members have collectively delivered more than 15,000 national and international courses and conference sessions on the subject of technology. To help us stay on top of technology, our team members read numerous technology trade magazines, attend many technology conferences, and talk with numerous technology companies each year to obtain the latest information to include in our courses. In addition, we personally experience everything we teach and recommend to our audiences. Because our shareholders and associates reside in different states, the use of technology is a necessity for the success and survival of our business.

The K2 Curriculum

Two-Day Seminar

- K2's Excel Boot Camp – Two Days of Intensive Excel Training

Full-Day Seminars

- K2's Advanced Excel
- K2's Budgeting and Forecasting Tools and Techniques
- K2's Business Continuity – Best Practices for Managing the Risks
- K2's Excel-Based Dashboards
- K2's Excel Best Practices
- K2's Excel Financial Reporting and Analysis
- K2's Excel PivotTables for Accountants
- K2's Excel Tables & Data Models – Efficiently Managing, Analyzing, & Reporting Your Data **NEW FOR 2016!**
- K2's Excel Tips, Tricks, and Techniques for Accountants
- K2's QuickBooks for Accountants
- K2's Small Business Internal Controls, Security, and Fraud Prevention and Detection
- K2's Technology for CPAs – Don't Get Left Behind

Half-Day Seminars

- K2's Advanced Excel Reporting – Best Practices, Tools, and Techniques
- K2's Advanced QuickBooks Tips and Techniques
- K2's Do It Yourself Business Intelligence
- K2's Introduction to Excel Macros
- K2's Microsoft Office 2016 – Improving Productivity With New Features **NEW FOR 2016!**
- K2's Microsoft Office Everywhere **NEW FOR 2016!**
- K2's QuickBooks Online – Changing the Paradigm of Small Business Accounting
- K2's Securing Your Data – Practical Tools for Protecting Information
- K2's Tech Tools and Gadgets for a More Efficient You!
- K2's Technology Update
- K2's Top PDF Features You Should Know **NEW FOR 2016!**

In addition to the twenty-four courses listed above, K2 Enterprises also produces approximately 22 two-day technology conferences each year. You can find a complete listing of our course offerings, including dates, locations, and course descriptions, on our web site at www.k2e.com/training.



On-Site Training

Working cooperatively with your state CPA organization, we are pleased to offer all of our seminars and conference sessions as on-site training opportunities. On-site training offers companies of all sizes numerous advantages, including the following:

- **Customization.** K2 Enterprises can customize the content of all on-site training events to meet the specific needs of your organization. For example, we can combine selected parts of existing courses into one customized course to meet your specific needs. Likewise, we can develop custom content from the ground up and deliver it to your team members, thereby ensuring that your training objectives are met.
- **Cost savings.** For organizations of all sizes, on-site training can provide significant cost savings when compared to public training venues. In addition to potentially reduced registration fees, these cost savings accrue in the form of reduced travel costs and reducing the amount of time team members spend away from the office.
- **Convenience.** Because you select the date, time, and location of your on-site training event, on-site training is more convenient for you and your team members than traditional training options. Many organizations schedule on-site training in conjunction with staff meetings, retreats, and customer/client events to leverage further the value of all participants' time.

Customization, cost savings, and convenience – the three C's of “training to go.” For more information on how you and your organization can experience these benefits or to schedule an on-site training event, please visit <http://www.k2e.com/training/on-site-training> or contact cheryl@k2e.com.



Web-Based Training

K2 Enterprises offers web-based CPE to accounting and business professionals throughout the United States. K2's team of award-winning instructors delivers two-hour, four-hour, and eight-hour webinars on numerous technology-focused topics, including Microsoft Excel, QuickBooks, PDF documents, technology trends and strategies, and security. For more information on these webinar offerings or to register, please visit <http://www.k2e.com/training/web-based-training>.



Self-Study Courses

K2 Enterprises provides self-study training opportunities. Self-study courses offer you the advantage of receiving top-quality training at a time and location that fits into your busy schedule. You can start a course today and complete it in the future, if that is necessary to help you achieve the work-life balance you desire. For more information on self-study offerings or to register, please visit <http://www.k2e.com/training/self-study>.



K2 Internet Sites

As a service to our attendees, K2 Enterprises maintains multiple Internet sites that contain information and links that are relevant to the needs of CPAs and other business professionals and that enrich your experience during and after our seminars.

The K2 web sites contain details on the topics that we present, tips on using the software and hardware that we cover, and information on the products that we use daily and can recommend with confidence. Since the hardcopy materials provided in our seminars often age quickly, you will find additional supporting content and the most up-to-date information on our web sites. Our pages contain links to the web sites of some of the top accounting and content management software solutions as well as to other web sites of interest and even to a few sites that are just fun places to visit. We invite you to visit our web sites often. If you have any comments, questions, or suggestions regarding our web pages, please send an email to webmaster@k2e.com.

www.k2e.com

This is the primary web site for **K2 Enterprises**. Here you can access information regarding CPE opportunities, including seminars, conferences, on-site training, web-based training, and self-study opportunities. You can also access a library of technology tips and articles, and you can link to K2-related web sites.



www.cpafirmtech.com

CPA Firm Technology contains technology-related information targeted specifically to public accounting firms. Included are product reviews, hardware and operating system discussions, affinity programs for CPA firms, and Cloud computing resources.



www.accountingsoftwareworld.com

Accounting Software World provides information related to accounting and financial management solutions. You can find software and service reviews, product comparisons, white papers, and other material of interest to those seeking information about accounting software and service solutions.



www.totallypaperless.com

For information about document management and paperless process, please visit **Totally Paperless**. This site provides reviews on hardware and software used by all types of businesses in document imaging and management processes.



Introducing Tables

Microsoft introduced tables with the 2007 release of Excel and, in the course of doing so, likely forever changed how many Excel users should work with data in spreadsheets. Though not yet thoroughly understood by most, tables build upon the list functionality introduced in Excel 2003 and provide superior options for storing, sorting, filtering, querying, and managing volumes of data in Excel. In this chapter, you will learn about the fundamentals of working with tables – one of Excel’s best features.

Learning Objectives

Upon completing this chapter, you should be able to:

- List five advantages of working with tables in Excel;
- Identify three approaches to creating tables and utilize each;
- Apply names to tables and rename tables when desired;
- Use table styles to format tables and to create new table styles to meet specific formatting objectives; and
- Convert table data back to an ordinary range of data in Excel.

Advantages of Using Tables

Many Excel users are accustomed to working with large volumes of data in Excel by just simply storing the data in rows and columns and addressing it by traditional row and column addresses; we refer to this method as the *R1C1* approach, where R is row and C is column as in the cell address “B4.” Some users assign *defined names* to the data, and, when they need to address the dataset as a whole, they use the defined name such as *Dataset* instead of an R1C1 addresses such as *B4:G5897*. A smaller number of users build formulas to create *dynamic defined names* so that as the volume of data in the dataset increases or decreases, the dimensions of the dynamic defined name automatically adjust to match the volume of data present. While there are no specific faults with any of these approaches, converting ranges of data into tables and then referencing the table name when needing to reference the data in the tables provides five distinct advantages compared to the three traditional approaches just described.

Quick and Easy Application of Formatting

One of the principal advantages of working with tables is that *when you convert a dataset into a table, Excel automatically formats the data* using one of sixty predefined table styles. If your formatting needs extend beyond those offered by these styles, you can create a new table style to satisfy your specific requirements as described beginning on page 14.

To illustrate, consider the data range pictured in **Figure 1**. It contains daily sales data for six stores for a full year and consumes 8 columns and 367 rows on this worksheet. To format this dataset manually for presentation purposes in a report would be a tedious and time-consuming process.

	A	B	C	D	E	F	G	H
1								
2	Date	Product A	Product B	Product C	Product D	Product E	Product F	Total
3	1/1/2017	664,861.33	375,025.09	534,366.90	490,662.72	563,025.05	502,559.67	3,130,500.76
4	1/2/2017	588,064.21	566,147.60	553,872.75	486,591.19	574,359.61	427,123.63	3,196,158.99
5	1/3/2017	564,533.07	653,110.03	621,220.55	541,478.21	419,819.79	554,492.59	3,354,654.24
6	1/4/2017	463,021.47	477,662.36	558,264.64	599,462.99	502,650.37	680,549.43	3,281,611.26
7	1/5/2017	400,817.04	478,509.20	626,357.35	683,833.47	542,043.32	544,888.83	3,276,449.21
362	12/26/2017	436,883.28	504,467.01	521,166.40	490,381.39	441,118.38	477,056.45	2,871,072.91
363	12/27/2017	553,474.12	453,223.17	590,621.64	427,989.10	486,843.23	629,700.70	3,141,851.96
364	12/28/2017	660,370.25	562,251.70	509,978.20	470,639.58	587,058.58	604,131.71	3,394,430.02
365	12/29/2017	426,050.53	478,276.37	556,128.84	490,376.06	464,307.26	510,288.44	2,925,427.50
366	12/30/2017	541,679.76	641,291.48	567,966.73	514,678.88	465,065.95	522,265.64	3,252,948.44
367	12/31/2017	533,602.33	525,499.43	531,010.37	563,860.01	484,215.66	572,819.29	3,211,007.09
368								

Figure 1 - Unformatted Data Prior to Converting to Table

As shown in **Figure 2**, upon converting the dataset to a table, Excel formats the entire range automatically.

	A	B	C	D	E	F	G	H
1								
2	Date	Product A	Product B	Product C	Product D	Product E	Product F	Total
3	1/1/2017	664,861.33	375,025.09	534,366.90	490,662.72	563,025.05	502,559.67	3,130,500.76
4	1/2/2017	588,064.21	566,147.60	553,872.75	486,591.19	574,359.61	427,123.63	3,196,158.99
5	1/3/2017	564,533.07	653,110.03	621,220.55	541,478.21	419,819.79	554,492.59	3,354,654.24
6	1/4/2017	463,021.47	477,662.36	558,264.64	599,462.99	502,650.37	680,549.43	3,281,611.26
7	1/5/2017	400,817.04	478,509.20	626,357.35	683,833.47	542,043.32	544,888.83	3,276,449.21
362	12/26/2017	436,883.28	504,467.01	521,166.40	490,381.39	441,118.38	477,056.45	2,871,072.91
363	12/27/2017	553,474.12	453,223.17	590,621.64	427,989.10	486,843.23	629,700.70	3,141,851.96
364	12/28/2017	660,370.25	562,251.70	509,978.20	470,639.58	587,058.58	604,131.71	3,394,430.02
365	12/29/2017	426,050.53	478,276.37	556,128.84	490,376.06	464,307.26	510,288.44	2,925,427.50
366	12/30/2017	541,679.76	641,291.48	567,966.73	514,678.88	465,065.95	522,265.64	3,252,948.44
367	12/31/2017	533,602.33	525,499.43	531,010.37	563,860.01	484,215.66	572,819.29	3,211,007.09
368								

Figure 2 - Data Converted to a Table Formatted Automatically

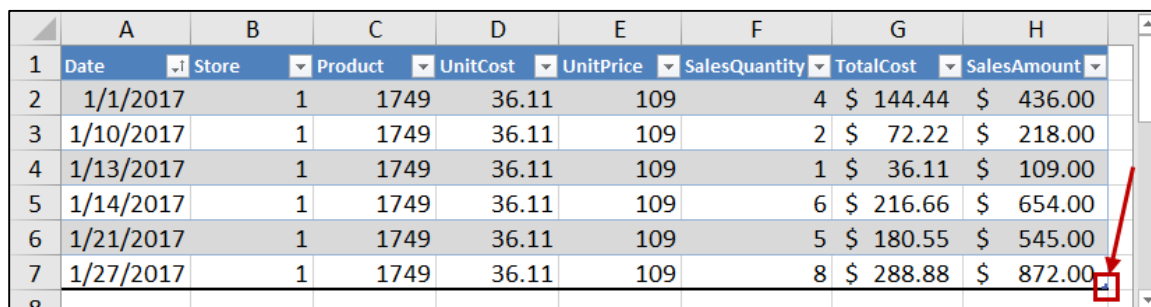
One of the unique benefits of using tables to apply formatting is that the formatting is *dynamic*. By this, we mean that as the volume of data in the table changes, the formats automatically adjust to match the volume of data present. Thus, returning to Figure 2, if you insert additional rows into the table, Excel automatically adjusts the shading effect on every other row so that it applies correctly throughout the table.

Automatic Resizing Data Ranges

Tables provide Excel users with *automatically resizing data ranges*. That is, the data included in the table automatically adjusts when you add one or more rows immediately below the table or one or more columns immediately to the right of the table. Likewise, if you delete either rows or columns from the table, the dimensions of the table automatically contract to reflect a smaller volume of data.

As the volume of data in the table changes, all calculations that use table data update to reflect the new dimensions of the table. Thus, if you have a formula outside the table that adds all the data in a single column of the table, and you then append one hundred new rows of data to the table, your formula automatically includes the additional data without any intervention on your part. Likewise, any object in Excel such as a chart that uses a table as its data source will also automatically update as the volume of data in the table changes.

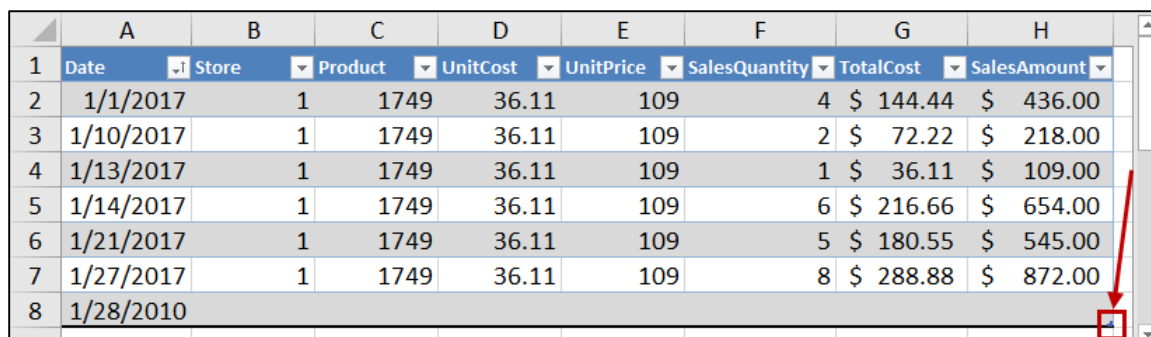
To illustrate, examine the data contained in the table pictured in **Figure 3**. This table contains sales data for a specific product. The small, encircled handle in the lower right corner of cell H7 indicates the outer boundary of the table.



	A	B	C	D	E	F	G	H
1	Date	Store	Product	UnitCost	UnitPrice	SalesQuantity	TotalCost	SalesAmount
2	1/1/2017	1	1749	36.11	109	4	\$ 144.44	\$ 436.00
3	1/10/2017	1	1749	36.11	109	2	\$ 72.22	\$ 218.00
4	1/13/2017	1	1749	36.11	109	1	\$ 36.11	\$ 109.00
5	1/14/2017	1	1749	36.11	109	6	\$ 216.66	\$ 654.00
6	1/21/2017	1	1749	36.11	109	5	\$ 180.55	\$ 545.00
7	1/27/2017	1	1749	36.11	109	8	\$ 288.88	\$ 872.00

Figure 3 - Table Containing Sales Data for a Single Product

Now examine the same dataset pictured in **Figure 4**. Notice that merely adding the date in cell A8 and pressing the **Tab** key causes the table to expand automatically, as indicated by the encircled handle now in cell H8. Further, you would achieve the same result if you paste data into the row immediately below the table and if you move data to the row immediately below the table. Likewise, if you enter, paste, or move data into the column immediately to the right of the table, the table will expand horizontally.



	A	B	C	D	E	F	G	H
1	Date	Store	Product	UnitCost	UnitPrice	SalesQuantity	TotalCost	SalesAmount
2	1/1/2017	1	1749	36.11	109	4	\$ 144.44	\$ 436.00
3	1/10/2017	1	1749	36.11	109	2	\$ 72.22	\$ 218.00
4	1/13/2017	1	1749	36.11	109	1	\$ 36.11	\$ 109.00
5	1/14/2017	1	1749	36.11	109	6	\$ 216.66	\$ 654.00
6	1/21/2017	1	1749	36.11	109	5	\$ 180.55	\$ 545.00
7	1/27/2017	1	1749	36.11	109	8	\$ 288.88	\$ 872.00
8	1/28/2010							

Figure 4 - Expanded Table Including Additional Row for Sales Data

Some Excel users may not understand the benefits of auto-expansion immediately. Recall that as the volume of data contained in the table changes, any formula or object that uses the table as its data source also changes. To illustrate, consider the example shown in **Figure 5**. The formulas in cells G3 and H3 refer to the data in the table in cells A5 through H11.

	A	B	C	D	E	F	G	H
1								
2							Total Cost	Total Sales
3							\$ 938.86	\$ 2,834.00
4								
5	Date	Store	Product	UnitCost	UnitPrice	SalesQuantity	TotalCost	SalesAmount
6	1/1/2017	1	1749	36.11	109	4	\$ 144.44	\$ 436.00
7	1/10/2017	1	1749	36.11	109	2	\$ 72.22	\$ 218.00
8	1/13/2017	1	1749	36.11	109	1	\$ 36.11	\$ 109.00
9	1/14/2017	1	1749	36.11	109	6	\$ 216.66	\$ 654.00
10	1/21/2017	1	1749	36.11	109	5	\$ 180.55	\$ 545.00
11	1/27/2017	1	1749	36.11	109	8	\$ 288.88	\$ 872.00

Figure 5 - Table with Summary Formulas outside the Table

Now see what happens when we add data to the table. Placing our cursor in cell A12, we add the following information to the table.

Date	Store	Product	UnitCost	UnitPrice	SalesQuantity	TotalCost	SalesAmount
1/28/2017	1	1749	36.11	109	2	72.22	218.00
1/31/2017	1	1749	36.11	109	1	36.11	109.00

Immediately upon doing so, the table “appends” the additional rows, and the formulas in cells G3 and H3 automatically include the additional data without any manual interventions whatsoever, as shown in **Figure 6**.

	A	B	C	D	E	F	G	H
1								
2							Total Cost	Total Sales
3							\$ 1,047.19	\$ 3,161.00
4								
5	Date	Store	Product	UnitCost	UnitPrice	SalesQuantity	TotalCost	SalesAmount
6	1/1/2017	1	1749	36.11	109	4	\$ 144.44	\$ 436.00
7	1/10/2017	1	1749	36.11	109	2	\$ 72.22	\$ 218.00
8	1/13/2017	1	1749	36.11	109	1	\$ 36.11	\$ 109.00
9	1/14/2017	1	1749	36.11	109	6	\$ 216.66	\$ 654.00
10	1/21/2017	1	1749	36.11	109	5	\$ 180.55	\$ 545.00
11	1/27/2017	1	1749	36.11	109	8	\$ 288.88	\$ 872.00
12	1/28/2017	1	1749	36.11	109	2	\$ 72.22	\$ 218.00
13	1/31/2017	1	1749	36.11	109	1	\$ 36.11	\$ 109.00

Figure 6 - Updated Table Data Reflected in Formulas

Similarly, in **Figure 7**, notice the “before” and “after” charts of the table data contained in Figure 5 and the table data contained in Figure 6. Again, due to auto-expansion of the table, as the table expands or contracts, formulas and objects that use the table as their data source adjust to reflect the new volume of data in the table.

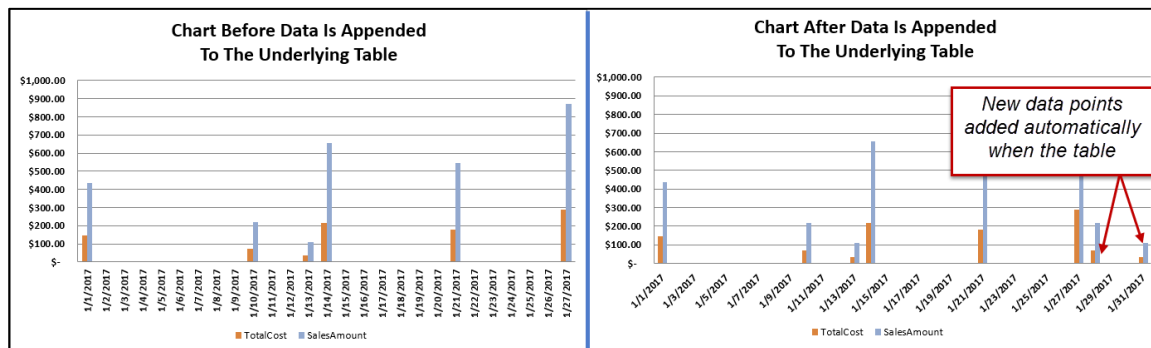


Figure 7 - "Before" and "After" Charts Showing Impact of Data Appended to Table

Structured Reference Formulas

When referencing table data, you can now use *structured references* instead of traditional R1C1 addresses. Structured references include column headers and other special symbols to ensure that calculations that refer to table data are complete and accurate.

For example, referring to the formulas in cells G3 and H3 in Figure 6, these are not traditional R1C1 formulas such as =SUM(G6:G13) or =SUM(H6:H13). Rather, these formulas make use of structured references to instruct Excel to add all of the data in TotalCost and SalesAmount columns of the table. The syntax for these two formulas is =SUM(Table13[TotalCost]) and =SUM(Table13[SalesAmount]), respectively. While the =SUM portion of these formulas is familiar, the Table13, TotalCost, and SalesAmount references might be a bit puzzling. Though we will cover structured reference formulas in detail beginning on page 35, for now, just know that Table13 is the name of the table used as a data source in the formulas, and TotalCost and

SalesAmount are column headers in the table. As constructed, the two formulas instruct Excel to sum all of the data in the *TotalCost* and *SalesAmount* columns of the table named *Table13*. Tying structured reference formulas back to auto-expansion, structured reference formulas are what allow Excel to perform calculations on entire columns of data, even as the volume of data in the columns changes.

Self-Replicating Formulas

In addition to the advantages cited previously, tables also feature *self-replicating formulas*. When you enter a formula in a cell inside a table, that formula automatically replicates up and down the column. Further, as you append data to the bottom of the table, any formulas you previously entered into the table automatically copy to the newly appended rows.

To illustrate, consider the formula being entered into cell I2 in the table pictured in **Figure 8**. Upon pressing either **Tab** or **Enter**, the formula copies all the way down the column automatically with no additional user action required.

	A	B	C	D	E	F	G	H	I	J	K
1	Date	Store	Product	UnitCost	UnitPrice	SalesQuantity	TotalCost	SalesAmount	Gross Margin		
2	1/1/2017	1	1749	36.11	109	4	\$ 144.44	\$ 436.00	=[@SalesAmount]-[@TotalCost]		
3	1/10/2017	1	1749	36.11	109	2	\$ 72.22	\$ 218.00			
4	1/13/2017	1	1749	36.11	109	1	\$ 36.11	\$ 109.00			
5	1/14/2017										
6	1/21/2017	1	1749	36.11	109	5	\$ 180.55	\$ 545.00			
7	1/27/2017	1	1749	36.11	109	8	\$ 288.88	\$ 872.00			

Figure 8 - Formulas Entered in a Table Automatically Replicate in a Column

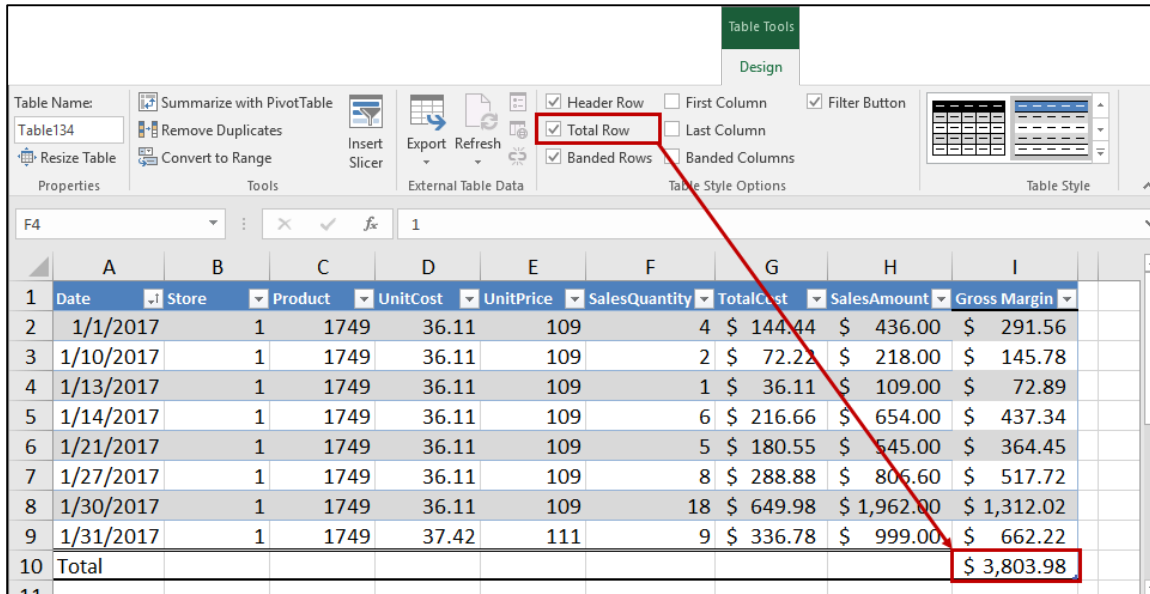
Further, as shown in **Figure 9**, as additional rows of data append to the table, the formula continues to self-replicate. This behavior occurs even if the table data is imported or linked from an external data source, as discussed beginning on page 10.

	A	B	C	D	E	F	G	H	I	J	K
1	Date	Store	Product	UnitCost	UnitPrice	SalesQuantity	TotalCost	SalesAmount	Gross Margin		
2	1/1/2017	1	1749	36.11	109	4	\$ 144.44	\$ 436.00	\$ 291.56		
3	1/10/2017	1	1749	36.11	109	2	\$ 72.22	\$ 218.00	\$ 145.78		
4	1/13/2017	1	1749	36.11	109	1	\$ 36.11	\$ 109.00	\$ 72.89		
5	1/14/2017										
6	1/21/2017	1	1749	36.11	109	5	\$ 180.55	\$ 545.00	\$ 364.45		
7	1/27/2017	1	1749	36.11	109	8	\$ 288.88	\$ 872.00	\$ 583.12		
8	1/30/2017	1	1749	36.11	109	18	\$ 649.98	\$ 1,962.00	\$ 1,312.02		
9	1/31/2017	1	1749	37.42	111	9	\$ 336.78	\$ 999.00	\$ 662.22		

Figure 9 - Table Formulas Copy Automatically to New Rows

Summary Rows

The last major advantage of working with tables is the ability of the table to *generate a summary row of information* at the bottom of the table. For instance, in the table shown in **Figure 10**, merely checking the **Total Row** box in the **Table Tools Design** contextual tab causes Excel to insert a row with a **SUBTOTAL** function in the last column of the table. If the last column of data is numeric, the SUBTOTAL function will be **SUM**. If the last column of data is text or dates, the SUBTOTAL function will be **COUNT**. No matter the function, Excel will always label the row **Total**.



The screenshot shows the 'Table Tools Design' tab in Excel. The 'Table Name' is 'Table134'. In the 'Table Style Options' group, the 'Total Row' checkbox is checked. A red arrow points from this checkbox to the 'Total' row in the table below.

	A	B	C	D	E	F	G	H	I
1	Date	Store	Product	UnitCost	UnitPrice	SalesQuantity	TotalCost	SalesAmount	Gross Margin
2	1/1/2017	1	1749	36.11	109	4	\$ 144.44	\$ 436.00	\$ 291.56
3	1/10/2017	1	1749	36.11	109	2	\$ 72.22	\$ 218.00	\$ 145.78
4	1/13/2017	1	1749	36.11	109	1	\$ 36.11	\$ 109.00	\$ 72.89
5	1/14/2017	1	1749	36.11	109	6	\$ 216.66	\$ 654.00	\$ 437.34
6	1/21/2017	1	1749	36.11	109	5	\$ 180.55	\$ 545.00	\$ 364.45
7	1/27/2017	1	1749	36.11	109	8	\$ 288.88	\$ 806.60	\$ 517.72
8	1/30/2017	1	1749	36.11	109	18	\$ 649.98	\$ 1,962.00	\$ 1,312.02
9	1/31/2017	1	1749	37.42	111	9	\$ 336.78	\$ 999.00	\$ 662.22
10	Total								\$ 3,803.98

Figure 10 - Inserting a Total Row into a Table

The default attribute for this formula is a sum function, but you can change that attribute to any of the other ten functions supported by SUBTOTAL. To do so, click in the cell containing the formula and then click the drop-down arrow that appears next to the cell. Finally, from the resulting contextual menu, choose the desired function as shown in **Figure 11**.

	A	B	C	D	E	F	G	H	I
1	Date	Store	Product	UnitCost	UnitPrice	SalesQuantity	TotalCost	SalesAmount	Gross Margin
2	1/1/2017	1	1749	36.11	109	4	\$ 144.44	\$ 436.00	\$ 291.56
3	1/10/2017	1	1749	36.11	109	2	\$ 72.22	\$ 218.00	\$ 145.78
4	1/13/2017	1	1749	36.11	109	1	\$ 36.11	\$ 109.00	\$ 72.89
5	1/14/2017	1	1749	36.11	109	6	\$ 216.66	\$ 654.00	\$ 437.34
6	1/21/2017	1	1749	36.11	109	5	\$ 180.55	\$ 545.00	\$ 364.45
7	1/27/2017	1	1749	36.11	109	8	\$ 288.88	\$ 806.60	\$ 517.72
8	1/30/2017	1	1749	36.11	109	18	\$ 649.98	\$ 1,962.00	\$ 1,312.02
9	1/31/2017	1	1749	37.42	111	9	\$ 336.78	\$ 999.00	\$ 662.22
10	Total								\$ 3,803.98
11									None
12									Average
13									Count
14									Count Numbers
15									Max
									Min
									Sum
									StdDev
									Var
									More Functions...

Figure 11 - Changing the Total Row in a Table

Additionally, though Excel seemingly creates the SUBTOTAL formula only in the right-most column of the table, you can click in any other column of the total row and activate the SUBTOTAL formula for that row by changing the attribute as shown in Figure 11.

The combination of automatic and dynamic formatting, auto-expansion, structured referencing, self-replicating formulas, and automatic summary rows should cause most users of Excel 2007 and higher to rethink how they store, manage, and query data in Excel.

Creating Tables

Knowing the benefits of working with tables, let us now turn our attention to creating tables so that we may experience these benefits. Tables are easy to create using any one of three methods:

1. Creating tables from the Ribbon,
2. Creating tables with a keyboard shortcut, and
3. Creating tables from external data sources.

Additionally, though beyond the scope of this course, you can create tables with macros.

Data Considerations before Creating a Table

Before you create a table using either of the first two techniques – creating a table from the Ribbon or using a keyboard shortcut – first consider how the data for the table is arranged and whether you should make any modifications. Generally, you should arrange the initial data for the table in contiguous rows and columns. Remember, you can always add additional rows and columns later, so there is no reason to include placeholders for future data when you create the table.

Additionally, while not necessary, you should have column headers at the top of each column that appropriately describe the data in that column; if your data does not have column headers, Excel will insert headers such as *Column1*, *Column2*, etc. into your table. All the column headers in a table must be unique. If there are duplicate names, Excel will append a number to

the name to differentiate the labels. For example, if there are three columns labeled Date, turning the data into a table will cause Excel to rename the second occurrence Date2 and the third occurrence Date3. Of course, you can change a column header at any point in the future by typing over it, and, when you do so, all structured reference formulas update to reflect the new column header.

Creating Tables from the Ribbon

With your data properly organized, you are ready to convert it into a table, and you can do just that by issuing a command from the Ribbon. First, however, click inside the range of contiguous rows and columns. Then, as shown in **Figure 12**, click **Format as Table** on the **Home** tab of the **Ribbon** and choose from one of the predefined Table Styles that appear in the resulting window. Doing so will transform your data into a table, formatted to the style that you chose. Note that as you drag your cursor over the various images of table styles, Excel's live preview will temporarily apply the formatting to your table. Clicking on a format will actually apply the formatting. Should you later decide you want to change the style, just click any cell in the table, choose the **Format as Table** icon again from the **Ribbon**, and select a different table style.

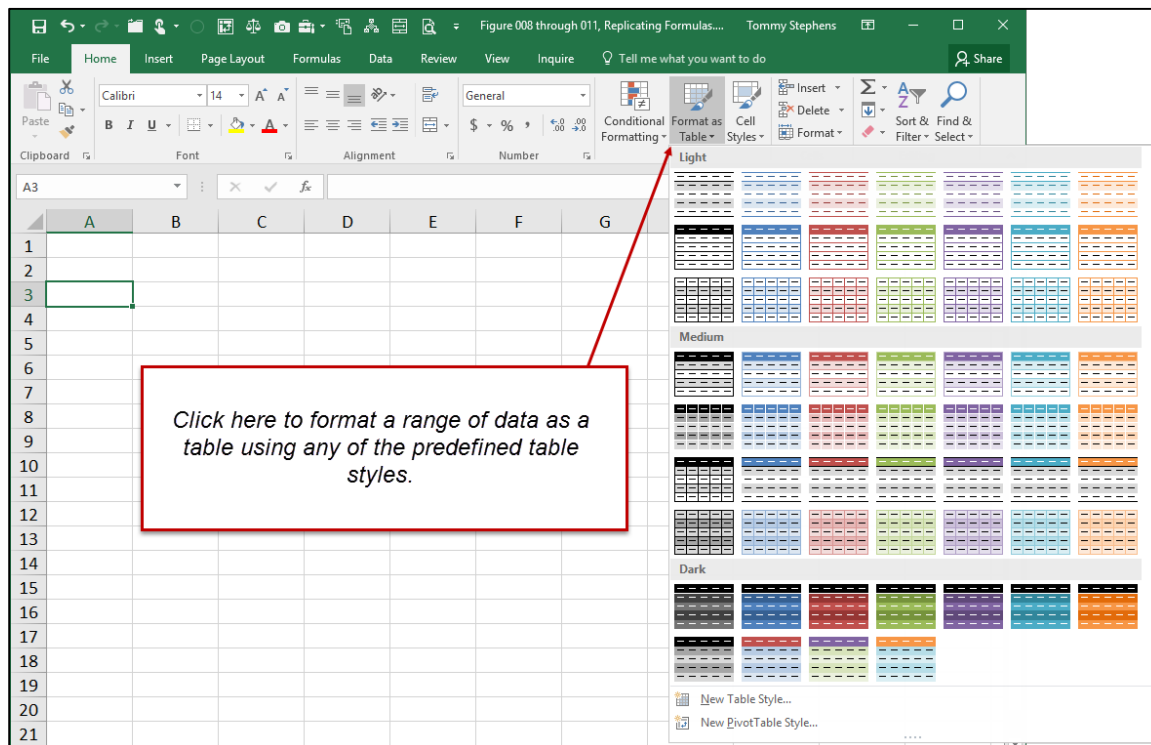


Figure 12 - Creating a Table from the Ribbon

Creating Tables with a Keyboard Shortcut

If you prefer to create your table by using a keyboard shortcut, click anywhere inside the range of data and press either **CTRL + T** or **CTRL + L**. Using either of these shortcuts will open the Create Table dialog box. Confirm the range of the data and whether your data has headers. Excel then transforms your data into a table using your default table style (described beginning on page 13). As with creating a table from the Ribbon, if you wish to reformat the table using a

different style, you can do so by choosing **Format as Table** from the **Ribbon** and choosing a new table style.

Creating Tables from External Data Sources

For those who use Excel to query data from external data sources, beginning with Excel 2007, query results must be returned to one of three choices: 1) tables, 2) PivotTables, or 3) a PivotChart with an accompanying PivotTable. The following discussion focuses on returning data queried from external data sources to tables.

To begin the process of querying data from an external data source, choose **Get External Data** from the **Data** tab of the **Ribbon** and then choose your data source as shown in **Figure 13**.

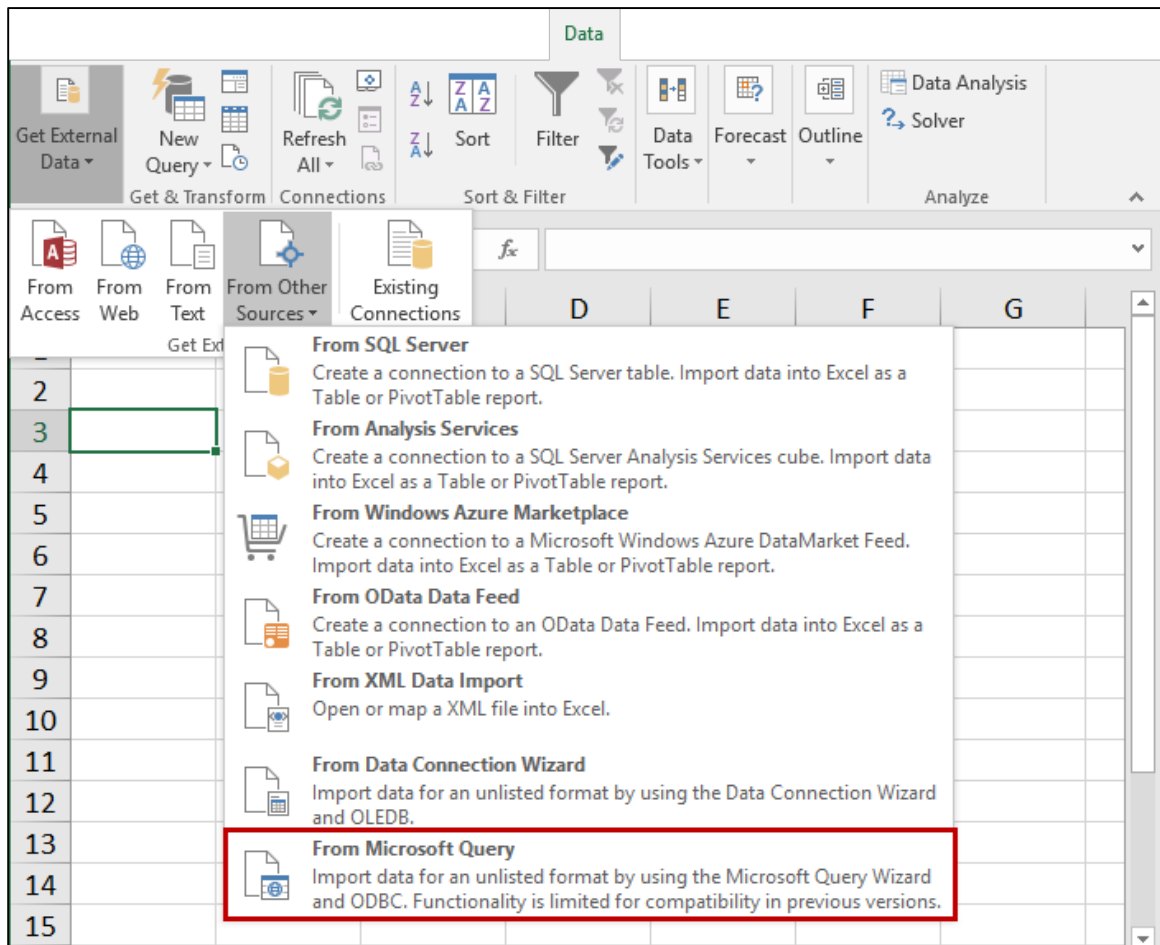


Figure 13 - Selecting a Data Source for a Data Query

Upon selecting the data source, select the table(s) and field(s) you wish to query and return to Excel as shown in **Figure 14**.

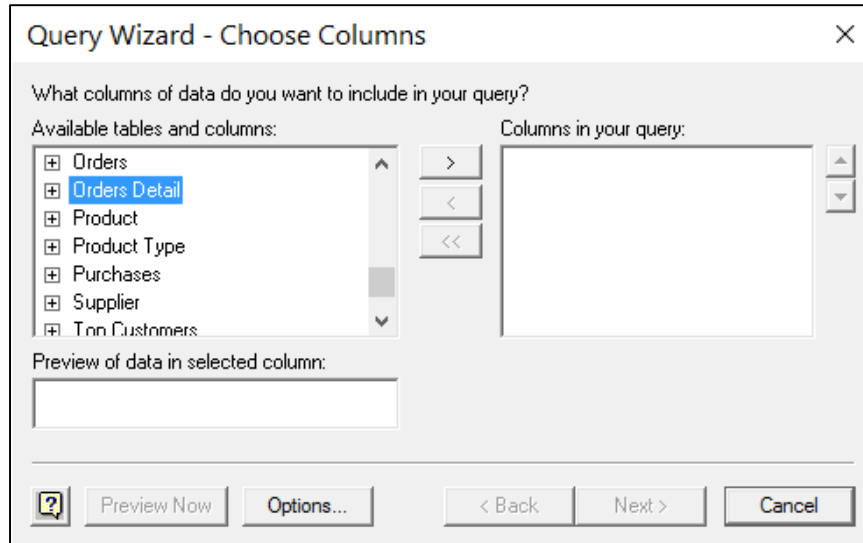


Figure 14 - Choosing Tables and Fields for Query

Click **Next**, **Next**, and **Next** again, followed by **Finish** to complete the query. Finally, in the resulting **Import Data** dialog box, choose the **Table** option and click **OK**. The query then runs, and all selected data flows into Excel as a table using the default table style.

With the data queried into an Excel table, several important considerations warrant expansion.

- From a practical perspective, the data links from the data source into Excel. Thus, as users make changes to the data in the source database, refreshing the query will refresh the results in the Excel table. To refresh the query, right-click the table and choose **Refresh**.
- If you add formulas to the table, they will self-replicate. Further, if additional records are added to the table when the query refreshes, then the formulas will self-replicate to the new rows.
- If you wish to “freeze” the results of the query in the table, you can choose **Connections** on the **Data** tab of the **Ribbon** to open the **Workbook Connections** dialog box. In that dialog box, select the query and click **Remove**. Select **OK** or **Cancel** in the warning dialog box. Select **Close** to close the **Workbook Connections** dialog box. Following this procedure de-couples the query from the workbook so that the data remains in the workbook, but you can no longer refresh the query.

Naming and Renaming Tables

Regardless of the method you choose to create a table, Excel automatically assigns a name to the table. The first table you create in a workbook inherits the name *Table1*, the second table you create in a workbook inherits the name *Table2*, and so on. Many users find these default names to be non-descriptive of the data contained in the table and, if you use numerous tables in a workbook, potentially confusing. Fortunately, you can easily rename a table and assign it a

name that makes it easier for all users of the workbook to understand the nature of the data contained in the table.

To rename a table, begin by clicking anywhere in the table. Then, from the **Table Tools Design** tab of the **Ribbon**, click in the **Table Name** field in the **Properties Group** and type the new name for your table as shown in **Figure 15**.

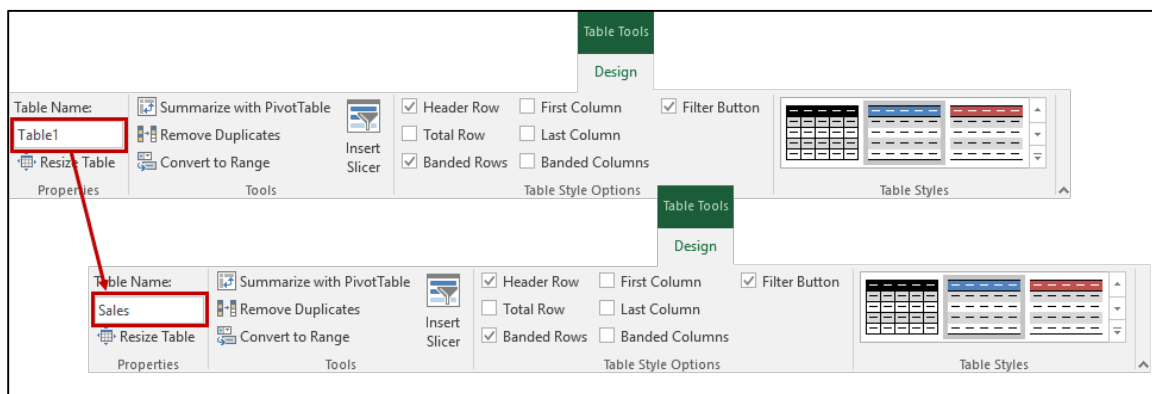


Figure 15 - Renaming a Table

While you have great freedom when creating a new table name, you must follow a few guidelines. For those familiar with the guidelines associated with defined names, the following list should look familiar.

1. The first character of the table name must be a letter, an underscore character, or a backslash. You can use numbers and special characters in a table name, but they cannot be the first character of the name. Likewise, you cannot have spaces in the table name.
2. The length of the table name must be 255 characters or fewer.
3. The table name cannot be a combination of letters and numbers that could otherwise be a row and column address. For example, *A100* cannot be a table name because it is the address of a cell in a worksheet.
4. All table names must be unique within a workbook. Table names are not case-sensitive. Excel recognizes *Revenue*, *revenues*, and *REVENUE* as the same table name.

Resizing Tables Manually

As discussed previously, after you create a table, Excel automatically resizes the table as you add and subtract data. However, if you subsequently find the need to resize the data manually, you can use either of the following techniques to do so.

First, you may do so by specifying the updated coordinates of the table in the **Resize Table** dialog box. To access the Resize Table dialog box, click anywhere in your table and then click **Resize Table** in the **Properties** group on the left side of the **Table Tools Design** contextual tab. Upon doing, Excel opens the Resize Table dialog box pictured in **Figure 16**. Note that you can

enter the new coordinates of the table directly into the dialog box, or you can use a click-and-drag approach to resize your table.

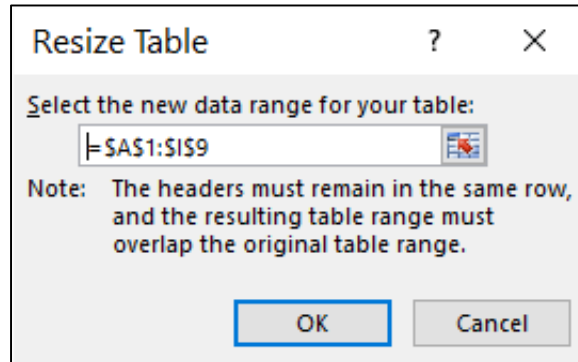


Figure 16 - Editing the Dimensions of a Table in the Resize Table Dialog Box

You can also click and drag the handle in the lower right corner of a table to resize the table. As shown in **Figure 17**, the handle is in the small icon in the lower right corner of the table.

	A	B	C	D	E	F	G	H	I
1	Date	Store	Product	UnitCost	UnitPrice	SalesQuantity	TotalCost	SalesAmount	Gross Margin
2	1/1/2017	1	1749	36.11	109	4	\$ 144.44	\$ 436.00	\$ 291.56
3	1/10/2017	1	1749	36.11					\$ 145.78
4	1/13/2017	1	1749	36.11					\$ 72.89
5	1/14/2017	1	1749	36.11					\$ 437.34
6	1/21/2017	1	1749	36.11	109	5	\$ 180.55	\$ 545.00	\$ 364.45
7	1/27/2017	1	1749	36.11	109	8	\$ 288.88	\$ 872.00	\$ 583.12
8	1/30/2017	1	1749	36.11	109	18	\$ 649.98	\$ 1,962.00	\$ 1,312.02
9	1/31/2017	1	1749	37.42	111	9	\$ 336.78	\$ 999.00	\$ 662.22
10									

Figure 17 - Click and Drag on the Handle to Resize the Table

Clicking and dragging the handle allows you to either enlarge or shrink the table, according to your current needs.

Formatting Data with Tables

You can choose to format your table data using any conventional formatting technique, such as applying boldface or italic characteristics or applying cell shading or conditional formatting. You can also format tables with table styles. In this section, we discuss how to work with table styles.

Establishing a Default Table Style

When you use the keyboard shortcut approach discussed on page 9 to create a table, the resulting table uses your default table style to format the table initially. If you wish to set a different table style as your default table style, Excel allows you to do so. Begin this process by clicking **Format as Table** on the **Home** tab of the **Ribbon** to open the list of all available table styles. Next, right-click on the table style you wish to set as your new default and, from the resulting

pop-up menu, select **Set As Default**. Upon doing so, that style will be the table style Excel applies in the future when you use the keyboard shortcut method of creating tables.

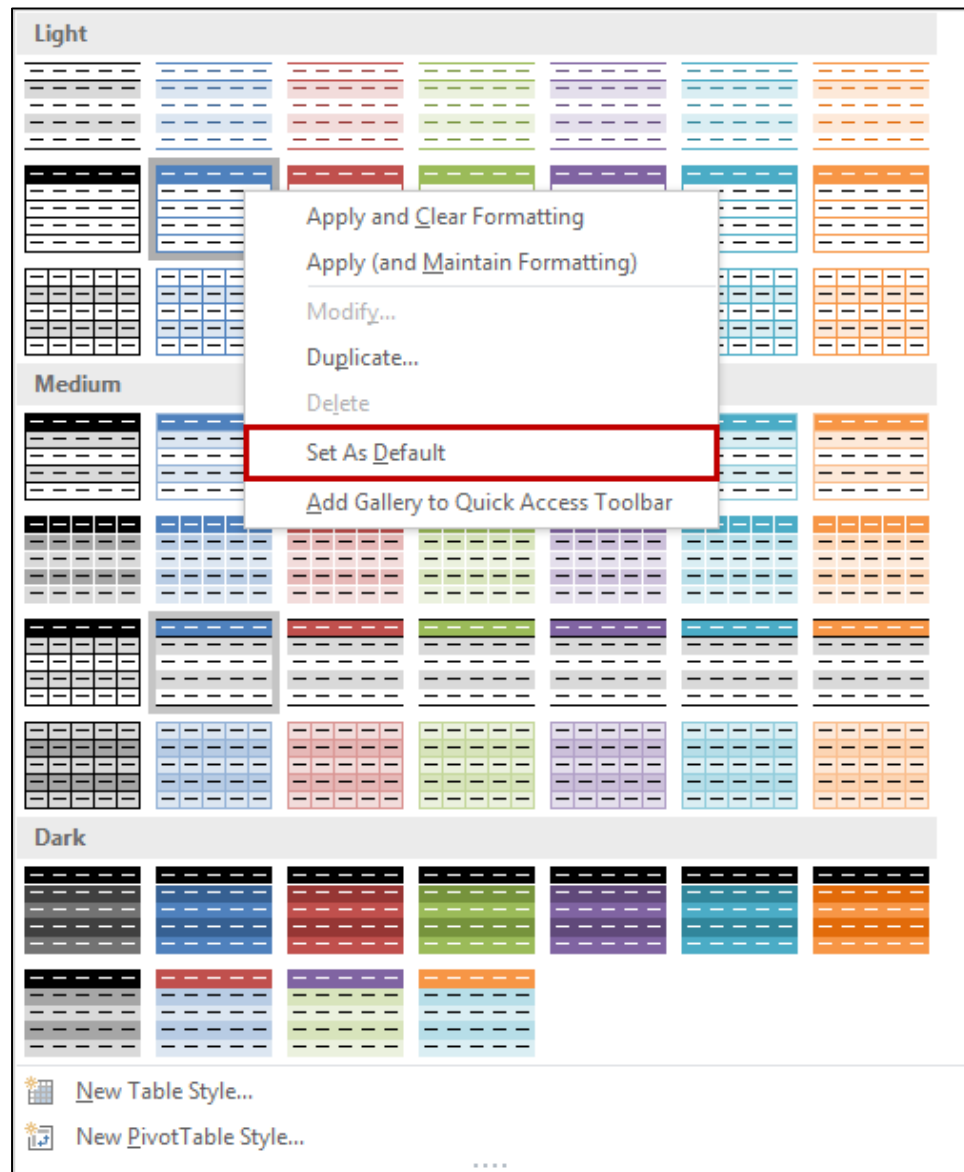


Figure 18 - Establishing a Different Default Table Style

Creating a New Table Style

If you do not find a table style that suits your needs from the list of predefined table styles, you can create one of your own. To do so, begin by clicking **Format as Table** on the **Home** tab of the **Ribbon** to open the list of all available styles shown in **Figure 18**. At the bottom of that window, click **New Table Style** to open the **New Table Style** dialog box pictured in **Figure 19**.

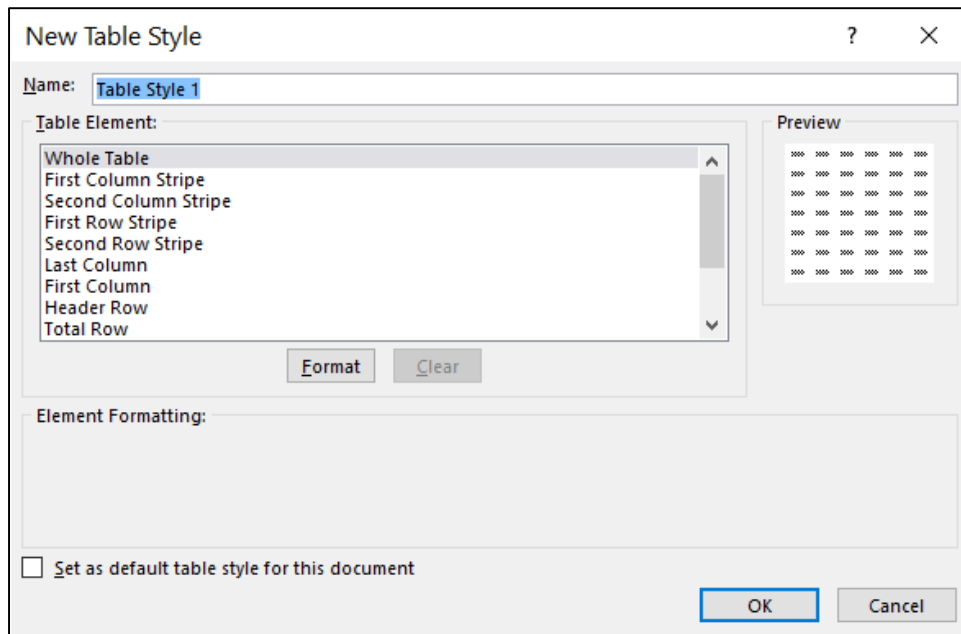


Figure 19 - New Table Style Dialog Box

In the New Table Style dialog box, enter a **Name** for the new style. Then, for each of the desired elements in the **Table Elements** list, click the element and then click **Format**. In the resulting **Format Cells** dialog box, select the formats you wish to apply to that element. Repeat this process for each element you wish to format. When you have formatted all desired elements, click **OK** to close the dialog box. Upon doing so, the style you created will appear in the **Custom** section at the top of the table styles list. Later, should you decide to delete the style, click **Format as Table**, right-click on the custom style you wish to delete, and click the **Delete** option from the resulting pop-up menu.

Converting a Table to a Range of Data

If you create a table and subsequently determine that you want to return the data back to a standard range of data in Excel, you can do so. Reasons for doing this include situations where someone using Excel 2003 or prior might need to open the workbook. In this case, if a table exists in the Excel 2007 or newer workbook, a user of Excel 2003 will not be able to work with the document.

To convert a table back to a range of data in Excel, click on the table. Then, on the **Table Tools Design** contextual tab, click **Convert to Range** as shown in **Figure 20**.

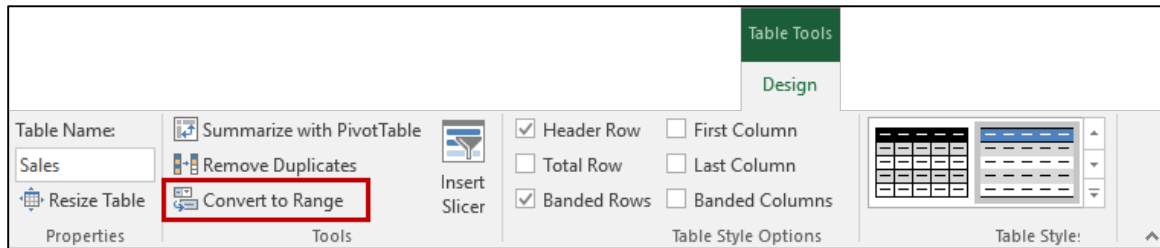


Figure 20 - Converting a Table to a Range

Upon clicking **Yes** in the resulting confirmation box, Excel converts the table to a range of data. Notably, Excel disables the filter feature on the column headers and any formulas that use structured referencing convert to standard formulas using R1C1 addresses. The range of data retains the formatting applied to the table. If the table had banded rows, the rows of the range of data are still banded. However, if you change the sequence of the rows (such as by sorting), Excel will not automatically maintain the alternating shading.

Sorting and Filtering in Tables

Now that we know how to create, rename, and resize tables, let us begin to explore some of the analytical options offered by using tables as cornerstones of our Excel workbooks. In this chapter, we explore sorting and filtering data in tables, two areas in which tables truly shine.

Learning Objectives

Upon completing this chapter, you should be able to:

- Perform basic and multiple field sorts on data in tables;
- Sort table data in non-standards ways, such as sorting horizontally and sorting based on cell formats;
- Filter table data by one or by multiple criteria;
- Utilize dynamic date filters to screen table data; and
- Create advanced table filters, such as those involving top and bottom *x* records, Slicers, and those using calculated fields as criteria.

Sorting Table Data

Like traditional ranges of data, you can easily sort table data in ascending or descending order. However, you can also sort table data using more advanced techniques, including multiple fields, case-sensitive sorts, and sorts based on cell formats.

Basic Sorting

Basic sorts of table data closely resemble basic sorts of traditional ranges of data. That is, basic sorts can be as easy as clicking in the column of data you wish to use as your sort criteria and then clicking either the **Sort A to Z** (ascending order) icon or the **Sort Z to A** (descending order) icon from the **Data** tab of the **Ribbon** as shown in **Figure 21**.

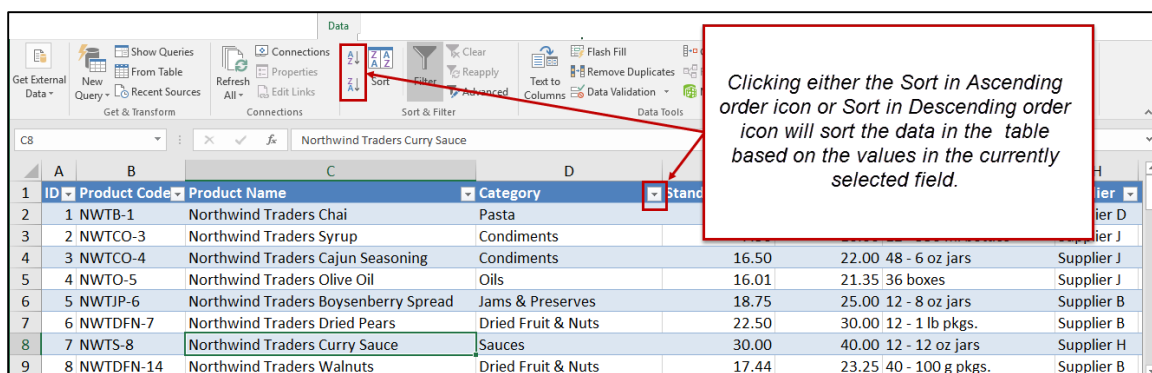


Figure 21 - Basic Sort Executed by Icon on Data Tab of the Ribbon

However, it is not necessary to sort from the Data tab of the Ribbon. Assuming the table's Filter function is active, you can sort by clicking the drop-down arrow next to a column header in a table. As shown in **Figure 22**, once you click on the drop-down arrow, you can execute simple ascending or descending sorts of your table data.

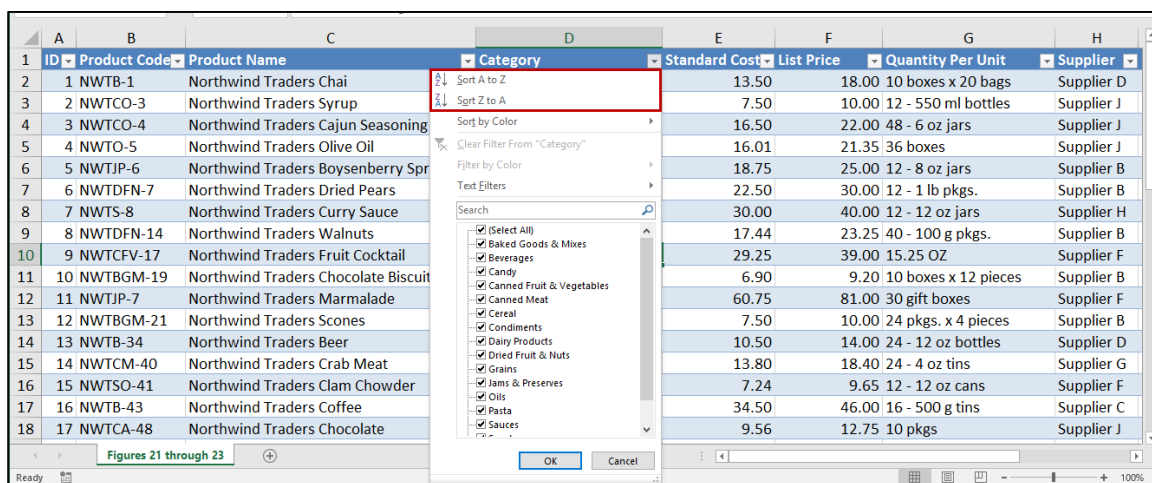


Figure 22 - Sorting a Table from the Pop-up Menu Associated with Filters

Multi-field Sorting

If you need to sort your data based on multiple criteria simultaneously, then neither of the first two options will be useful to you. Rather, you will need to click the **Sort** icon on the **Data** tab of the **Ribbon** to begin a multi-field sort. As shown in **Figure 23**, upon clicking the **Sort** icon, the **Sort** dialog box appears. In the Sort dialog box, enter desired values in the *Column*, *Sort On*, and *Order* fields and click **OK** to sort the table based on the criteria you entered in the dialog box. Note that you can enter up to sixty-four criteria for your sort.

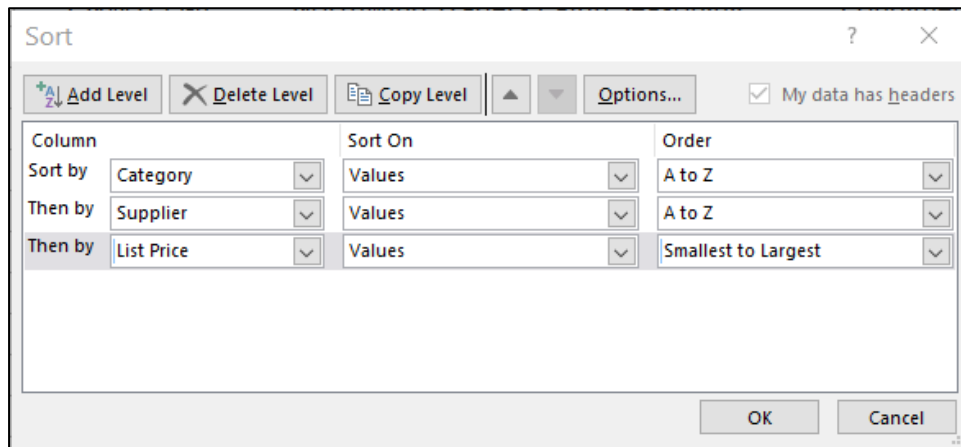


Figure 23 - Sorting Table Data Based on Multiple Criteria

Case-sensitive Sorting

By default, Excel ignores case variants when sorting. Thus, it treats *Q* the same as it treats *q* when sorting. For any given sort operation, you can change Excel's default behavior regarding case-sensitivity by clicking the **Options** button in the **Sort** dialog box and checking the box labeled **Case sensitive** in the resulting **Sort Options** dialog box pictured in **Figure 24**.

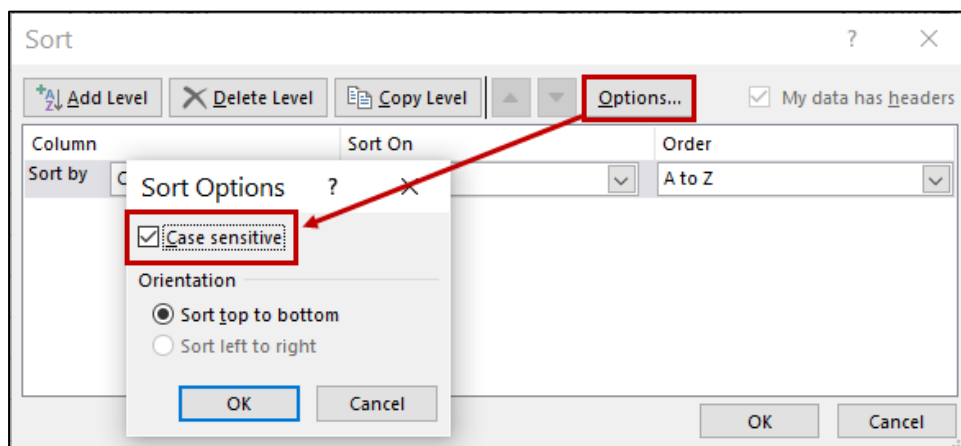


Figure 24 - Enabling Case-sensitive Sorting

Sorting Based on Cell Formats

One of the enhancements added to Excel with the 2007 release is the ability to sort data – including data in a table – based on cell formats. More specifically, you can sort based on cell color, font color, or cell icon. Further, it does not matter if you add the formats to the cell directly – such as shading a cell or changing the font color for a cell – or indirectly – such as cell colors or icons applied through conditional formatting.

To illustrate how to sort table data based on cell format criteria, consider the portion of a table shown in **Figure 25**. In this illustration, conditional formats cause the colored arrows to appear

	B	C	F
1	Product Code	Product Name	List Price
2	NWTJP-7	Northwind Traders Marmalade	81.00
3	NWTB-43	Northwind Traders Coffee	46.00
4	NWTCFV-17	Northwind Traders Fruit Cocktail	39.00
5	NWTD-72	Northwind Traders Mozzarella	34.80
6	NWTDFN-51	Northwind Traders Dried Apples	53.00
7	NWTDFN-7	Northwind Traders Dried Pears	30.00
8	NWTP-56	Northwind Traders Gnocchi	38.00
9	NWTS-8	Northwind Traders Curry Sauce	40.00
10	NWTB-1	Northwind Traders Chai	18.00
11	NWTB-34	Northwind Traders Beer	14.00
12	NWTB-81	Northwind Traders Green Tea	2.99
13	NWTB-87	Northwind Traders Tea	4.00
14	NWTBGM-19	Northwind Traders Chocolate Biscuits Mix	9.20
15	NWTBGM-21	Northwind Traders Scones	10.00

Figure 27 - Results of a Sort Based on Conditional Formats

Notice in the example shown above that because we only sorted on conditional formats, the list prices are not in any particular order within the same set of conditional formats. For instance, rows 3 through 9 in the table all possess the sideways arrow conditional format, but within that grouping, the arrangement of the data appears to be somewhat random. We can remedy this condition by adding an additional criterion to the Sort dialog box as shown in **Figure 28**.

The screenshot shows the 'Sort' dialog box with the following settings:

- Column:** List Price
- Sort On:** Values
- Order:** Largest to Smallest
- Then by:** List Price
- Sort On:** Cell Icon
- Order:** On Top
- Then by:** List Price
- Sort On:** Cell Icon
- Order:** On Top
- Then by:** List Price
- Sort On:** Cell Icon
- Order:** On Top

The 'My data has headers' checkbox is checked. The 'Options...' button is visible. The 'OK' and 'Cancel' buttons are at the bottom right.

Figure 28 - Expanded Sorting Criteria

As shown in **Figure 29**, by sorting the entire table first based on the values contained in the List Price field and then by the conditional formats in the List Price field, we find that Excel properly arranges our data.

	B	C	F
1	Product Code	Product Name	List Price
2	NWTJP-7	Northwind Traders Marmalade	81.00
3	NWTDFN-51	Northwind Traders Dried Apples	53.00
4	NWTB-43	Northwind Traders Coffee	46.00
5	NWTS-8	Northwind Traders Curry Sauce	40.00
6	NWTCFV-17	Northwind Traders Fruit Cocktail	39.00
7	NWTP-56	Northwind Traders Gnocchi	38.00
8	NWTD-72	Northwind Traders Mozzarella	34.80
9	NWTDFN-7	Northwind Traders Dried Pears	30.00
10	NWTJP-6	Northwind Traders Boysenberry Spread	25.00
11	NWTDFN-14	Northwind Traders Walnuts	23.25
12	NWTCO-4	Northwind Traders Cajun Seasoning	22.00
13	NWTO-5	Northwind Traders Olive Oil	21.35
14	NWTS-65	Northwind Traders Hot Pepper Sauce	21.05
15	NWTP-57	Northwind Traders Ravioli	19.50

Figure 29 - Data Properly Sorted Based on Multiple Criteria, Including Conditional Formats

Note that once you apply a sort or a filter to a table, if you later add additional records or modify existing ones, Excel does not automatically sort or filter those records. You, however, can reapply the original sort or filter so that it covers the preexisting records, along with the newly added or modified ones. To do so, as shown in **Figure 30**, choose **Reapply** from the **Sort & Filter** group on the **Data** tab of the **Ribbon**.

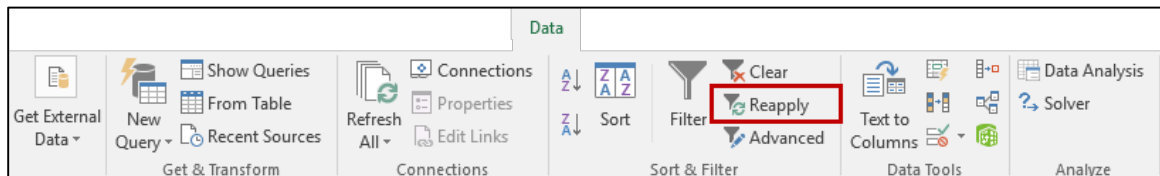


Figure 30 - Reapplying Sort and Filter Settings

Likewise, to remove all sort and filter settings quickly, click the **Clear** icon also located in the **Sort & Filter** group on the **Data** tab of the **Ribbon**.

Filtering Table Data

In addition to sorting, you can apply a variety of filters on your table data to meet varying needs. These filters range from simple, single-field filters to complex filters that utilize Boolean operators and other advanced functions to ferret out just the records in which you have interest. However, before you can begin filtering table data, you must first ensure that table filters are active. If you see drop-down arrows next to each column header, then you know filters are active. On the other hand, if you do not see drop-down arrows, then click the **Filter** icon on the **Data** tab of the **Ribbon** as shown in **Figure 31**.

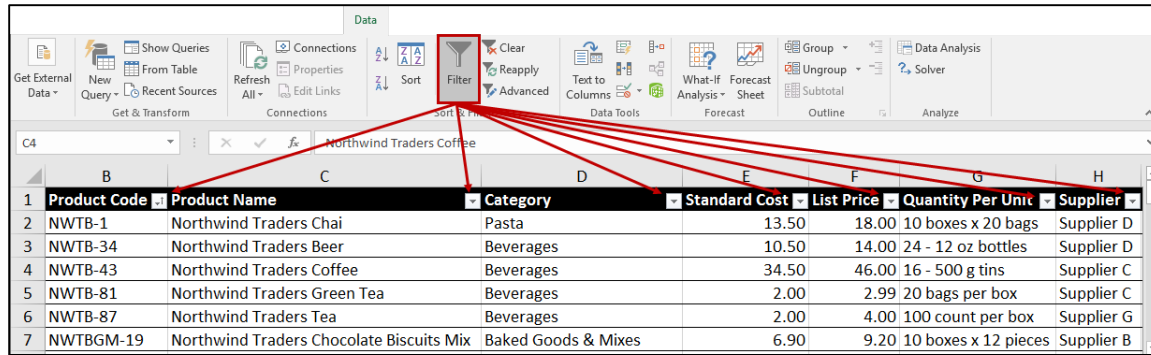


Figure 31 - Activating Table Filters

Filtering Tables on Single Fields

With table filters active, to filter a table on a single field, click the drop-down arrow in the field you wish to filter and select the filter condition(s) by checking and unchecking the boxes in the resulting pop-up menu. For example, **Figure 32** presents an illustration of filtering the Category column of the table to just those records involving *Cereal*, *Grains*, and *Pasta*.

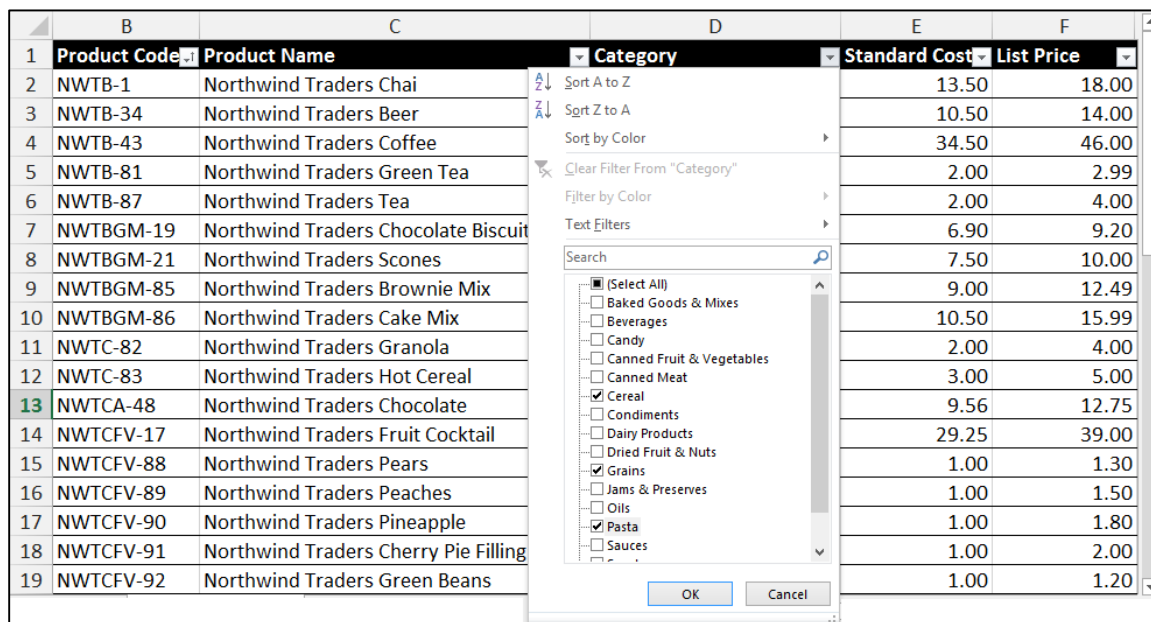


Figure 32 - Filtering Table Data in a Single Field

Text Filters

In Figure 32, note the *Text Filters* option in the pop-up menu. Because the data in the Category column is text, Excel presents the option of utilizing these special filters. For example, if you would like to filter to all items that involve *Cereal*, *Candy*, *Canned Fruit & Vegetables*, *Canned Meat*, and *Condiments*, you may find it faster to use a text filter. As shown in **Figure 33**, you could apply a text filter that filters the Category field to just those items that begin with a "C."

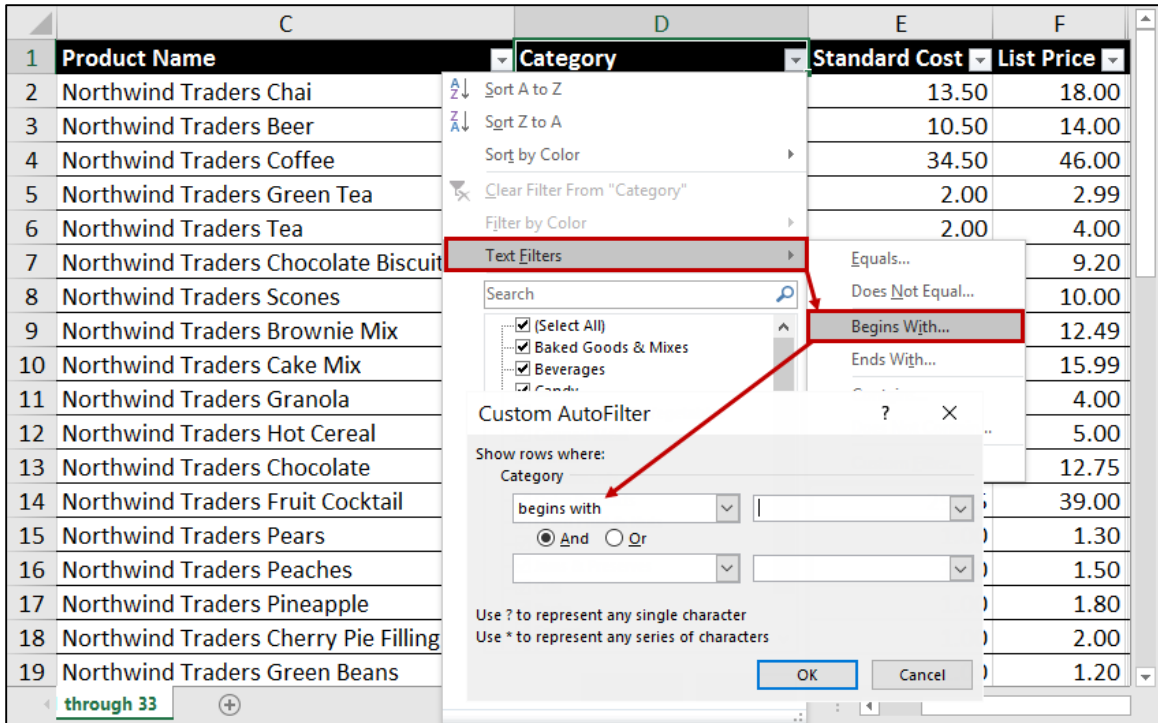


Figure 33 - Applying a Text Filter to a Table

Text filters provide a very powerful option when your tables contain a large number of disparate items. For instance, perhaps your table contains a list of all of your inventory items, and that list exceeds 100,000 different part numbers. You could use text filters with wildcard characters to filter the table to just those parts you need to analyze at a specific point in time. The wildcards “?” and “*” and their usages are shown at the bottom of the Custom AutoFilter dialog box in Figure 33.

Dynamic Date Filters

If the field that you are filtering contains date data, you can utilize Excel’s *dynamic data filters* to screen your data. These filters read the date from the system clock on your computer and filter according to selected filter settings and to the date on your computer’s clock.

To illustrate, consider the data pictured in **Figure 34**. When you apply a date filter to the Date column in the table – *Year to Date* for instance – the data in the table filters to the specified criteria, and all objects and formulas that use the table as a data source update to reflect the filtered data in the table.

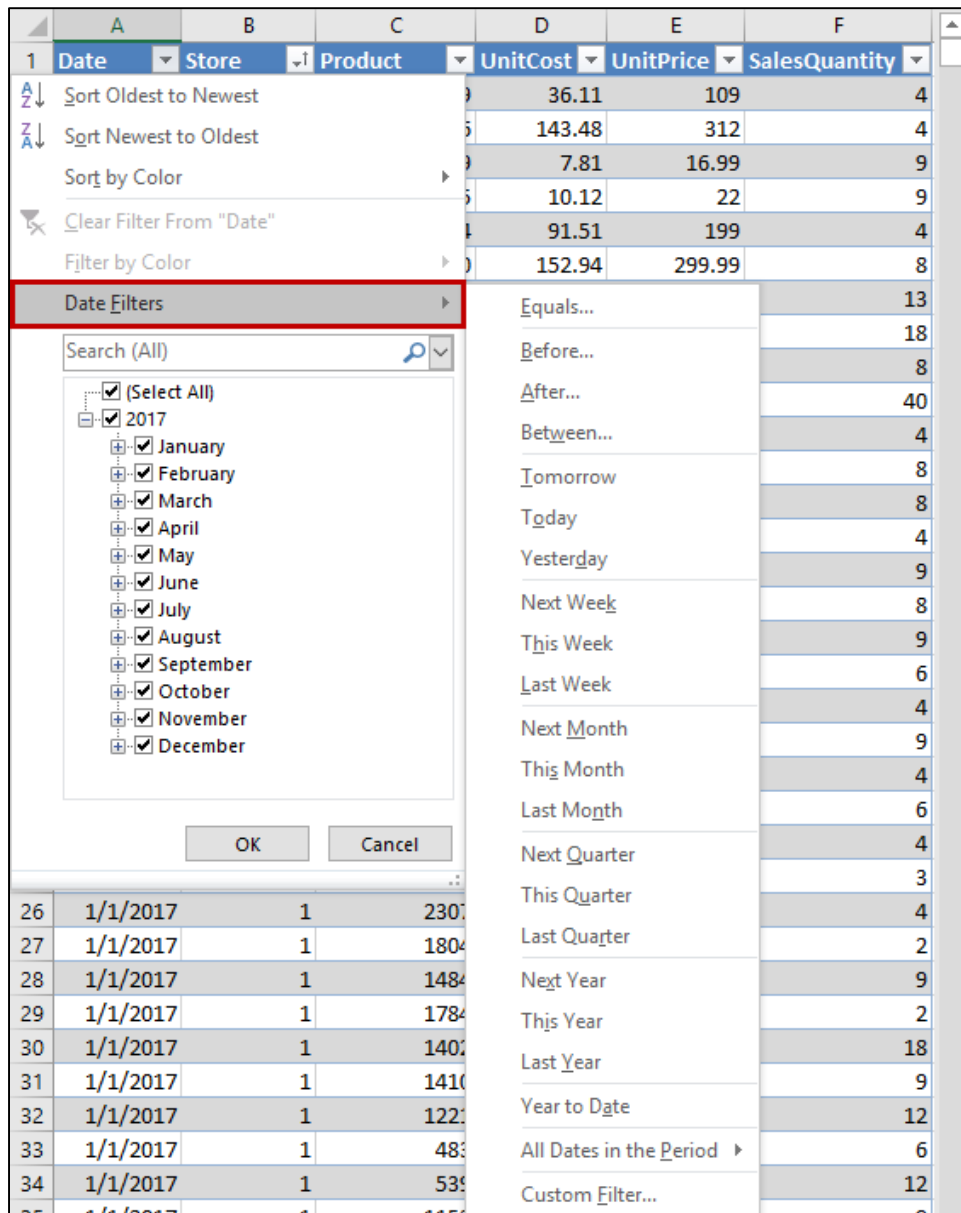


Figure 34 - Applying Date Filters to a Table

If you save and close a table with a dynamic date filter enabled in the table and subsequently open the workbook on a future date, the filter will not automatically update. To cause the filter to update based on the current date in the computer's clock, click the **Reapply** button as described on page 22.

Number Filters

In addition to text filters and dynamic date filters, you can filter table data based on number filters. Similar to date and text filters, number filters offer options that are, as their name implies, numerically oriented. For example, as shown in **Figure 35**, you can apply a number filter to screen the table for data where the Standard Cost is greater than \$10.00.

	C	D	E	F	G
1	Product Name	Category	Standard Cost	List Price	Quantity Per Unit
2	Northwind Traders Chai	Pasta	18.00	10 boxes x 20 bags	
3	Northwind Traders Syrup	Condimer	10.00	12 - 550 ml bottles	
4	Northwind Traders Cajun Seasoning	Condimer	22.00	48 - 6 oz jars	
5	Northwind Traders Olive Oil	Oils	21.35	36 boxes	
6	Northwind Traders Boysenberry Spread	Jams & Pr	25.00	12 - 8 oz jars	
7	Northwind Traders Dried Pears	Dried Fru			
8	Northwind Traders Curry Sauce	Sauces			
9	Northwind Traders Walnuts	Dried Fru			
10	Northwind Traders Fruit Cocktail	Canned F			
11	Northwind Traders Chocolate Biscuits Mix	Baked Go			
12	Northwind Traders Marmalade	Jams & Pr			
13	Northwind Traders Scones	Baked Go			
14	Northwind Traders Beer	Beverage			
15	Northwind Traders Crab Meat	Canned M			
16	No Custom AutoFilter				
17	No Show rows where:				
18	No Standard Cost				
19	No is greater than	10			
20	No And Or				
21	No				
22	No Use ? to represent any single character				
23	No Use * to represent any series of characters				
24	No				
25	Northwind Traders Mozzarella	Dairy Products	26.10	34.80	24 - 200 g pkgs.
26	Northwind Traders Almonds	Dried Fruit & Nuts	7.50	10.00	5 kg pkg.

Figure 35 - Applying Number Filters to a Table

Search Filters

Search filters, available beginning in Excel 2010, provide yet another way of filtering table data. With search filters, you type your search criteria into the Search box, and Excel executes the search based on the criteria. For example, typing the phrase “canned” in the Search box pictured in **Figure 36** automatically filters the data to those items where the Category is equal to *Canned Fruit & Vegetables* and *Canned Meat*.

	B	C	D	E
1	Product Code	Product Name	Category	Standard Cost
2	NWTB-1	Northwind Traders Chai	Sort A to Z	13.50
3	NWTCO-3	Northwind Traders Syrup	Sort Z to A	7.50
4	NWTCO-4	Northwind Traders Cajun Seasoning	Sort by Color	16.50
5	NWTO-5	Northwind Traders Olive Oil	Clear Filter From "Category"	16.01
6	NWTJP-6	Northwind Traders Boysenberry Spr	Filter by Color	18.75
7	NWTDFN-7	Northwind Traders Dried Pears	Text Filters	22.50
8	NWTS-8	Northwind Traders Curry Sauce	canned	30.00
9	NWTDFN-14	Northwind Traders Walnuts	(Select All Search Results)	17.44
10	NWTCFV-17	Northwind Traders Fruit Cocktail	Add current selection to filter	29.25
11	NWTBGM-19	Northwind Traders Chocolate Biscuit	Canned Fruit & Vegetables	6.90
12	NWTJP-7	Northwind Traders Marmalade	Canned Meat	60.75
17	NWTD-45	Northwind Traders Coffee		34.50
18	NWTCA-48	Northwind Traders Chocolate		9.56
19	NWTDFN-51	Northwind Traders Dried Apples		39.75
20	NWTG-52	Northwind Traders Long Grain Rice		5.25

Figure 36 - Using Search Filters to Filter Table Data

With those two filters in place, if you later return to the Search Box, enter **grains**, and then check the box entitled **Add current selection to filter**, your filter criteria are updated to *Canned Fruit & Vegetables*, *Canned Meat*, and *Grains*. **Figure 37** illustrates this process.

	B	C	D	E
1	Product Code	Product Name	Category	Standard Cost
10	NWTCFV-17	Northwind Traders Fruit Cocktail	Sort A to Z	29.25
15	NWTCM-40	Northwind Traders Crab Meat	Sort Z to A	13.80
35	NWTCFV-88	Northwind Traders Pears	Sort by Color	1.00
36	NWTCFV-89	Northwind Traders Peaches	Clear Filter From "Category"	1.00
37	NWTCFV-90	Northwind Traders Pineapple	Filter by Color	1.00
38	NWTCFV-91	Northwind Traders Cherry Pie Filling	Text Filters	1.00
39	NWTCFV-92	Northwind Traders Green Beans	grains	1.00
40	NWTCFV-93	Northwind Traders Corn	(Select All Search Results)	1.00
41	NWTCFV-94	Northwind Traders Peas	Add current selection to filter	1.00
42	NWTCM-95	Northwind Traders Tuna Fish	Grains	0.50
43	NWTCM-96	Northwind Traders Smoked Salmon		2.00
47				
53				
54				

Figure 37 - Adding Additional Filter Criteria as a Search Filter

Using Slicer Filters on Table Data (Excel 2013/2016 only)

Microsoft added the **Slicer** feature to Excel 2010 but only for use with PivotTables; beginning with the 2013 release of Excel, Slicers also operate with tables to filter data. Slicers are graphical filters, allowing end users to filter data quickly to the desired criteria.

To insert one or more Slicers into a workbook to filter a table, begin by selecting the table you wish to filter. Then, from the **Table Tools Design** contextual tab, choose **Insert Slicer** to open the Insert Slicers dialog box pictured in **Figure 38**.

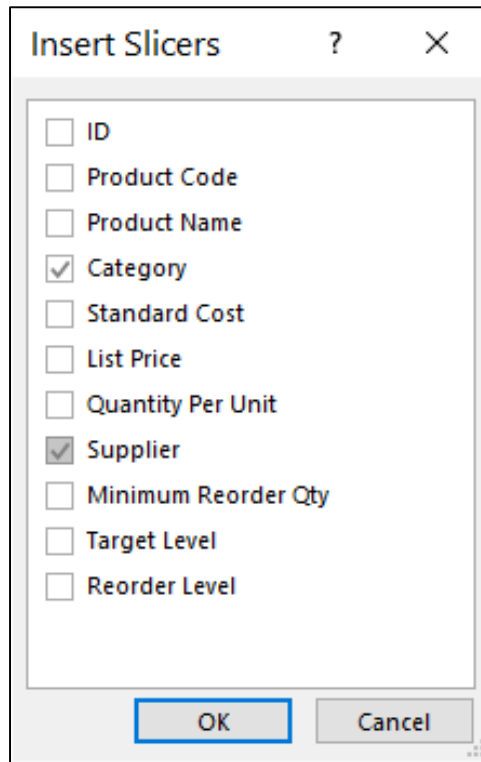


Figure 38 - Inserting Slicers into a Table in Excel 2013/2016

The Insert Slicers dialog box presents a list of each of the fields in the selected table. In the dialog box, check the boxes next to each field you wish to filter graphically. Upon clicking **OK** to complete the selection process, Excel adds the filters to the worksheet as shown in **Figure 39**. With the Slicers on the worksheet, any user can filter the table by merely selecting the desired items on the Slicers. Further, if desired, you can relocate the slicers to another worksheet. This might be advisable, for instance, in cases where you want to hide the worksheet on which the table resides but still provide interactivity with users of the spreadsheet.

You can use Slicer filters to filter table data quickly and easily. In this example, two Slicers are in use to filter the table data on two criteria. You can select multiple items in each Slicer to fine-tune your filters.

Product Code	Product Name	Category	Standard Cost	List Price	Quantity Per Unit	Supplier
NWTCFV-17	Northwind Traders Fruit Cocktail	Canned Fruit & Vegetables	29.25	39.00	15.25 OZ	Supplier F
NWTB-34	Northwind Traders Beer	Beverages	10.50	14.00	24 - 12 oz bottles	Supplier D
NWTCM-40	Northwind Traders Crab Meat	Canned Meat	13.80	18.40	24 - 4 oz tins	Supplier G
NWTB-43	Northwind Traders Coffee	Beverages	34.50	46.00	16 - 500 g tins	Supplier C
NWTB-81	Northwind Traders Green Tea	Beverages	2.00	2.99	20 bags per box	Supplier C
NWTB-87	Northwind Traders Tea	Beverages	2.00	4.00	100 count per box	Supplier G

Figure 39 - Slicer Filters Added to a Table in Excel 2013/2016

Filtering on Multiple Fields

To filter on multiple fields in a table, just apply a desired filter to each field you wish to filter. For example, **Figure 40** shows the sample data where the Category field is filtered to *Cereal*, *Grains*, and *Pasta*, the Standard Cost field is filtered to show items with a cost *greater than \$10.00*, and the Supplier field is filtered to *Supplier B*.

Product Code	Product Name	Category	Standard Cost	List Price	Supplier
NWTP-56	Northwind Traders Gnocchi	Pasta	28.50	38.00	Supplier B
NWTP-57	Northwind Traders Ravioli	Pasta	14.63	19.50	Supplier B

Figure 40 - Filtering Table Data on Multiple Fields

Advanced Filtering

In addition to the filters applied above, you can apply any number of advanced filters to table data. Among these are filters to screen for top and bottom *x* entries, filters that use formulas, and filters created using the **Advanced Filter** function.

Filtering for Top and Bottom *x* Entries

A special form of a number filter is one that screens data down to just those entries in the top or bottom echelons of the table – a top 10 list, for example. To apply this type of filter, begin by applying a number filter to a field in the table. Choose the **Top 10** option when selecting the type of number filter to apply to open the **Top 10 AutoFilter** dialog box shown in **Figure 41**. In the Top 10 AutoFilter dialog box, adjust the settings to those that meet your specific needs. Note that you can filter to any number of items from one to 500.

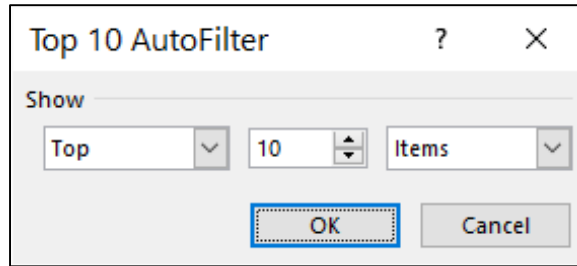


Figure 41 - Creating a Top/Bottom x Filter

Filtering Data on Calculated Values

You can also use formulas to assist in filtering table data. For example, in lieu of using dynamic date filters described beginning on page 24, you may wish to consider building a formula-based date filter instead. Doing so dispenses with the requirement to reapply the filter when the date changes.

For instance, suppose you wish to filter transactional data to just the items from yesterday. To do so, you could add a field to the table and, in that field, enter the following formula.

=IF([@Date] = TODAY()-1,"Y","N")

In this formula, *Date* is the label of the column in the table that contains the date of each transaction. The “@” preceding Date and the square brackets are required by structured referencing. You will learn more about this in Chapter 3. **Figure 42** illustrates this technique.

M86592

$\text{=IF}([\text{@Date}] = \text{TODAY}()-1, "Y", "N")$

	Store	Date	TotalCost	SalesAmount	Date Filter
86592	2	3/31/2016	93.72	190.29	Y
86593	2	3/31/2016	1,009.44	1,979.82	Y
86594	2	3/31/2016	1,572.66	3,420.00	Y
86595	2	3/31/2016	820.80	1,608.00	Y
86596	2	3/31/2016	859.56	1,686.00	Y
86597	2	4/1/2016	283.48	556.00	N
86598	2	4/1/2016	57.51	122.09	N
86599	2	4/1/2016	612.24	1,755.60	N
86600	2	4/1/2016	570.96	1,091.92	N

Figure 42 - Filtering a Table Based on a Calculated Value

When using this formula-based approach to creating date filters, whenever the workbook opens or recalculates, the results of the formula update automatically. Further, if you use the table as the data source for a PivotTable, you can filter the PivotTable to show only those items where the result of the date formula is “Y,” without the need to reapply any dynamic date filters manually.

Using the Advanced Filter Function

The *Advanced Filter* function allows you to filter table data and provides some filtering options that you otherwise do not have available. For example, the Advanced Filter function allows you to:

- Simultaneously view all filters you have applied to the table;
- Create filter criteria involving two or more fields and the conjunction OR;
- Specify three or more criteria for a single column where at least one OR conjunction is used; and
- Extract rows from a table and paste them elsewhere on the worksheet.

To apply a filter using the Advanced Filter function, first create a section in your worksheet that includes one or more labels that *exactly* match one or more column headers in the table. Then, in the rows below the labels you created, enter your criteria, keeping in mind that criteria that you place on the *same* row are treated as *AND* criteria, and criteria that you place on *separate* rows are treated as *OR* criteria. Finally, choose **Advanced Filter** from the **Data** tab of the **Ribbon** and enter into the **Advanced Filter** dialog box the name of the table as the **List range** and the cell addresses for the **Criteria range**.

To illustrate, we will use cells A1 through D3 in **Figure 43** to filter a table to all items where the *Employee Last Name* is *Harris* AND the *Department* is *Accounting* OR where the *Department* is *Sales*, regardless of any other criteria.

	A	B	C	D
1	Employee Last Name	Employee First Name	Department	Bonus
2	Harris		Accounting	
3			Sales	

Figure 43 - Filter Criteria for the Advanced Filter Function

In the **Advanced Filter** dialog box shown in **Figure 44**, *Bonus* is the name of the table that Excel will filter using the criteria in cells *A1:D3*.

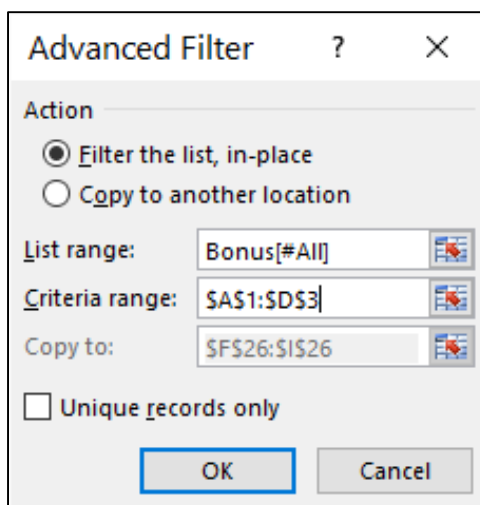


Figure 44 - Advanced Filter Dialog Box

Upon clicking **OK** in the Advanced Filter dialog box, Excel filters the table to satisfy the criteria in cells A1:D3 as shown in **Figure 45**.

	A	B	C	D
1	Employee Last Name	Employee First Name	Department	Bonus
2	Harris		Accounting	
3			Sales	
4				
5				
6				
7	Employee Last Name	Employee First Name	Department	Bonus
9	Bolinski	Heather	Sales	\$1,500.00
10	Harris	Frank	Accounting	\$2,000.00
11	Hendro	Kirby	Sales	\$2,000.00
13	Jones	Bob	Sales	\$5,000.00
14	Mason	Bradley	Sales	\$500.00
15	Miller	Lucas	Sales	\$1,000.00
21	Total			\$12,000.00

Figure 45 - Table Data Filtered by Excel's AutoFilter Function

As an option, in the Advanced Filter dialog box pictured in Figure 44, checking the box labeled **Copy to another location** and specifying a **Copy to** range would have caused Excel to leave the original table unfiltered but to create a separate output range containing the rows that match the filter criteria. Another option when using Advanced Filter is to build drop-down lists in Excel to allow users to select criteria instead of entering them manually. Finally, keep in mind that the criteria range need not be on the same worksheet as the table you are filtering. Thus, you could place criteria on a “welcome” screen in your workbook, apart from the data they affect, facilitating a dashboard environment.

Building Structured Reference Formulas

Structured reference formulas provide a different approach to building formulas that reference some or all of the data in a table. While you can still use the traditional R1C1 approach when building formulas that reference table data, structured reference formulas are the preferred approach because the dimensions or volume of table data changes frequently, and structured reference formulas adjust “on-the-fly” to changes in the table, reducing, if not eliminating, the need to edit formulas manually after table dimensions change. In this chapter, we will explore structured reference formulas and learn how to construct and utilize these tools in Excel.

Learning Objectives

Upon completing this chapter, you should be able to:

- Identify structured reference formulas and the benefits associated with using them;
- List and define the four key components of structured reference formulas;
- Create structured reference formulas that refer to portions of a table or to the entire table;
- Distinguish between qualified and unqualified structured reference formulas; and
- Create formulas that use structured referencing to handle special situations, including absolute versus relative referencing, renaming tables and column names, and rewriting a structured reference formula when converting a table to a range.

Introducing Structured Reference Formulas

As mentioned previously, structured reference formulas are different from traditional R1C1 formulas. Formulas built using structured references rely on specific references to table components instead of row and column addresses. To illustrate, compare and contrast the following two formulas.

Traditional R1C1 formula approach: **=SUM(D4:D17)**

Structured reference formula approach: **=SUM(Sales[April])**

The first formula, **=SUM(D4:D17)**, makes specific references to cells in the worksheet. If you add additional rows of data to the worksheet, you will need to revise the formula so that it includes the appended data. The second formula, **=SUM(Sales[April])**, makes reference to the *structure* of the worksheet as opposed to *cells* in the worksheet. The structured reference formula instructs Excel to sum all entries in the *April* column of the *Sales* table. This is a striking difference because, as the volume of data in the April column of the Sales table changes, the structured reference formula automatically adjusts to match the new volume of data with no end-user interaction required.

Components of a Structured Reference Formula

Before building structured reference formulas, you should understand the four basic components of structured references. These components are:

1. Table name,
2. Column specifiers,
3. Special item specifiers, and
4. Table specifier.

The following structured reference formula is used to explain each of these components.

=SUM(SalesInfo[#Totals],[Sales]],SalesInfo[#Data],[Commission]))

Table Name

In the sample formula shown above, *SalesInfo* is a *table name*. A table name is simply the name of the table containing the data the formula analyzes. A table name is not required unless the formula is built outside of the table that it references.

Column Specifiers

[Sales] and *[Commission]* are *column specifiers* in the sample formula shown above. A column specifier is the name of the column containing the data you wish to analyze with the structured reference formula. Header and subtotal rows are not included in the data referenced by a column specifier, unless the *#All* special item specifier discussed below is included. Column specifiers are not required and, when used, must be enclosed by square brackets.

Special Item Specifiers

A *special item specifier* refers to specific portions of a table, such as the header or Totals row. Special item specifiers are always prefaced by a “pound” sign (#) and, like column specifiers, are always enclosed by square brackets. In the example shown above, *[#Totals]* and *[#Data]* are special item specifiers. **Table 1** summarizes the five special item specifiers used in structured references.

Special Item Specifier	Refers To
[#All]	The entire table, including column headers and total rows
[#Data]	Only the data in the table, specifically excluding headers and total rows
[#Headers]	Just the header row
[#Totals]	Just the total row, if one exists
@ <i>Excel 2010/2013/2016</i> [#ThisRow] <i>Excel 2007</i>	Just the portion of the columns in the current row (You cannot combine [#ThisRow] with any other special item specifiers.)

Table 1 - Structured Reference Special Item Specifiers

Table Specifier

A table specifier is the outer portion of the structured reference that is enclosed in square brackets following the table name. In the example shown above, **[[#Totals],[Sales]]** and **[[#Data],[Commission]]** are both table specifiers. A table specifier can consist of a column specifier only, a special item specifier only, or a combination of the two.

Sample Structured Reference Formulas

The structured reference formulas shown in **Figure 46** are based on the table named *SalesInfo* pictured in **Figure 47**. Study how structured references are used to create each formula.

Structured Reference Formula	Result	Comment
=SUM(SalesInfo[Sales])	342,120,503	Sums the data in cells B6 through B27
=SalesInfo[[#Totals],[Sales]]+SalesInfo[[#Totals],[Commission]]	347,489,113	Adds cells B28 and C28, the Total row for Sales and Commission
=SUM(SalesInfo[#Data])	347,489,113	Adds the data in cells B6 through C27, all of the data in the table with the exception of the Total row.
=SalesInfo[[#Totals],[Commission]]	5,368,610	Returns the value in cell C28, the Total of the Commission column
=AVERAGE(SalesInfo[Commission])	244,028	Averages the data in cells C6 through C27, the Commission column
=VLOOKUP("Harris",SalesInfo,2)	17,677,019	Performs a VLOOKUP on the first column of the table and returns the related value from the second column of the table
=VLOOKUP(19407851,SalesInfo[Sales]:SalesInfo[Commission],2,FALSE)	291,118	Perform a VLOOKUP using only the Sales and Commission columns of the table and returns the related value from the Commission column
=SalesInfo[@Salesperson]	Harris	Returns the entry in the Salesperson column for the same row in which this formula is entered in the table
=SalesInfo[[#Headers],[Commission]]	Commission	Returns the Header for the Commission column

Figure 46 - Sample Structured Reference Formulas

	A	B	C
5	Salesperson ▾	Sales ▾	Commission ▾
6	Adams	14,246,779	170,961
7	Alford	15,578,025	311,561
8	Boskie	14,179,405	283,588
9	Bush	16,897,759	270,364
10	Goldberg	14,665,352	278,642
11	Greenfield	13,444,120	201,662
12	Gzehoviak	18,922,588	359,529
13	Harris	17,677,019	300,509
14	Jones	19,407,851	291,118
15	Kennedy	12,922,032	245,519
16	Lincoln	12,763,309	153,160
17	Ling	17,066,719	238,934
18	Martinez	13,906,832	250,323
19	Roberts	14,726,733	176,721
20	Rodriguez	18,509,898	333,178
21	Roebuck	18,316,258	256,428
22	Roosevelt	12,656,054	215,153
23	Smith	17,685,337	265,280
24	Thatcher	12,677,686	139,455
25	Toner	14,141,251	169,695
26	Washington	13,761,887	151,381
27	Woo	17,967,609	305,449
28	Total	342,120,503	5,368,610

Figure 47 - Sample Table for Structured Reference Formulas

Special Situations for Structured Reference Formulas

Fully Qualified and Unqualified Structured References

A fully qualified structured reference is one that references the table name in addition to any column specifiers or special item specifiers, whereas an unqualified structured reference uses only column specifiers or special item specifiers as demonstrated below.

Fully qualified structured reference: **=SUM(SalesInfo,[Sales])**

Unqualified structured reference: **=([Sales])+([Commission])**

When working with structured reference formulas, you may use unqualified structured references while working inside the table that contains the data referenced in the formula.

However, if you are building the structured reference formula outside the table, you must use fully qualified structured references.

Formula Rewrite When Converting Tables to Ranges

As discussed beginning on page 16, if you create a table, you can subsequently convert the table back to a traditional range of data. If you do this, what happens to any structured reference formulas you built while the data existed in table form? In other words, what would happen to the structured reference formula `=SUM(SalesInfo[Sales])` if the SalesInfo table is converted back to a normal range of data?

If you convert a table back to a range, all structured reference formulas adjust automatically to reflect traditional row and column addresses. Thus, the formula `=SUM(SalesInfo[Sales])` shown on the second line of Figure 46 would convert to `=SUM(Sheet1!$B$6:$B$27)`.

Likewise, if you choose to rename a table, or if you choose to rename one or more columns in a table, Excel automatically adjusts all structured reference formulas that refer to the renamed table or renamed columns to reflect the new names. Again, no user interaction is necessary for these formulas to adjust.

Absolute Versus Relative Referencing

In general, structured reference formulas are *relative*, as opposed to *absolute*, in nature. Thus, if you construct a structured reference formula that refers to a column in a table, and you subsequently insert another column into the table so that the relative position of the referenced column changes, the structured reference “follows” the referenced column to its new position. Likewise, if you build a structured reference formula and then drag it one cell to the right, the resulting formula automatically refers to the column in the table that is one position to the right of the column that was the “source” of the new formula. However, if you copy and paste a structured reference formula to another location, it continues to refer to the original data source.

Table Column Headers and Structured Reference Formulas

Table Column Headers as Text Strings

Tables treat all column headers, including dates and numbers, as text strings. Though this normally does not create too many issues for those building formulas in tables, if you intend the column header to behave like a number or like a date, you may need to make appropriate adjustments in your formulas, including structured reference formulas.

Consider, for instance, the table and formula pictured in **Figure 48**. The formula compares the year of the acquisition date to the column header to return a value of *TRUE* or *FALSE*. However, as constructed, the formula fails to calculate correctly because it is comparing a *number* from the **Acquisition Date** column to *text* from the column header. Of course, Excel forces this result because *Excel always treats table headers as text*.

Formula in C3: `=YEAR([@[Acquisition Date]])=Table1[[#Headers],[2016]]`

	A	B	C	D	E
1					
2	Asset	Acquisition Date	2016		
3	Computer	6/30/2016	FALSE		
4	Desk	2/16/2016	FALSE		
5	Chair	12/1/2015	FALSE		
6	Printer	8/7/2014	FALSE		
7	Building	4/16/2013	FALSE		

Figure 48 - Formula Referring to Text Column Header

A slight alteration of the formula is all that is necessary to cause it to calculate correctly. As shown in **Figure 49**, merely adding a *VALUE* function to the section of the formula that refers to the column header causes the formula to calculate correctly.

Formula in C4: `=YEAR([@[Acquisition Date]])=VALUE(Table13[[#Headers],[2016]])`

	A	B	C	D	E
1					
2	Asset	Acquisition Date	2016		
3	Computer	6/30/2016	TRUE		
4	Desk	2/16/2016	TRUE		
5	Chair	12/1/2015	FALSE		
6	Printer	8/7/2014	FALSE		
7	Building	4/16/2013	FALSE		

Figure 49 - Formula Correctly Calculating Based on Text Column Header

Special Characters in Column Headers

Another anomaly associated with tables and structured reference formulas is the issue associated with special characters in column headers. If a column header contains any of the following characters –

space, tab, line feed, carriage return, comma (,), colon (:), period (.), left bracket ([), right bracket (]), pound sign (#), single quotation mark ('), double quotation mark ("), left brace ({), right brace (}), dollar sign (\$), caret (^), ampersand (&), asterisk (*), plus sign (+), equal sign (=), minus sign (-), greater than symbol (>), less than symbol (<), or division sign (/)

– you must enclose the entire column header in brackets when using the column header in a structured reference formula. However, if a space is the only special character used in a column header, you need not enclose the header in brackets.

Using Tables as Data Sources

So far in this course, we have focused on the advantages of working with tables and how to create them, sorting and filtering table data, and building structured reference formulas based on table data. Now, let us turn our attention to the topic of building other objects – namely charts, PivotTables, and PivotCharts – that use tables as their data sources. In addition to structured reference formulas which update automatically based on table auto-expansion, using tables as data sources for charts, PivotTables, and PivotCharts provides some of the biggest potential productivity gains associated with using tables.

Learning Objectives

Upon completing this chapter, you should be able to:

- Construct Excel charts that utilize tables as their data source, including charts that receive filtered data from tables and creating multiple charts from the same table;
- Create PivotTables that use tables as their data source, including PivotTables that use multiple tables as their data source; and
- Build PivotCharts based on table data.

Building Charts from Tables

If the data that you wish to incorporate into a chart resides in a table, building the chart from the table is a straightforward process. Perhaps the simplest way of accomplishing this task is to begin by clicking in the table you wish to serve as the data source for the chart. Next, from the **Insert** tab of the **Ribbon**, click the type of chart you wish to create, and Excel builds the chart for you automatically. Finally, edit the chart to suit specific tastes and needs. To illustrate this process, consider the table data pictured in **Figure 50**.

	A	B	C
1	Salesperson	Sales	Commission
2	Adams	14,246,779	170,961
3	Boskie	14,179,405	283,588
4	Goldberg	14,665,352	278,642
5	Gzehoviak	18,922,588	359,529
6	Jones	19,407,851	291,118
7	Kennedy	12,922,032	245,519
8	Ling	17,066,719	238,934
9	Martinez	13,906,832	250,323
10	Roberts	14,726,733	176,721
11	Rodriguez	18,509,898	333,178
12	Toner	14,141,251	169,695
13	Woo	17,967,609	305,449
14	Total	190,663,049	3,103,657

Figure 50 - Table Data for Chart

Upon clicking in the table data and choosing a **Clustered Column** chart from the **Insert** tab of the **Ribbon**, the resulting unedited chart appears as shown in **Figure 51**.

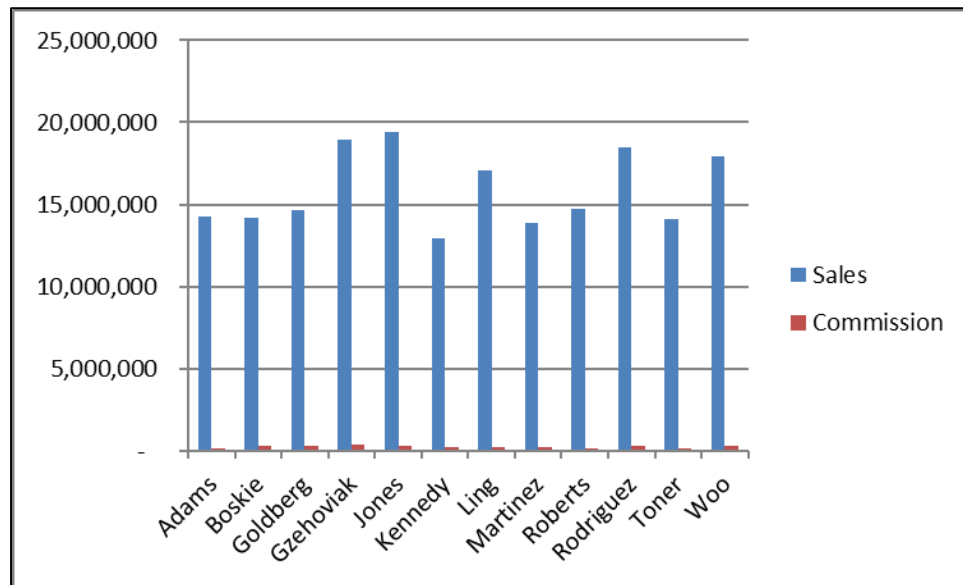


Figure 51 - Unedited Chart Built from Table Data

While the unedited chart is technically correct, it leaves much to be desired from a practical standpoint. For example, because the data in the *Commission* field is much smaller than the data in the *Sales* field, having both numbers plotted on the same vertical axis renders the *Commission* data on the chart unreadable. To address this issue, perform the following four steps.

1. Select the chart by clicking it.
2. Click the **Chart Tools Format** contextual tab.
3. In the **Current Selection** group, choose **Series “Commission”** from the drop-down menu.
4. Also in the **Current Selection** group, choose **Format Selection** and in the resulting **Format Data Series** dialog box, choose **Plot Series On Secondary Axis**.

Upon doing so and after making other formatting changes, the updated chart appears as shown in **Figure 52**.

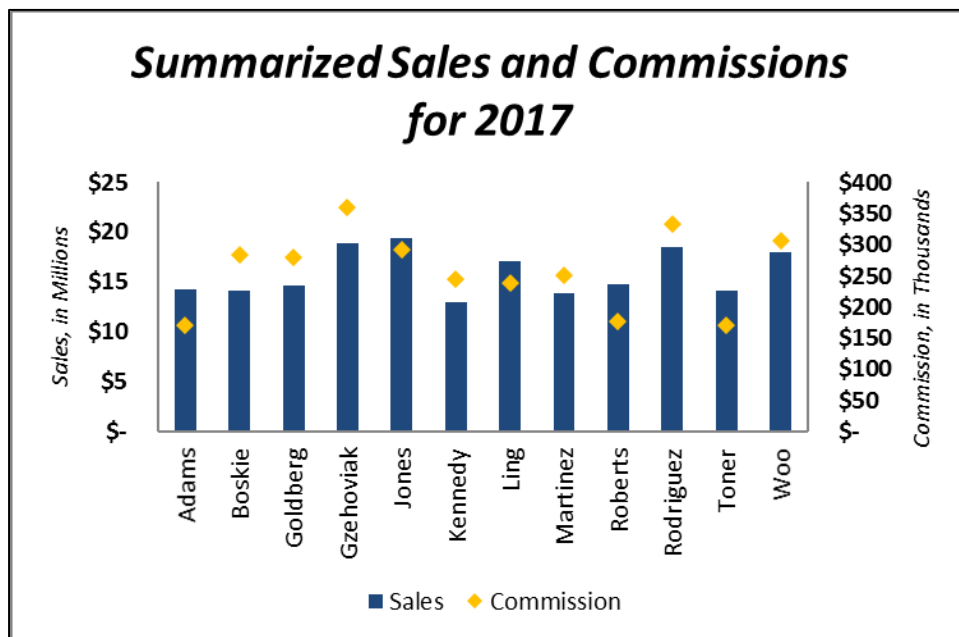


Figure 52 - Formatted Chart Based on Table Data

Importantly, with the chart built based on table data, as you add more data to the table, the chart automatically expands to include the additional data. Likewise, if you delete data from the table, the chart automatically contracts.

Filtering Charts by Filtering Tables

Additionally, once you have built a chart based on table data, if you subsequently filter the table, the chart also automatically updates to reflect the filtered data. To illustrate, expanding on the example presented above, suppose the table data source for the chart includes a *Region* field as pictured in **Figure 53**.

Salesperson	Region	Sales	Commission
Adams	East	14,246,779	170,961
Boskie	West	14,179,405	283,588
Goldberg	North	14,665,352	278,642
Gzehoviak	South	18,922,588	359,529
Jones	East	19,407,851	291,118
Kennedy	West	12,922,032	245,519
Ling	North	17,066,719	238,934
Martinez	South	13,906,832	250,323
Roberts	East	14,726,733	176,721
Rodriguez	West	18,509,898	333,178
Toner	North	14,141,251	169,695
Woo	South	17,967,609	305,449
Total		190,663,049	3,103,657

Figure 53 - Modified Table Data Including Region Field

Upon constructing a chart based on the table data in Figure 53 and filtering the table to display only the data from the North region, the chart automatically updates as shown in **Figure 54**.

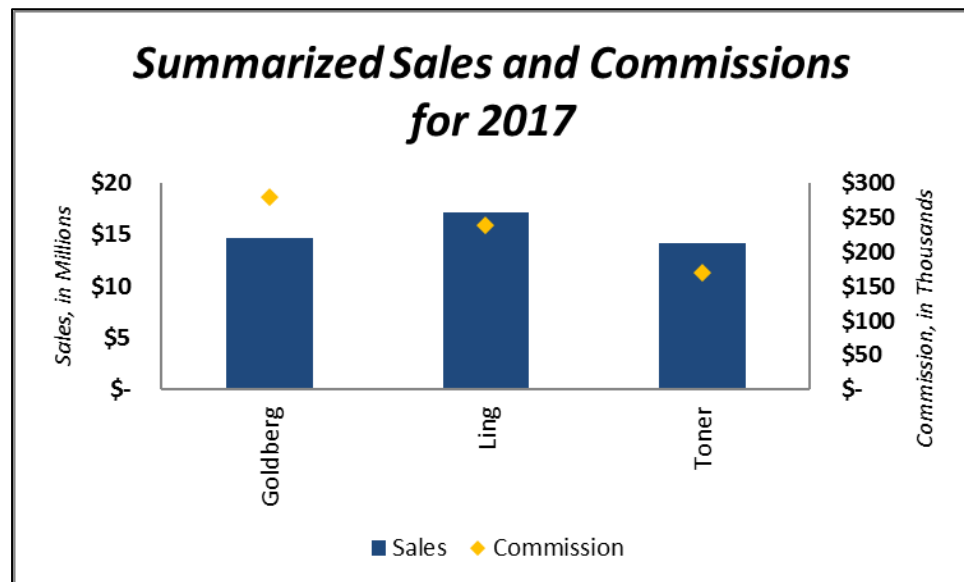


Figure 54 - Chart with Filtered Table Data

Multiple Charts from the Same Table

You are not limited to building a single chart from a table; in fact, you can construct a virtually unlimited number of charts using the same table as a data source. Be aware, however, that if you choose to do this, any change you make to the table data or any filters you apply to the table will flow through to *all* charts that use the table as their data source.

Returning to the data pictured in Figure 50, suppose that you also wanted to build a pie chart that uses the table data as a data source in addition to the chart shown in Figure 52. To do so, return to the table, click in it, and then follow the steps outlined previously for creating a chart.

If you want to build two charts using the same table data but want to sort or filter the data differently for each chart, you must base each chart on its own copy of the table data. Begin by making a copy of the table before you build the second chart. Then, use the copied table as the data source for the second chart.

Using Tables as Sources for PivotTables

Just as tables lend themselves readily as sources of data for charts, they also work exceedingly well as the source of data for PivotTables. In fact, once you have created a table, Excel incorporates a shortcut on the **Table Tools Design** contextual tab to building a PivotTable from the table as shown in **Figure 55**.

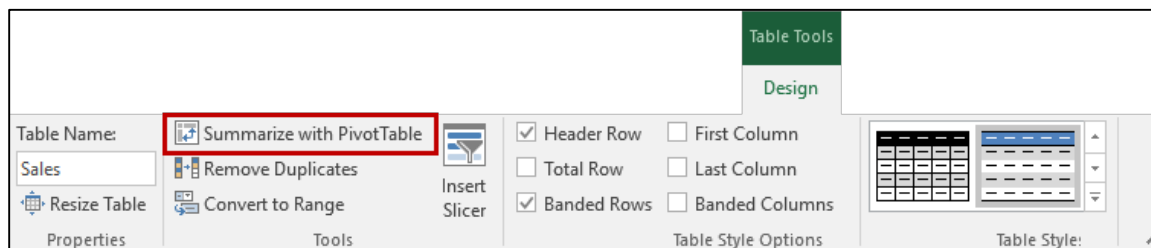


Figure 55 - Summarize with PivotTable Shortcut on Table Tools Design Contextual Tab

Upon clicking the **Summarize with PivotTable** option, Excel opens the **Create PivotTable** dialog box pictured in **Figure 56**. Here, you can confirm the table as the data source and choose between placing the PivotTable on a new worksheet or on the existing worksheet. Upon clicking **OK** in the Create PivotTable dialog box, you can complete the process of building your PivotTable using table data as the source.

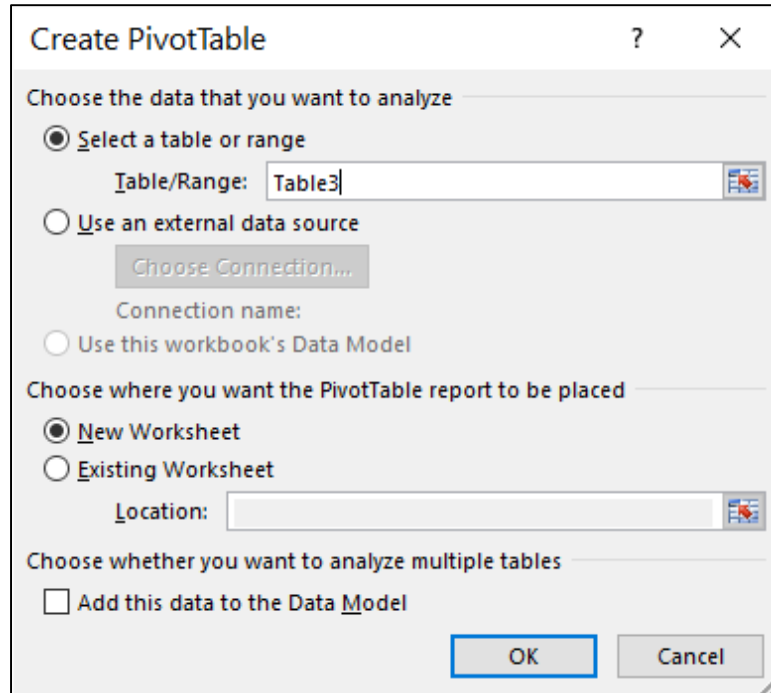


Figure 56 - Create PivotTable Dialog Box

Of course, the primary advantage of using a table as the data source for a PivotTable is the auto-expansion feature of the table. As you add data to the table and **Refresh** the PivotTable, the PivotTable automatically includes the newly appended data without any manual interaction required by the end user.

In addition to the method described above for incorporating table data into a PivotTable, you can launch the process a different way. From the **Insert** tab of the **Ribbon**, click **PivotTable** to open the **Create PivotTable** dialog box. Next, enter the name of the table you wish to use as your data source for the PivotTable in the **Table/Range:** box, specify the location of the PivotTable, and click **OK** to complete the process. Note that this process is identical to the one previously described, except that it originates from the Insert tab of the Ribbon instead of the Table Tools Design contextual tab.

Building a PivotTable from Multiple Tables

Suppose the data you wish to incorporate into a PivotTable resides in multiple tables. If you are thoroughly familiar with PivotTables, you already know that you cannot build a PivotTable from multiple data sources, even if the data sources are tables. Therefore, strictly speaking, you cannot build a PivotTable from multiple data sources. However, three workaround solutions do exist if you need to build a PivotTable from multiple tables.

1. You could consolidate the data from multiple tables into one table, likely by using Excel's VLOOKUP function, or
2. If you are using Excel 2010 or newer, you could use the Power Pivot feature available for those products.

- If you are using Excel 2013 or Excel 2016, you can create Data Models, which are multiple tables linked together based on the presence of one or more common fields. Once you create a data model, you can use it as the data source for a PivotTable.

Consolidating Multiple Tables into One

Suppose you have two (or more) tables of data that you would like to incorporate into a PivotTable, similar to the two tables pictured in **Figure 57**.

Product Code	Sales	COGS	Gross Margin
NWTB-1	257,924	165,071	92,853
NWTCO-3	337,695	192,486	145,209
NWTCO-4	319,745	163,070	156,675
NWTO-5	264,418	166,583	97,835
NWTJP-6	205,567	108,951	96,616
NWTFN-7	178,246	99,818	78,428
NWTS-8	304,829	152,465	149,366
NWTFN-14	288,783	167,115	127,065
NWTCFV-17	285,711	179,813	105,604
NWBTGM-19	177,839	96,033	81,806
NWTJP-2	186,893	100,922	85,971
NWBTGM-11	259,338	155,603	103,735
NWTCO-34	54,711	30,712	11,540

Product Code	Product Name	Category
NWTFN-74	Northwind Traders Almonds	Dried Fruit & Nuts
NWTFN-34	Northwind Traders Beer	Beverages
NWTJP-6	Northwind Traders Boysenberry Spread	Jams & Preserves
NWBTGM-85	Northwind Traders Brownie Mix	Baked Goods & Mixes
NWTCO-4	Northwind Traders Cajun Seasoning	Condiments
NWBTGM-86	Northwind Traders Cake Mix	Baked Goods & Mixes
NWTFN-1	Northwind Traders Chai	Tea
NWTCFV-91	Northwind Traders Cherry Pie Filling	Canned Fruit & Vegetables
NWTSO-99	Northwind Traders Chicken Soup	Soups
NWTCO-48	Northwind Traders Chocolate	Candy
NWBTGM-19	Northwind Traders Chocolate Biscuits Mix	Baked Goods & Mixes
NWTSO-41	Northwind Traders Clam Chowder	Soups
NWTFN-43	Northwind Traders Coffee	Beverages

Figure 57 - Multiple Tables to Combine as a Single Table

In this example, one of the tables contains summarized transactional data – *Sales*, *COGS*, and *Gross Margin* – and one contains masterfile data – *Product Name* and *Category*. Because there is a common field between the two tables, we can relate the two tables together and append the data from one table into the second table. To do so, we will use two VLOOKUP formulas to add the *Product Name* and *Category* fields found in the Masterfile table to the Transactional table. The three steps for completing this process are listed below.

- Add two fields to the Transactional table – *Product Name* and *Category*.
- In the newly added *Product Name* field, enter the following VLOOKUP formula.

=VLOOKUP([@[Product Code]],Masterfile[#All],2,FALSE)

- In the newly added *Category* field, enter the following VLOOKUP formula.

=VLOOKUP([@[Product Code]],Masterfile[#All],3,FALSE)

At this point, the Transactional table should now resemble that shown in **Figure 58**.

	A	B	C	D	E	F
1	Product Code	Sales	COGS	Gross Margin	Product Name	Category
2	NWTB-1	257,924	165,071	92,853	Northwind Traders Chai	Pasta
3	NWTCO-3	337,695	192,486	145,209	Northwind Traders Syrup	Condiments
4	NWTCO-4	319,745	163,070	156,675	Northwind Traders Cajun Seasoning	Condiments
5	NWTO-5	264,418	166,583	97,835	Northwind Traders Olive Oil	Oils
6	NWTJP-6	205,567	108,951	96,616	Northwind Traders Boysenberry Spread	Jams & Preserves
7	NWTFN-7	178,246	99,818	78,428	Northwind Traders Dried Pears	Dried Fruit & Nuts
8	NWTS-8	304,829	155,463	149,366	Northwind Traders Curry Sauce	Sauces
9	NWTFN-14	288,783	161,718	127,065	Northwind Traders Walnuts	Dried Fruit & Nuts
10	NWTCFV-17	285,417	179,813	105,604	Northwind Traders Fruit Cocktail	Canned Fruit & Vegetables

Figure 58 - Transactional Data Appended with Masterfile Data

Assuming the VLOOKUP formulas are entered correctly, and the Transactional table updates correctly, you can now build your PivotTable directly from the Transactional table. Upon completion, it might resemble that shown in **Figure 59**.

3	Product Category/Name	Monthly Sales	Cost of Goods Sold	Gross Profit	Percent of Contribution
4	Baked Goods & Mixes				
5	Northwind Traders Brownie Mix	226,611	120,104	106,507	26.40%
6	Northwind Traders Cake Mix	309,530	198,099	111,431	27.62%
7	Northwind Traders Chocolate Biscuits Mix	177,839	96,033	81,806	20.28%
8	Northwind Traders Scones	259,338	155,603	103,735	25.71%
9	Baked Goods & Mixes Total	973,318	569,839	403,479	100.00%
10	Beverages				
11	Northwind Traders Beer	354,758	202,212	152,546	31.67%
12	Northwind Traders Coffee	204,672	110,523	94,149	19.55%
13	Northwind Traders Green Tea	220,386	116,805	103,581	21.51%
14	Northwind Traders Tea	298,529	167,176	131,353	27.27%
15	Beverages Total	1,078,345	596,716	481,629	100.00%
16	Candy	336,158	191,610	144,548	100.00%
17	Canned Fruit & Vegetables	2,180,718	1,287,740	892,978	100.00%
18	Canned Meat	886,907	498,443	388,464	100.00%
19	Cereal	449,717	267,093	182,624	100.00%
20	Condiments	969,494	530,306	439,188	100.00%
21	Dairy Products	279,122	156,308	122,814	100.00%
22	Dried Fruit & Nuts	1,298,231	744,117	554,114	100.00%
23	Grains	344,326	223,812	120,514	100.00%
24	Jams & Preserves	392,460	209,873	182,587	100.00%
25	Oils	264,418	166,583	97,835	100.00%
26	Pasta	881,950	559,359	322,591	100.00%
27	Sauces	681,341	377,788	303,553	100.00%
28	Snacks	212,194	108,219	103,975	100.00%
29	Soups	702,978	422,512	280,466	100.00%
30	Grand Total	11,931,677	6,910,318	5,021,359	

Figure 59 - PivotTable Prepared from Combined Table Data

An alternate approach to solving this problem would be to copy the Transactional table and append the Masterfile data to the copied table through VLOOKUP formulas as shown above. An advantage to this approach is that the original Transactional table remains undisturbed. However, the copied Transactional table would not have any links back to the original Transactional table; therefore, if data changes in the original table, the copied table will not reflect those changes.

Using the Data Models and the Power Pivot Add-in to Consolidate Table Data for PivotTables

If you are using Excel 2010 or newer, you should consider using Excel's Data Model feature to construct PivotTables that use multiple tables for source data. If you are using Excel 2010, you can access data models through Microsoft's **Power Pivot** add-in feature. The Power Pivot add-in is an extremely valuable tool for creating advanced PivotTables, including those that draw data from multiple sources such as two or more Excel tables. Though a full discussion and demonstration of using Power Pivot to create PivotTables from multiple tables is beyond the scope of this course, know that you can use this tool to create PivotTables from multiple tables, regardless of whether those tables are stored in the same workbook or in different workbooks. For more information on Power Pivot, we encourage you to visit www.powerpivot.com.

If you are using Excel 2013 or Excel 2016, you can access data models through Power Pivot, just as you can in Excel 2010. However, Excel 2013 and Excel 2016 also expressly allow you to access data models using other techniques, including by clicking **Manage Data Model** on the **Data** tab of the Ribbon. You will learn more about how to manage and utilize data models beginning in Chapter Six.

Using Tables as Sources for PivotCharts

PivotCharts are simply graphical extensions of PivotTables. That is, PivotCharts just represent pictorially the same information that is in a PivotTable. Because tables can serve as data sources for PivotTables, they can also serve as data sources for PivotCharts.

You can build a PivotChart based on table data using one of two methods. First, if you have already built your PivotTable based on table data, you can build the chart from the PivotTable data. To do so, click on the PivotTable and from the **PivotTable Tools Analyze**¹ contextual tab, choose **PivotChart**. Next, select the chart type and click **OK**. Upon doing so, Excel constructs the PivotChart from the PivotTable. In this example, because the PivotTable used table data as its data source, the PivotChart is based on that same table data.

A second approach to building a PivotChart from table data is to build the PivotChart and the PivotTable at the same time using table data as the data source.

¹ In Excel 2007 and 2010, go to PivotTable Tools Options to begin this process.

	A	B	C	D
1	Salesperson	Region	Sales	Commission
2	Adams	East	14,246,779	170,961
3	Boskie	West	14,179,405	283,588
4	Goldberg	North	14,665,352	278,642
5	Gzehoviak	South	18,922,588	359,529
6	Jones	East	19,407,851	291,118
7	Kennedy	West	12,922,032	245,519
8	Ling	North	17,066,719	238,934
9	Martinez	South	13,906,832	250,323
10	Roberts	East	14,726,733	176,721
11	Rodriguez	West	18,509,898	333,178
12	Toner	North	14,141,251	169,695
13	Woo	South	17,967,609	305,449
14	Total		190,663,049	3,103,657

Figure 60 - Table Data for PivotChart

Given the table pictured in **Figure 60**, perform the following steps to prepare a PivotChart simultaneously with a PivotTable.

1. From the **Insert** tab of the **Ribbon**, click the arrow below **PivotChart** and choose **PivotChart & PivotTable** to open the **Create PivotTable with PivotChart** dialog box pictured in **Figure 61**.

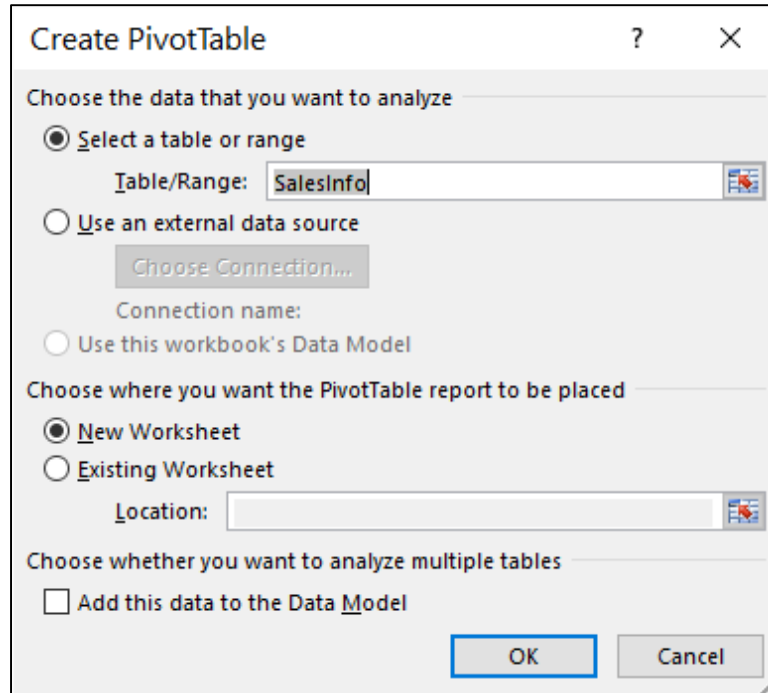


Figure 61 - Create PivotTable with PivotChart Dialog Box

2. Excel will default to entering the name of the active table in the **Table/Range:** box.
3. Select the location for the PivotTable and PivotChart. The default will be **New Worksheet**.
4. Click **OK**.
5. In the resulting window shown in **Figure 62**, click and drag the fields from the **Field List** to the appropriate PivotTable quadrants to build the PivotTable and PivotChart.

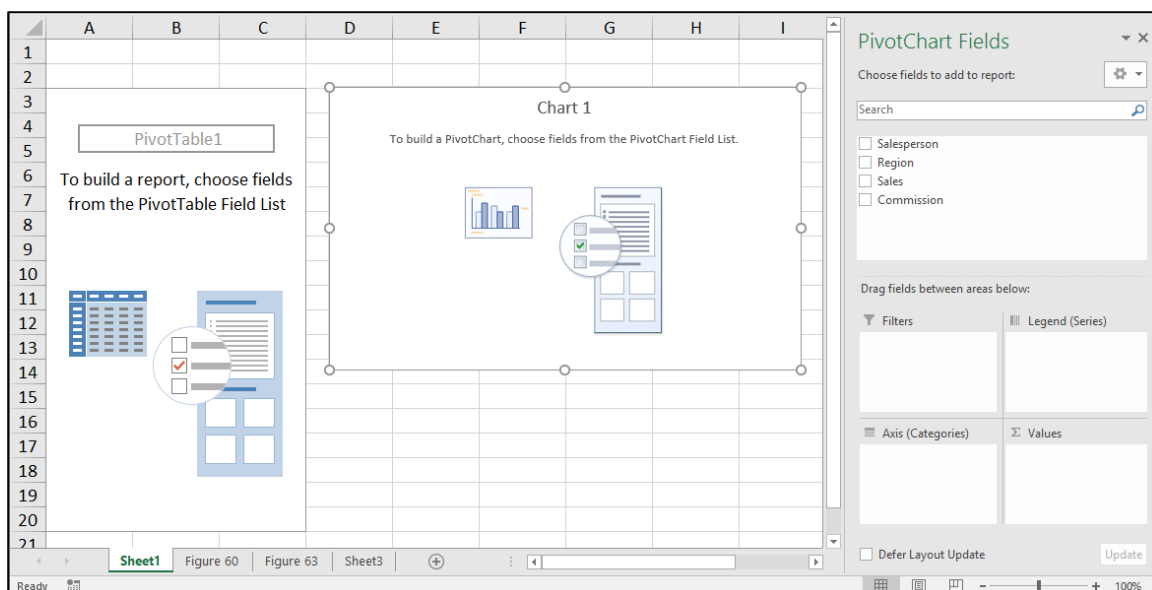


Figure 62 - Building a PivotTable and PivotChart Simultaneously

The resulting PivotChart might resemble that shown in **Figure 63**. Notice the presence of the on-screen filters in this PivotChart, indicating that Excel 2010 or newer created the PivotChart. If you choose to utilize the filters, they will affect both the PivotChart and the related PivotTable; however, they will have no bearing on the underlying table data.

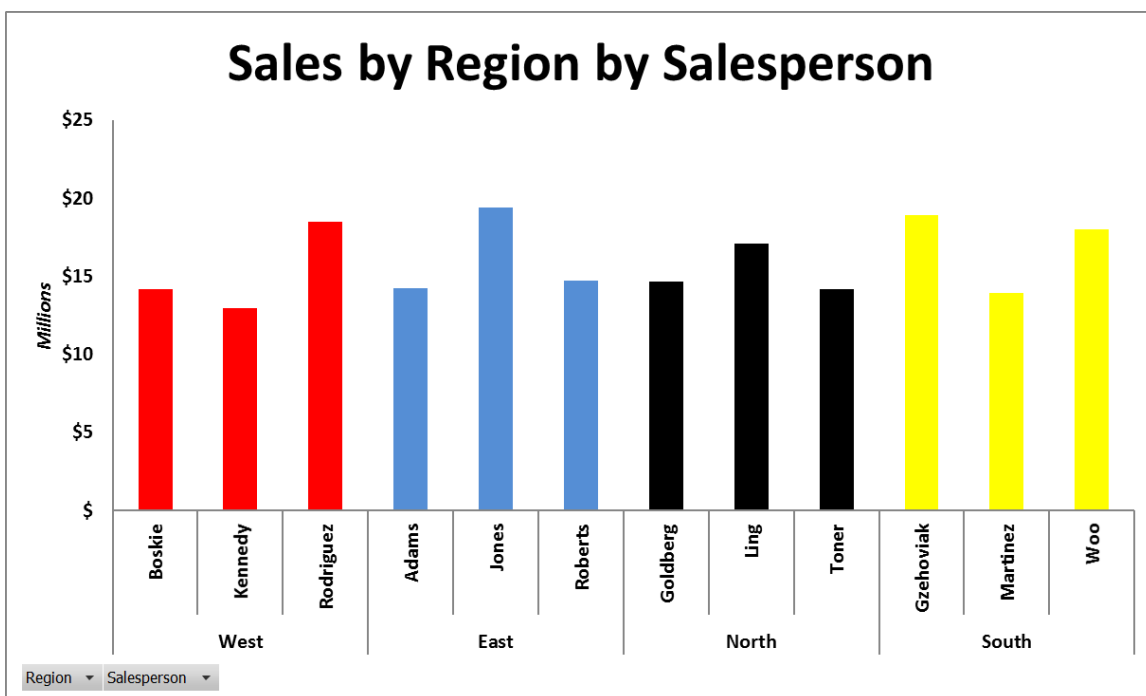


Figure 63 - PivotChart Created from Table Data

Advanced Table Techniques

In previous chapters, we have covered some of the extraordinary features of tables. When you are working with tables, you will no doubt on occasion face out-of-the ordinary circumstances associated with tables, their features, and their limitations. In this chapter, we cover various tips and techniques to assist you in handling these types of circumstances.

Learning Objectives

Upon completing this chapter, you should be able to:

- Create formulas outside of tables that refer to data inside tables, including VLOOKUP formulas;
- Build multi-worksheet and multi-workbook formulas that reference tables;
- Generate VLOOKUP formulas that analyze table data;
- Utilize shortcuts to select table rows and columns and to insert table rows and columns;
- Disable unwanted table functionality; and
- Use Excel's Fuzzy Lookup add-in with tables.

Creating Formulas That Refer to Table Data

As powerful as they are, tables will certainly not be the only Excel feature you use. Moving forward, you will most likely begin using tables to store data – potentially large volumes of data – but you may find that you need to build formulas outside a table that reference data inside a table. Creating such formulas is relatively easy, if you understand a few simple rules first.

Building VLOOKUPS on Table Data

A need to build formulas outside of a table that reference formulas inside a table will probably emerge for many professionals in the form of VLOOKUP formulas. VLOOKUP formulas examine the first column data range for an exact or approximate match and return a value from a specified column in the same row where the match or approximate match occurs. In Excel, VLOOKUP formulas use the syntax of

=VLOOKUP(Lookup_value, Table_array, Col_index_num, Range_lookup)

where

- **Lookup_value** is the value to be found in the first column of the list of data. Lookup_value can be a value, reference, or a string.
- **Table_array** is the range of data, numbers, or logical values which are analyzed by VLOOKUP and from which VLOOKUP retrieves data. Table_array can be a simple range of data, a defined name, or a table.
- **Col_index_num** is the column number that contains the data to be returned by the formula.
- **Range_lookup** is a logical value of TRUE or FALSE. If Range_lookup is set to FALSE, then the VLOOKUP must find an exact match, or no data is returned. If Range_lookup is set to TRUE or is omitted, the VLOOKUP returns data based on an approximate match.

Before building VLOOKUPS that operate with tables, let us quickly review two basic VLOOKUP examples. In the first example, pictured in **Figure 64**, a VLOOKUP formula searches a database of employee last names, and when it finds an exact match, it returns the amount of the employee's bonus for the previous quarter. The syntax of the formula used in this case is **=VLOOKUP(B1,A6:D18,4,FALSE)**. Notice the formula uses the FALSE argument at the end to force an exact match of the data.

B2					
	A	B	C	D	E
1	Enter Employee Last Name>>>	Jimenez			
2	Employee Bonus>>>	\$ 1,500.00			
3					
4					
5	Employee Last Name	Employee First Name	Department	Bonus	
6	Baumen	Jason	Production	\$ 1,500.00	
7	Bolinski	Heather	Sales	\$ 1,500.00	
8	Harris	Frank	Accounting	\$ 2,000.00	
9	Hendro	Kirby	Sales	\$ 2,000.00	
10	Jimenez	Doug	Production	\$ 1,500.00	
11	Jones	Bob	Sales	\$ 5,000.00	
12	Mason	Bradley	Sales	\$ 500.00	
13	Miller	Lucas	Sales	\$ 1,000.00	
14	Schneider	Eloise	Accounting	\$ 4,000.00	
15	Smith	Ann	Production	\$ 1,500.00	
16	Valdez	Suzie	Production	\$ 1,000.00	
17	Washington	Adel	Production	\$ 3,000.00	
18	Young	Orestus	Accounting	\$ 2,500.00	
19					

Figure 64 - VLOOKUP Formula with Exact Match Returning Employee Bonus Amount

Shown in **Figure 65**, the second VLOOKUP example uses a series of approximate match VLOOKUP formulas to calculate how much income tax is due based on the tax table in cells D4:F11. Notice that each formula omits the optional fourth argument, which defaults the formulas to approximate match formulas.

B6					
	A	B	C	D	E
1					
2					
3					
4	Taxable Income	\$136,425			
5					
6	Estimated Income Tax	\$36,456			
7					
8					
9					
10					
11					
12					

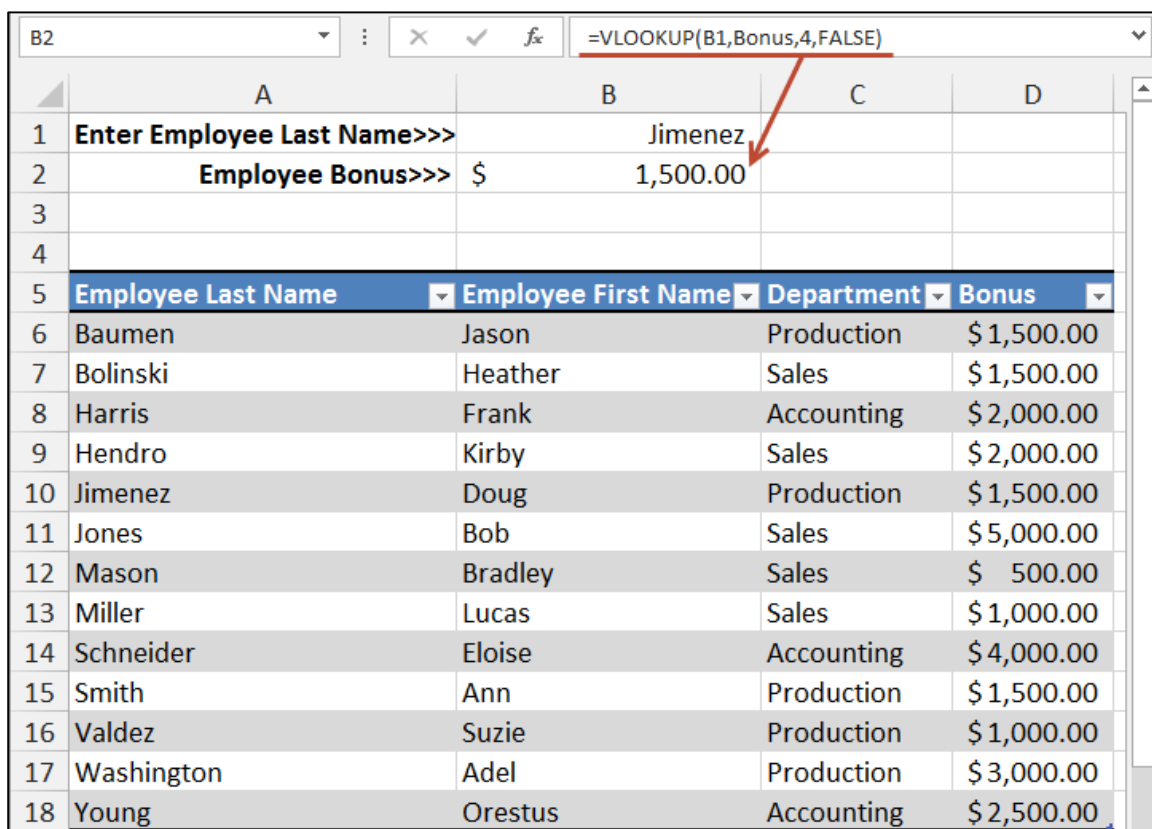
Figure 65 - VLOOKUP Formula with Approximate Match Returning Income Tax Due

Neither of the first two illustrations utilizes tables in any capacity; they serve only as a primer/refresher on general VLOOKUP formulas. Now that we have a fundamental understanding of how VLOOKUP formulas work, let us proceed with two examples that integrate VLOOKUP and tables. In the first example, we will use a VLOOKUP formula to query a table and return a value from the table, while the second example will place the VLOOKUP formula within a table.

VLOOKUP Querying Table Data

A VLOOKUP formula that queries data from a table is remarkably similar to a traditional VLOOKUP formula; the only notable difference will be that the VLOOKUP substitutes the table name for the *Table_array* described previously.

To illustrate, let us now assume that the data pictured in Figure 64 is converted to a table and is assigned the name *Bonus*. As shown in **Figure 66**, the VLOOKUP formula to identify the amount owed to a given employee is exactly the same as in Figure 64, except for the fact that *Bonus* is substituted for the cell reference *A6:D18*.

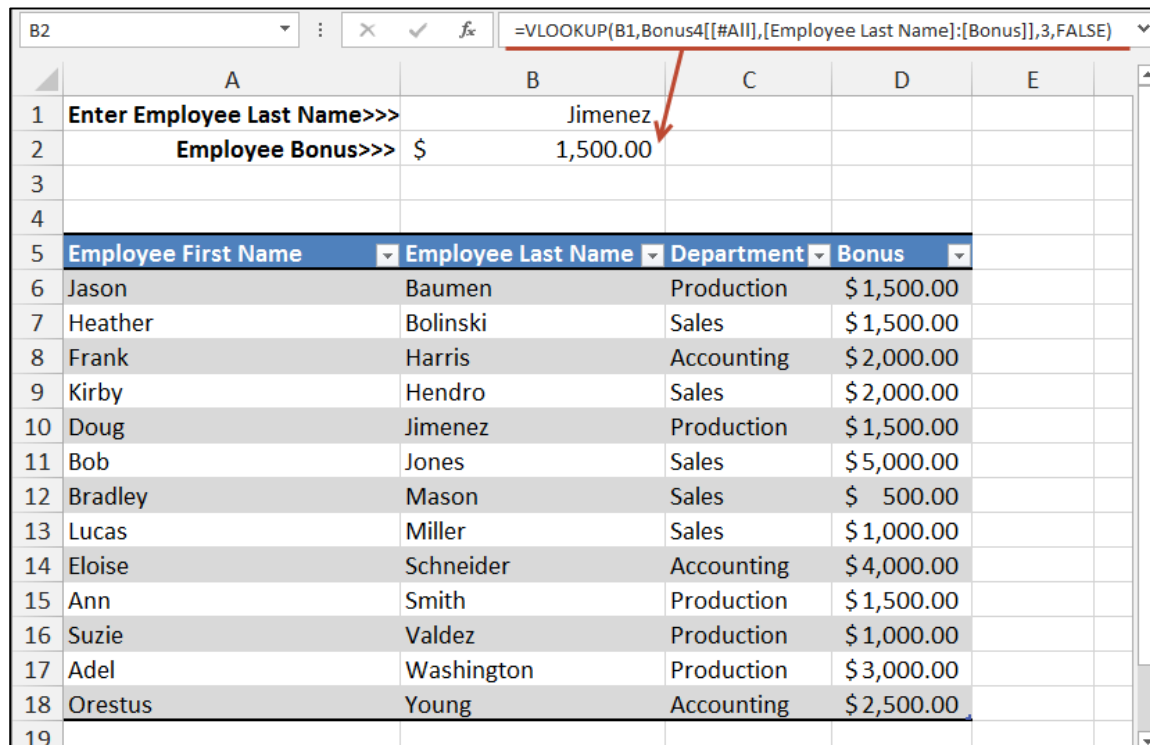


	A	B	C	D
1	Enter Employee Last Name>>>	Jimenez		
2	Employee Bonus>>>	\$ 1,500.00		
3				
4				
5	Employee Last Name	Employee First Name	Department	Bonus
6	Baumen	Jason	Production	\$ 1,500.00
7	Bolinski	Heather	Sales	\$ 1,500.00
8	Harris	Frank	Accounting	\$ 2,000.00
9	Hendro	Kirby	Sales	\$ 2,000.00
10	Jimenez	Doug	Production	\$ 1,500.00
11	Jones	Bob	Sales	\$ 5,000.00
12	Mason	Bradley	Sales	\$ 500.00
13	Miller	Lucas	Sales	\$ 1,000.00
14	Schneider	Eloise	Accounting	\$ 4,000.00
15	Smith	Ann	Production	\$ 1,500.00
16	Valdez	Suzie	Production	\$ 1,000.00
17	Washington	Adel	Production	\$ 3,000.00
18	Young	Orestus	Accounting	\$ 2,500.00

Figure 66 - VLOOKUP Formula Referring to a Table

Keeping in mind that VLOOKUPS always operate on the first column of data in the *Lookup_array*, what happens if the data in the table is not positioned in a fashion conducive to VLOOKUPS? For instance, what if the relative position of the *Employee Last Name* and *Employee First Name* fields in the example above is reversed? As shown in **Figure 67**, you can

use structured referencing to build a VLOOKUP formula that operates on the second through fourth columns of the table and excludes the first column.



	A	B	C	D	E
1	Enter Employee Last Name>>>	Jimenez			
2	Employee Bonus>>>	\$ 1,500.00			
3					
4					
5	Employee First Name	Employee Last Name	Department	Bonus	
6	Jason	Baumen	Production	\$ 1,500.00	
7	Heather	Bolinski	Sales	\$ 1,500.00	
8	Frank	Harris	Accounting	\$ 2,000.00	
9	Kirby	Hendro	Sales	\$ 2,000.00	
10	Doug	Jimenez	Production	\$ 1,500.00	
11	Bob	Jones	Sales	\$ 5,000.00	
12	Bradley	Mason	Sales	\$ 500.00	
13	Lucas	Miller	Sales	\$ 1,000.00	
14	Eloise	Schneider	Accounting	\$ 4,000.00	
15	Ann	Smith	Production	\$ 1,500.00	
16	Suzie	Valdez	Production	\$ 1,000.00	
17	Adel	Washington	Production	\$ 3,000.00	
18	Orestus	Young	Accounting	\$ 2,500.00	
19					

Figure 67 - VLOOKUP Formula Identifying Data Not Positioned in the First Column of a Table

VLOOKUP inside a Table

If you build a VLOOKUP formula inside a table to reference data outside a table, no special considerations are necessary, other than those for structured references to data in the table. To illustrate, consider the example shown in **Figure 68**. In it, a VLOOKUP in the table formula refers to data external to the table to identify and insert the Department Name into the table based on the Department Number, which is already stored in the table. Again, other than considerations for structured references, this formula is the same as any VLOOKUP formula that would have been constructed outside the table.

	A	B	C	D	E
1					
2	Employee First Name	Employee Last Name	Dept #	Dept Name	Bonus
3	Jason	Baumen	1	Production	\$ 1,500.00
4	Heather	Bolinski	2	Sales	\$ 1,500.00
5	Frank	Harris	3	Accounting	\$ 2,000.00
6	Kirby	Hendro	2	Sales	\$ 2,000.00
7	Doug	Jimenez	1	Production	\$ 1,500.00
8	Bob	Jones	2	Sales	\$ 5,000.00
9	Bradley	Mason	2	Sales	\$ 500.00
10	Lucas	Miller	2	Sales	\$ 1,000.00
11	Eloise	Schneider	3	Accounting	\$ 4,000.00
12	Ann	Smith	1	Production	\$ 1,500.00
13	Suzie	Valdez	1	Production	\$ 1,000.00
14	Adel	Washington	1	Production	\$ 3,000.00
15	Orestus	Young	3	Accounting	\$ 2,500.00
16					
17					
18			1	Production	
19			2	Sales	
20			3	Accounting	
21					

Figure 68 - VLOOKUP Formula inside a Table

Referencing Tables across Multiple Worksheets and Workbooks

The topic of unqualified and fully qualified table references was discussed beginning on page 39. To recap that discussion, an unqualified reference does not include the table name of data that is referenced as a structured reference, whereas a fully qualified reference does include the table name of data that is referenced as a structured reference. When using structured references to refer to data within the same table, you need not utilize a fully qualified reference. If the data to which you are referring is in a different table in the same workbook, you must use a fully qualified reference that includes the name of the table; the fully qualified reference need not include the name of the workbook.

If you wish to refer to data in another table located in a separate workbook, you must use a fully qualified reference that includes the name of the “source” workbook. Further, you must enclose the name of the workbook, including file extension, in single quotation marks and follow it with an exclamation point as shown in the following structured reference formula which presents a fully qualified reference, including the name of the external workbook.

= 'Table Headers.xlsx'!Table1[[#Headers],[Acquisition Date]]

Linked Workbooks and Structured References

If you link from one workbook to an Excel table in another workbook, you must have the *source* workbook open in order to avoid #REF errors in the *destination* workbook. However, if you open the destination workbook without the source workbook being open, the errors that appear should resolve when you subsequently open the source workbook.

Selecting Rows and Columns in Tables

As objects on worksheets, tables provide some unique features when selecting a row or column of data within a table. For example, if you wish to select a row of data within the table, position your mouse pointer on the left edge of the left-most cell in the row. Upon doing so, your mouse pointer changes to a solid arrow. At that point, click the mouse, and you will select the entire row. For those who prefer keyboard shortcuts, pressing **Shift + Spacebar** selects the current row of the table, regardless of which cell the mouse is positioned in that row.

Similar to selecting a row, to select an entire column of the table, position your mouse pointer near the top of the topmost cell in a column. When the mouse pointer changes to a solid arrow, click the mouse to select the entire column. Alternatively, pressing **CTRL + Spacebar** selects an entire column of table data, except for the header and total row (if present).

If you wish to select the entire table, position your mouse near the top, left corner of the top, left cell in the table. Upon doing so, the mouse pointer will change to a diagonal arrow. When it does so, click the mouse to select the entire table. To use a keyboard shortcut to select an entire table, press **CTRL + A** with the table selected.

Custom Views Unavailable When Working with Tables

Custom Views are a seldom-used feature of Excel, but those who do utilize Custom Views appreciate the functionality offered by this tool. Among the productivity enhancements associated with Custom Views are the feature's ability to memorize hidden column and hidden row settings, filter settings, and print settings for quick recall. Unfortunately, when working in a *workbook* that contains a table, Excel disables Custom Views for all worksheets within that workbook. If, however, you subsequently convert the table back to a range of data, Custom Views become available in all worksheets in that workbook.

Excel Disables Shared Workbook Functionality When Working with Tables

As with Custom Views, Excel disables the **Shared Workbook** functionality when a table resides in a workbook. Thus, using Shared Workbook mode to enable multiuser, simultaneous collaboration on the same workbook is not possible when a table exists in a workbook.

Disabling Specific Table Features

Suppose you create a table in Excel but do not wish it to possess auto-expansion and formula replication features. For instance, perhaps you would like to confirm that data appended to the

bottom of the table by one of your team members is correct before the table includes that data in structured reference formulas using the table as the data source or before a PivotTable update incorporates all of the newly added data. You can disable one or both of these features when you decide that it is appropriate to do so; likewise, at any point in the future, you can enable one or both of these features.

To do so, from the **Ribbon**, click **File**, **Options**, **Proofing**, and **AutoCorrect Options**. In the resulting **AutoCorrect** dialog box, click the **AutoFormat As You Type** tab and uncheck the boxes labeled **Include new rows and columns in table** and **Fill formulas in tables to create calculated columns**. Figure 69 presents an example of the AutoCorrect dialog box with both of these options disabled.

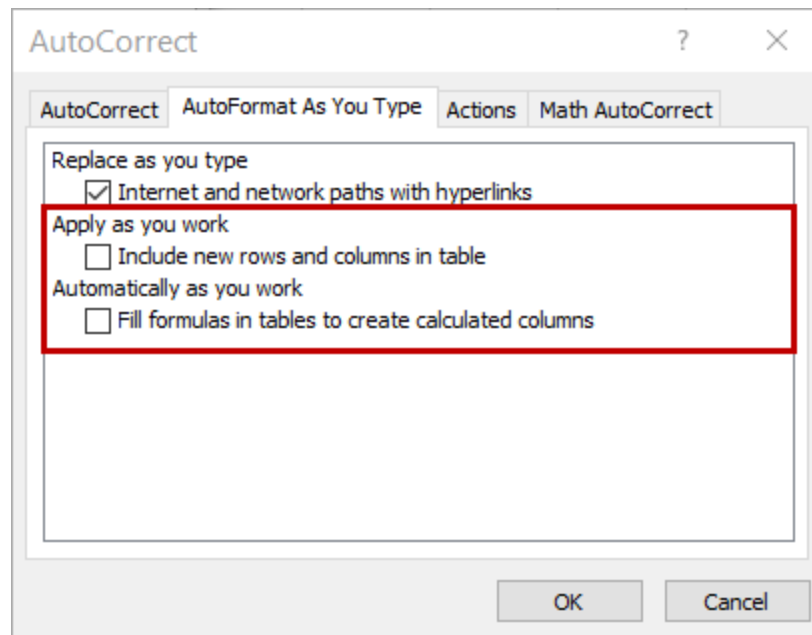


Figure 69 - Disabling Auto-expansion and Formula Replication Features for Tables

Using Excel's Fuzzy Lookup Add-In

Sometimes when using Excel's VLOOKUP and HLOOKUP functions, we do not get the results we are seeking, not because of anything we are doing wrong, but because the data we are trying to match together is not an exact match. For example, if you are trying to perform a VLOOKUP to find *123 Main Street*, Excel will not match that to *123 Main St*. Fortunately, for those who need to match inexact data, Microsoft offers the **Fuzzy Lookup Add-in** to provide a means of matching inexact data stored in tables.

Download the Add-in

Before you can use the Fuzzy Lookup Add-in for Excel, you must download it from Microsoft and install it on your computer. You can do this by visiting <http://www.microsoft.com/en-us/download/details.aspx?id=15011> and following the instructions provided by Microsoft. Note that as part of the download package, Microsoft also provides you with a sample data set you can use for practice; however, the example presented below uses a different data set.

Creating a Fuzzy Lookup

Keep in mind that the Fuzzy Lookup Add-in matches inexact textual data and allows you to join data from multiple tables into one output range. Therefore, for your Fuzzy Lookup to work, you must first convert your source data ranges to Excel tables.

Figure 70 presents two tables we will use for the basis of our Fuzzy Lookup. **SSN** is the name of the table on the left, and **Comp** is the name of the table on the right. We would like to match the *Name* field in SSN to the *Name* field in Comp and create a results list that shows employees' names, Social Security numbers, and compensation. The challenge we face is that not all of the names are spelled and arranged the same way; therefore, if we attempt to use a VLOOKUP function to complete this task, we will not be successful. For example, consider the name *Jan Kotas* on row 4 of SSN. In looking at row 8 of Comp, we see that what should be the matching name is arranged as *Kotas, Jan*; thus, a VLOOKUP function would not match the two names together. Likewise, *Robert Zare* on row 8 and *Thomas Axen* on row 13 of SSN are not exact matches to *Robert Zare, III* on row 11 and *T G Axen* on row 17 of Comp, again presenting problems if we try to match based on traditional methods.

	A	B	C	D	E
1	Name	SSN		Name	Compensation
2	Nancy Freehafer	111-11-1111		Jean Philippe Bagel	132,294
3	Andrew Cencini	111-11-1112		Ming-Yang Xie	131,754
4	Jan Kotas	111-11-1113		Anne Hellung-Larsen	131,622
5	Mariya Sergienko	111-11-1114		O'Donnell, Martin	131,306
6	Steven Thorpe	111-11-1115		Carlos Grilo	128,691
7	Michael Neipper	111-11-1116		Catherine Autier Miconi	127,624
8	Robert Zare	111-11-1117		Kotas, Jan	124,021
9	Laura Giusani	111-11-1118		Mikael Sandberg	118,776
10	Anne Hellung-Larsen	111-11-1119		Cornelia Weiler	118,388
11	Anna Bedecs	111-11-1120		Robert Zare, III	117,838
12	Antonio Gratacos Solis	111-11-1121		Johnathan Edwards	115,046
13	Thomas Axen	111-11-1122		Francisco Pérez-Olaeta	114,247
14	Christina Lee	111-11-1123		L T Sousa	113,109
15	Martin O'Donnell	111-11-1124		Ludick, Jr., Andre	112,387
16	Francisco Pérez-Olaeta	111-11-1125		Run Liu	112,022

Figure 70 - Tables for Fuzzy Lookup

However, we can match the data in these two tables using the Fuzzy Lookup Add-in. To do so, click **Fuzzy Lookup** on the **Fuzzy Lookup** tab of the Ribbon to open the Fuzzy Lookup task pane on the right side of the window shown in **Figure 71**. Upon doing so, Excel automatically senses and inserts the names of the tables into the Fuzzy Lookup task pane. Excel also automatically analyzes the columns in the tables and joins the tables if a column in one table has the same column header as a column in the other table. Of course, you can edit both the table and column selections, if necessary.

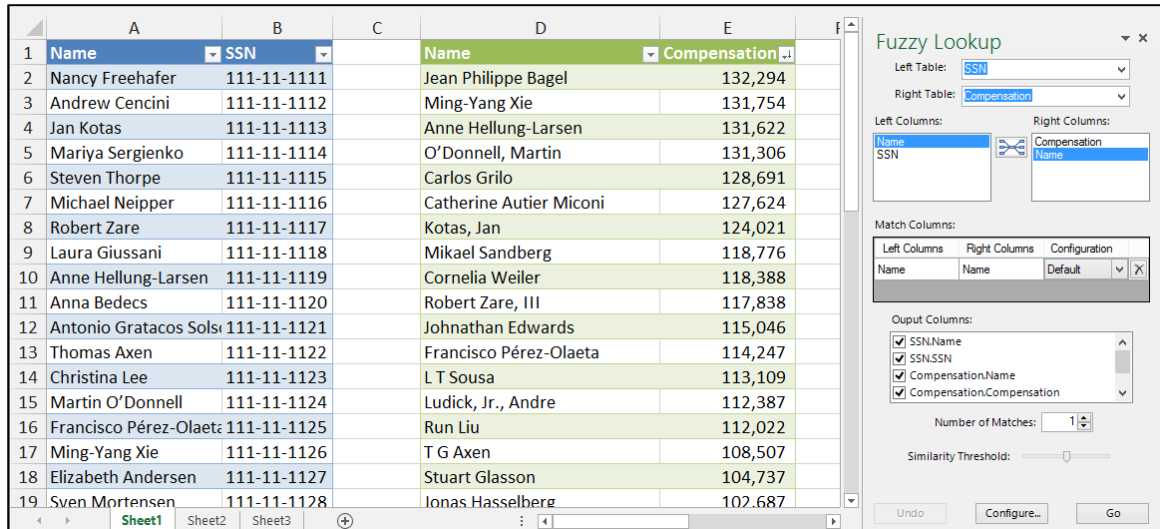


Figure 71 - Creating a Fuzzy Lookup in Excel

In the **Output Columns** section, check the box next to each field you want included in your results list. In this example, we are indicating that we want the **Name** and **SSN** fields from the **SSN** table and the **Compensation** field from the **Compensation** table to be included in the output. We are also indicating that we want the system-generated *Similarity* score to be included. The *Similarity* score provides a measure of how “confident” Excel is in its matching of inexact data. With these processes complete, all that is left to do is click **Go**. Upon doing so, the Fuzzy Lookup Add-in analyzes the data in the two tables and provides the results list pictured in **Figure 72**.

Name	SSN	Compensation	Similarity
Nancy Freehafer	111-11-1111	73977	0.9455
Andrew Cencini	111-11-1112	70353	0.9455
Jan Kotas	111-11-1113	124021	1.0000
Mariya Sergienko	111-11-1114		
Steven Thorpe	111-11-1115	79014	0.9455
Michael Neipper	111-11-1116	78288	1.0000
Robert Zare	111-11-1117	117838	0.9333
Laura Giussani	111-11-1118	93669	1.0000
Anne Hellung-Larsen	111-11-1119	131622	1.0000
Anna Bedecs	111-11-1120	80582	1.0000
Antonio Gratacos Solsona	111-11-1121	66270	1.0000
Thomas Axen	111-11-1122		

Figure 72 - Sample Fuzzy Lookup Results

A close inspection of the data in Figure 72 reveals that the tool worked well, though not flawlessly. In particular, it did not find matches for Mariya Sergienko and Thomas Axen. Returning to the Fuzzy Lookup task pane, let us modify the **Similarity Threshold** setting to see if reducing the confidence level slightly increases the number of matches. To do this, drag the Zoom Slider to the left so that a smaller value – in this case, 0.4 – is in place.

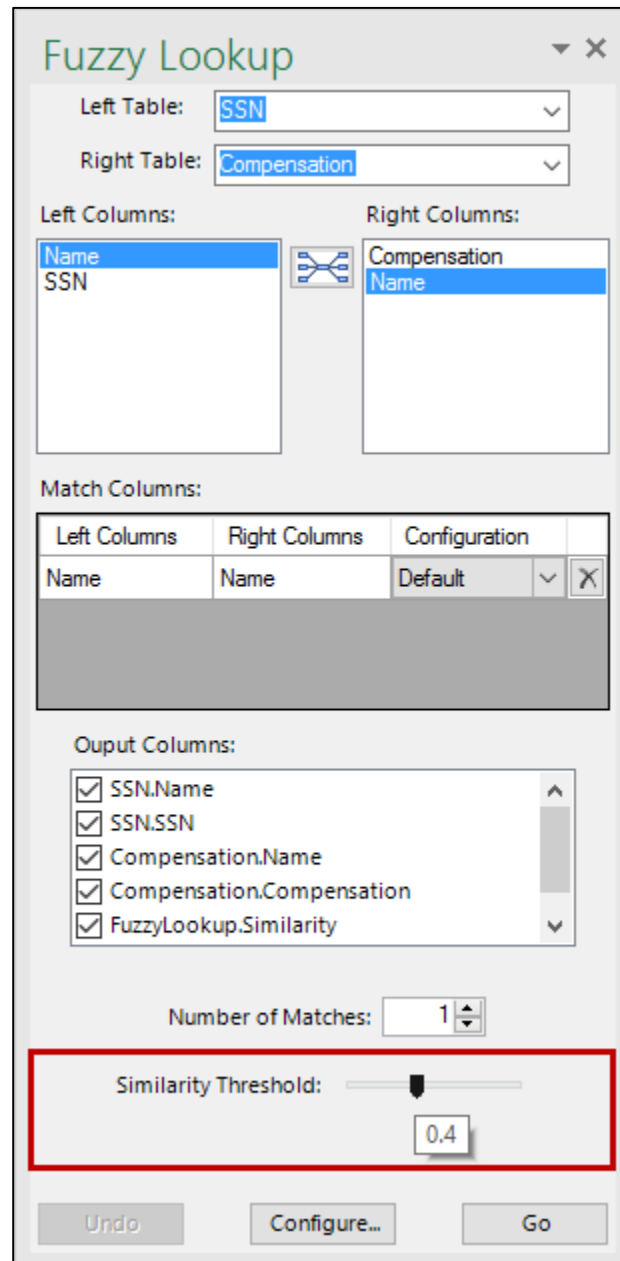


Figure 73 - Modifying the Similarity Threshold for a Fuzzy Lookup

After modifying the Similarity Threshold, click **Go** to generate a new set of results. Upon doing so, as shown in **Figure 74**, the Fuzzy Lookup Add-in matches all of the data in the two tables and does so without any errors.

G	H	I	J	K
Name	SSN	Compensation	Similarity	Similarity
Nancy Freehafer	111-11-1111	73977	0.9455	0.9455
Andrew Cencini	111-11-1112	70353	0.9455	0.9455
Jan Kotas	111-11-1113	124021	1.0000	1.0000
Mariya Sergienko	111-11-1114	70211	0.8811	0.8811
Steven Thorpe	111-11-1115	79014	0.9455	0.9455
Michael Neipper	111-11-1116	78288	1.0000	1.0000
Robert Zare	111-11-1117	117838	0.9333	0.9333
Laura Giussani	111-11-1118	93669	1.0000	1.0000
Anne Hellung-Larsen	111-11-1119	131622	1.0000	1.0000
Anna Bedecs	111-11-1120	80582	1.0000	1.0000
Antonio Gratacos Solsona	111-11-1121	66270	1.0000	1.0000
Thomas Axen	111-11-1122	108507	0.8411	0.8411

Figure 74 - Completed Fuzzy Lookup Example

Fuzzy Lookup Summary

When faced with the task of matching inexact data, including names and addresses, many Excel users expend vast amounts of time cleaning up the data before using traditional techniques such as VLOOKUP. With the Fuzzy Lookup Add-in for Excel, that need not be the case. By downloading and installing this free tool from Microsoft, you now have an extremely powerful and efficient means for matching data when perfect matches do not exist.

For a video demonstration of this tip, please visit www.tinyurl.com/k2tips121.

Introducing Excel's Data Model Feature

Beginning with the Power Pivot add-in made available in Excel 2010, Microsoft added the Data Model feature to Excel. However, with the 2013 release of Excel and continuing with the 2016 release, the feature became accessible to all users, not just those running Power Pivot. Data models can be real “game changers,” and, in this chapter, you will learn how to construct data models and build simple analytical tools that use a data model as their data source.

Learning Objectives

Upon completing this chapter, you should be able to:

- Define the term *data model* and identify situations where using data models can be advantageous;
- Create data models from multiple tables in an Excel workbook;
- Add, modify, and delete simple relationships between multiple tables in a data model; and
- Create PivotTables based on data models.

What Is a Data Model?

A data model is a collection of two or more tables that you *relate* or link together based on the presence of one or more common fields to create what Excel interprets as a single source of data². If you are familiar with the concept of a relational database management system (RDBMS) such as Access or SQL Server, the concept of a data model should be one that you easily recognize. Once you create a data model, you can use the data in the model in PivotTables

² Technically, a data model can consist of a single table. Reasons for building a single-table model could include, for example, wanting to use Power Pivot's KPI features on the data in a single table. However, because most of the power associated with using data models vests with multi-table data models, we will focus on that approach in this course.

(including those built using Power Pivot), PivotCharts, Power View, and 3D Maps (formerly known as Power Maps).

A workbook can contain only one data model. Because of the tremendous analytical capabilities enabled through data models, it can be tempting to load extremely large data sets into a data model. Recognize, however, that while the volume of data you can load into a data model is virtually unlimited, you will always be bound by the overall limit on the size of an Excel workbook. Currently, if you are running a 32-bit version of Excel 2013 or Excel 2016, that limit is 2 GB. However, if you are running a 64-bit version of Excel 2013 or Excel 2016, the size of the workbook is limited only by the amount of memory and system resources you have on your computer. If you find that the volume of data with which you are attempting to work in your data model is problematic, you may consider installing Microsoft's **Workbook Size Optimizer**; it is available at <https://www.microsoft.com/en-us/download/details.aspx?id=38793>. In some cases, the Optimizer can enable better data compression and enhance performance.

Conceptually, a data model might appear as shown in **Figure 75**. In this simplified illustration, three tables are joined together to create a single data model. You can use the data stored in the data model to build analytical tools such as PivotTables and Power View analyses.

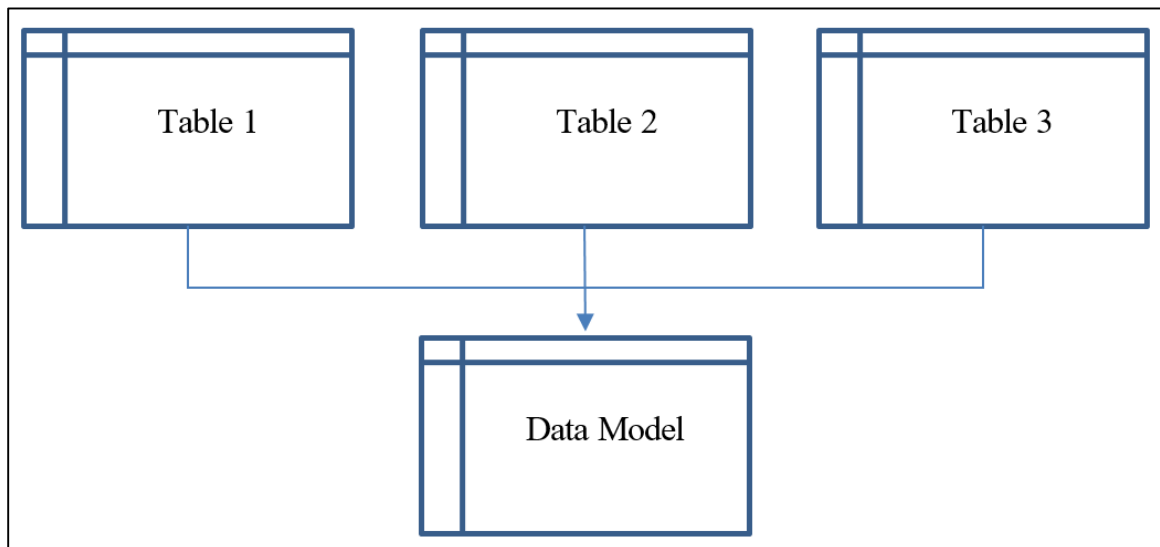


Figure 75 - Schematic of an Excel Data Model

Advantages of Working with Data Models

Often, the data that you need to analyze and summarize is stored in different Excel objects, including tables. For example, beginning on page 48 of this text, we discussed how you could use a series of formulas that incorporate Excel's VLOOKUP function to link together from multiple tables so that you could build a PivotTable from that data. While that process certainly works, as you will see, using a data model to link together that data is a far simpler and easier approach to implement.

In addition to simplicity, data models open up new functionality in Excel. For example, if you build a PivotTable from a data model, then you have access to **Quick Explore**, a means of drilling down and around through all of the fields in the data model, even those that are not part of the PivotTable report. **Figure 76** demonstrates this process.

The figure illustrates the Quick Explore feature in Excel, showing how to drill down into a PivotTable to explore related data fields. The top screenshot shows a PivotTable with 'Product Class' as the filter and 'Bicycle' selected. The 'Quick Explore' pane is open, showing a tree view of fields. A red box highlights 'Drill To Product Name' with a red arrow pointing to the 'Product Name' field in the list. The middle screenshot shows the 'Quick Explore' pane with 'Mini Nicros' selected. A red box highlights 'Drill To Customer Name' with a red arrow pointing to the 'Customer Name' field in the list. The bottom screenshot shows the resulting PivotTable after drilling down to 'Mini Nicros' and 'Customer Name'. The table has 'Product Class' and 'Product Name' as filters, and 'Sum of calcExtension' as the value field. The columns are 'Q1', 'Q2', 'Q3', 'Q4', and 'Grand Total'. The rows list various bicycle products and their sales data.

Product Class	Product Name	Q1	Q2	Q3	Q4	Grand Total
Alley Cat Cycles					281.85	281.85
Belgium Bike Co.				1,071.04		1,071.04
Benny - The Spokes Person				845.55		845.55
Bikes for Tykes		845.55				845.55
Bikes, Bikes, and More Bikes		845.55				845.55
Blazing Saddles		563.70			563.70	1,127.40
Canal City Cycle				803.28		803.28
C-Gate Cycle Shoppe				253.67		253.67
Changing Gears		281.85				281.85
City Cyclists		803.28				803.28
Crank Components			845.55			845.55
Crazy Wheels		281.85	845.55			1,127.40
Cycles and Sports			845.55			845.55
Cyclopath				845.55		845.55
Furia				563.70		563.70
Hercules Mountain Bikes				563.70		563.70

Figure 76 - Using the Quick Explore Feature Associated with PivotTables Constructed from Data Models

Additionally, if you build a PivotTable from a data model, you can convert the PivotTable to an ordinary range of data, while still retaining the full calculation capabilities of the PivotTable.

This allows you to insert rows and columns into the PivotTable, and even to click and drag data in the PivotTable to rearrange it to your exacting specifications.

Further, if you relate multiple tables together as a data model, you can take advantage of tools such as Power View and 3D Maps to summarize the data into interactive reporting dashboards such as the one pictured in **Figure 77**.

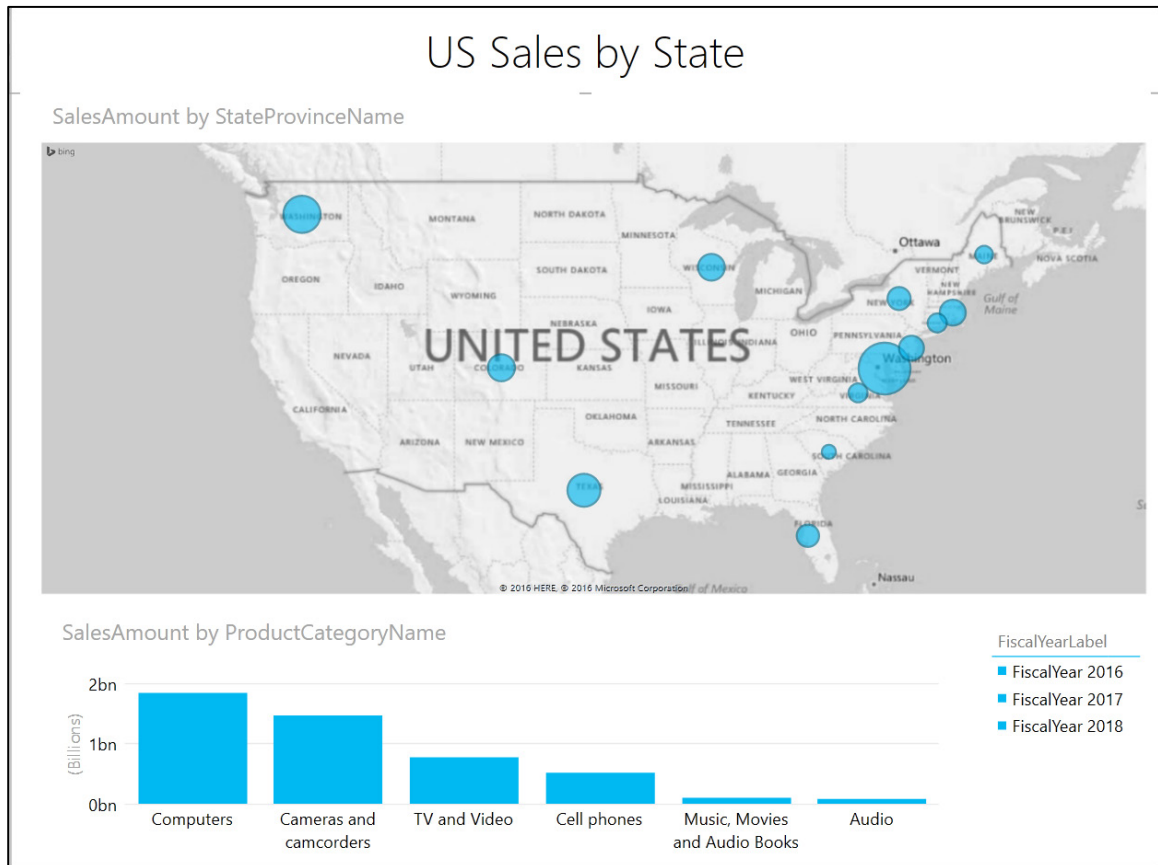


Figure 77 - Interactive Dashboard Created in Power View from a Data Model

Finally, because data models compress data, you can work with significantly larger volumes of data in a data model than you can with ordinary Excel worksheets. In other words, you are no longer bound by the 1,048,576 row limitation in an Excel spreadsheet.

Creating Simple Data Models from Excel Tables

Creating data models in Excel can be quite easy; in fact, you can do so with a single click of the mouse. However, before you attempt to relate multiple Excel tables, you should first verify that each table has a field that is common to the other table(s) to which you wish to establish the relationship. For example, suppose you are attempting to relate two tables containing inventory data; in this case, you would probably look for some type of part number or SKU code that is common to each table. Likewise, if you desire to relate multiple employee-related tables, you would likely do so based on a common field of employee identification numbers or a similar field.

Our first example of relating tables together to create a data model will be in the context of creating a PivotTable. In this example, we have an Excel workbook that contains two tables – *Transactions* and *Product*. We would like to build a PivotTable that uses data from both tables, but we do not want to use the VLOOKUP approach discussed earlier to consolidate the data from the two tables. Fortunately, we can use data models to solve the problem.

Begin by clicking on either of the tables. From the **Table Tools Design** tab of the Ribbon, click **Summarize with PivotTable** to open the **Create PivotTable** dialog box pictured in **Figure 78**. In this dialog box, check the box next to **Add this data to the Data Model**. Importantly, checking this box adds *all* tables in the workbook – not just the currently-selected table – to the workbook's data model.

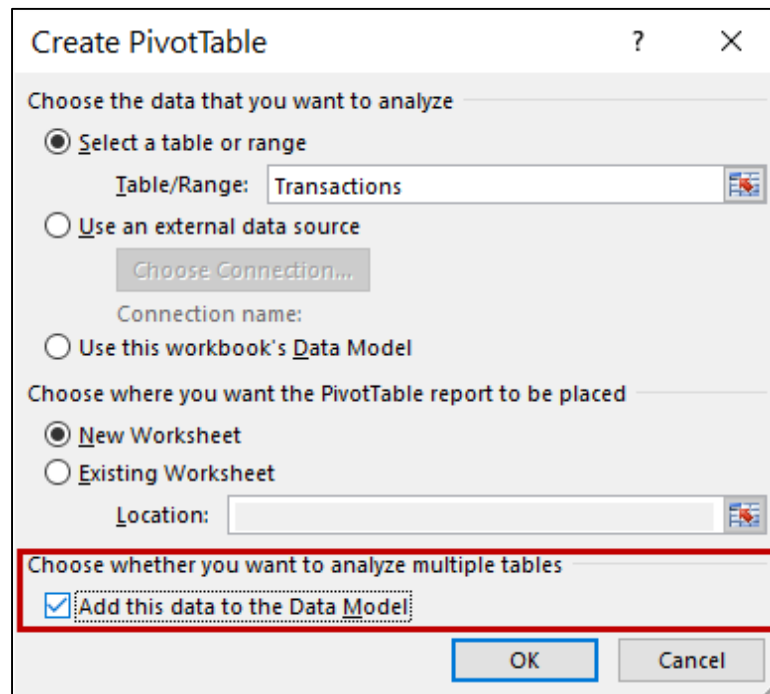


Figure 78 - Adding Tables to a Workbook's Data Model

Click **OK** to complete the process, and Excel opens a new workbook with the **PivotTable Field List** available as shown in **Figure 79**. As indicated in the figure, click **All** to see all of the tables available for use in the PivotTable. Further, notice the faint line that separates the *Product* table from the *Transactions* table. This indicates that the two tables are not yet related. You should never build a PivotTable (or any other analytical tool) from tables that are not properly related in the data model! In fact, if you do, Excel will display a warning message indicating that you may need to relate your tables. Further, if you do build a PivotTable from unrelated tables, you will probably notice that the calculations in your PivotTable do not seem to make sense. In short, always ensure that proper relationships exist before continuing with your PivotTable.

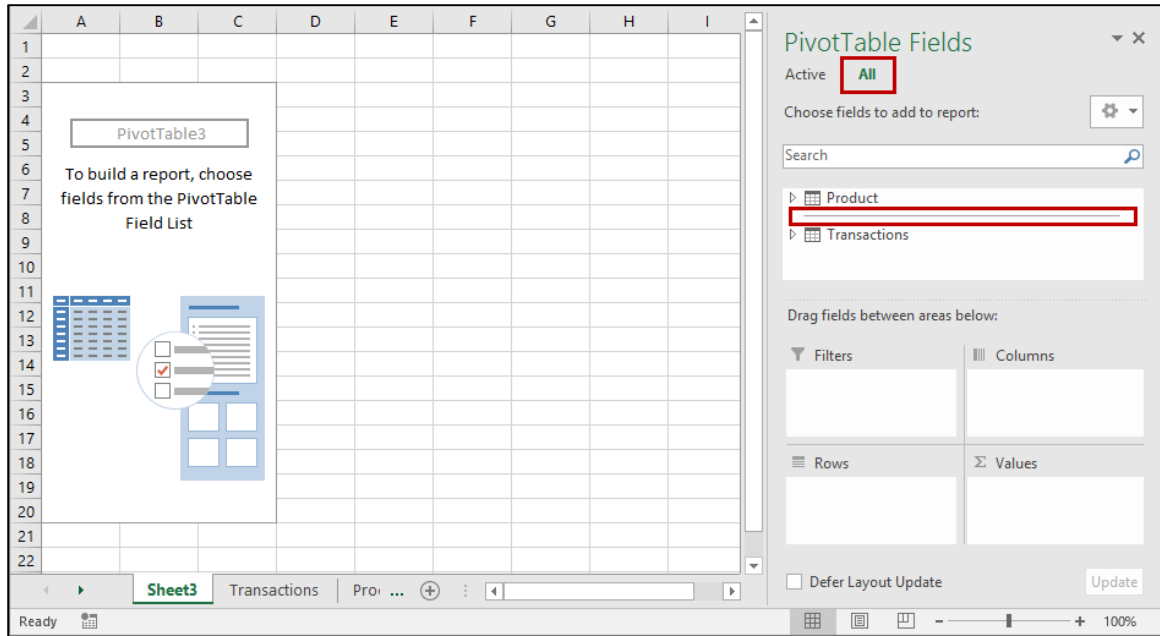


Figure 79 - Unrelated Tables in PivotTable Field List

You can establish relationships between the tables in the data model using multiple approaches. First, if Excel provides you with an **Auto-Detect** option similar to that pictured in **Figure 80**, you can click on the **Auto-Detect** button, and Excel will attempt to identify and relate the tables together automatically. In relatively simple data models, there is a high probability that this process will be successful. However, if it is not, you will need to establish the relationships manually. To do this, you can click the **Create** button pictured in Figure 80, or you can click **Relationships** from the **PivotTable Tools Analyze** tab of the Ribbon. Either of these approaches will open the **Manage Relationships** dialog box pictured in **Figure 81**. You can also access the Manage Relationships dialog box by clicking the **Relationships** button on the **Data** tab of the Ribbon.

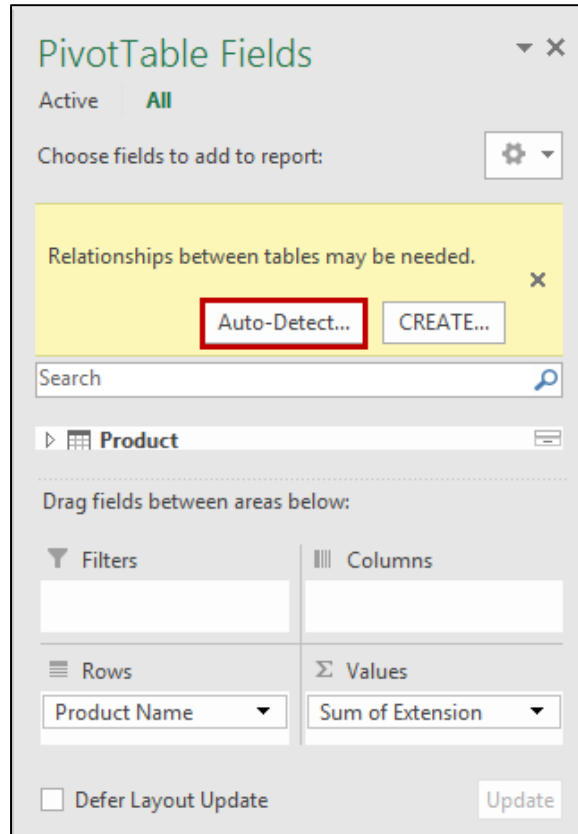


Figure 80 - Using Excel's Auto-detect Feature to Establish Relationships in Data Models



Figure 81 - Excel's Manage Relationships Dialog Box

In the Manage Relationships dialog box, click the **New** button to open the **Create Relationship** dialog box shown in **Figure 82**. On the left side of this dialog box, choose the tables you wish to relate and, on the right side of the dialog box, choose the fields that are common to the two

tables. Click **OK** to complete the process of relating these two tables and then repeat this process for as many tables as you need to relate in the data model.

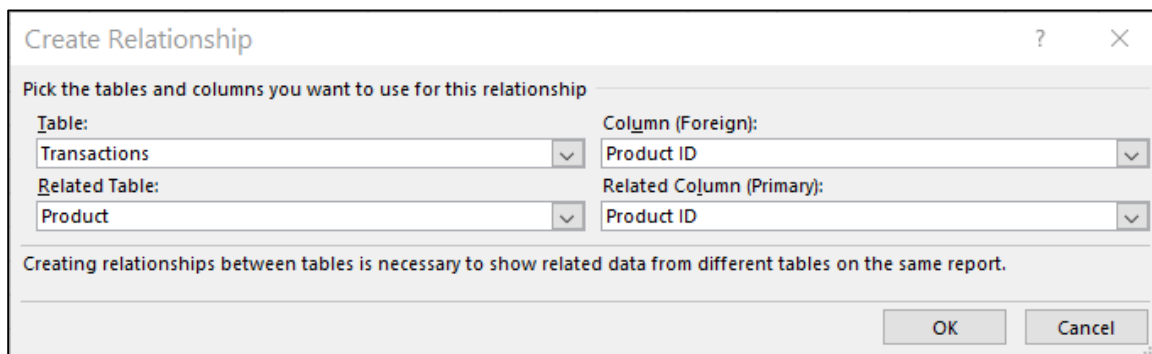


Figure 82 - Manually Establishing Relationships between Tables in a Data Model

After successfully establishing the relationship, the Manage Relationships dialog box will change to reflect the relationship and its current status as shown in **Figure 83**.

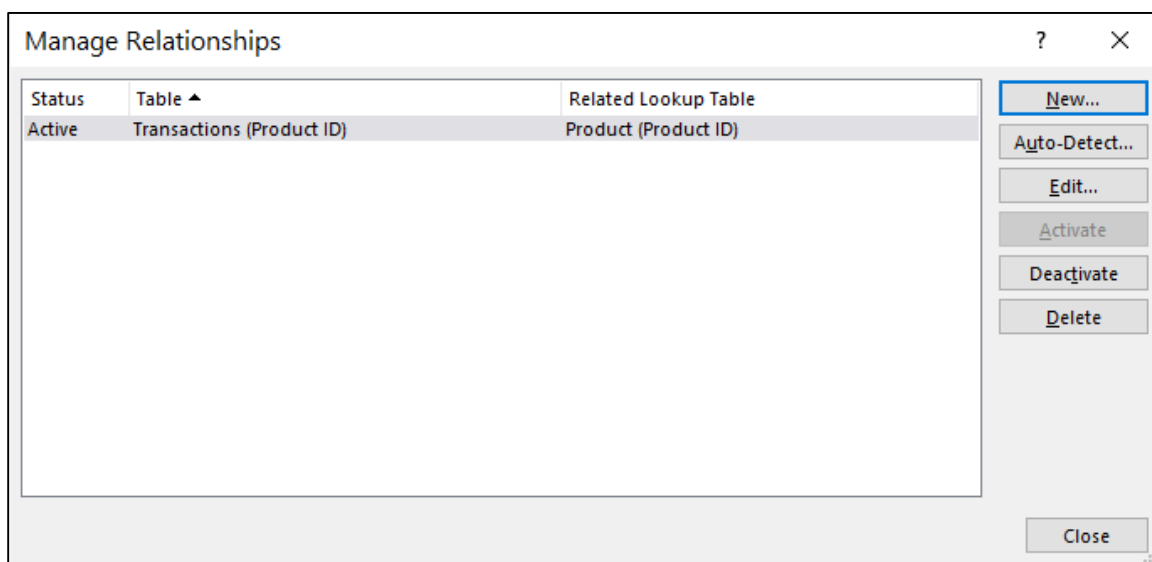


Figure 83 - Displaying All Relationships in the Manage Relationship Dialog Box

With the relationship between the two tables in place, we can easily complete the process of constructing a PivotTable that utilizes data from both of the tables in the data model. **Figure 84** presents an excerpt of that PivotTable.

	A	B	C	D	E	F
3	Summary of Sales by Product/Year					
4		2016	2017	2018	Grand Total	
5	Active Outdoors Crochet Glove	516.20	3,563.44	775.03	4,854.67	
6	Active Outdoors Lycra Glove	674.85	4,922.01	1,560.09	7,156.95	
7	Descent	77,759.06	1,312,790.24	495,070.87	1,885,620.17	
8	Endorphin	12,192.99	204,626.02	60,020.01	276,839.02	
9	Guardian "U" Lock	35.00	763.90	105.00	903.90	
10	Guardian ATB Lock	39.42	782.93	306.60	1,128.95	
11	Guardian Chain Lock	27.00	250.68	39.38	317.06	
12	Guardian Mini Lock	325.23	825.63	434.73	1,585.59	
13	Guardian XL "U" Lock	19.90	1,104.46	179.10	1,303.46	
14	InFlux Crochet Glove	54.00	1,732.13	693.23	2,479.36	
15	InFlux Lycra Glove	369.69	1,948.37	593.67	2,911.73	
16	Micro Nicros	4,307.31	24,568.02	8,436.25	37,311.58	
17	Mini Nicros	5,524.28	27,128.10	6,426.19	39,078.57	
18	Mozzie	57,154.08	452,709.11	122,659.45	632,522.64	
19	Nicros	13,556.88	147,789.54	42,435.27	203,781.69	
20	Rapel	18,450.26	161,853.64	52,759.56	233,063.46	

Figure 84 - Sample PivotTable Created from a Data Model

Data Model Relationships

When working with data models, you should understand a few very important points regarding relationships between tables. First, the names of the columns that you relate together need not be the same. For example, in the previous illustration one of the column names could have been *Product ID*, and the other column name could have been *Part Number*. What is important is that the columns have the same type of data or compatible types of data in them as discussed below.

Second, Excel data models support only two types of relationships: 1) one-to-one relationships and 2) one-to-many relationships. An example of a one-to-one relationship exists when you have an employee's name and address in one table and a single corresponding record for that same employee's tax withholding status in another table. An example of a one-to-many relationship exists when you have a single product in one table and multiple transactions associated with that same product in a separate table, as is the case with the simple relationship created in Figure 78 through Figure 83. Excel does not support many-to-many relationships in data models. An example of a many-to-many relationship might exist when attempting to relate a product table to a customer table because each customer can buy many products, and the same product can be bought by many customers.

Third, three types of *keys* are particularly important to Excel data models.

1. **Primary key:** uniquely identifies a row in a table, such as Product ID in the Product table.
2. **Alternate key:** a column other than the primary key that is unique. For example, the Product table might also store a SKU code, which is unique.
3. **Foreign key:** a column that refers to a unique column in another table, such as Product ID in the Transactions table, which refers to Product ID in the Product table.

In a Data Model, the primary key or alternate key is referred to as the *related column*. If a table has both a primary and an alternate key, you can use either one as the basis of a table relationship. The foreign key is referred to as the source column or just *column*. In the example above, a relationship is defined between Product ID in the Transactions table (contains the *column*) and Product ID in the Product table (contains the *related column*). When creating a one-to-many relationship, the “one” side of the relationship will always be the *related column*.

Fourth, to create relationships between tables, three requirements must be met.

1. Each table must have a single column of data that uniquely identifies each row in that table. As indicated above, this column of data is that table's primary key or an alternative key.
2. The data values in the lookup column must be unique – that is to say, they cannot contain duplicate values, including null values.
3. The data types in the source column and lookup column must be compatible; whole numbers, decimal numbers, true/false, text, dates, currency, and blanks are all compatible data types for the purposes of establishing relationships.

Simple Relationships in Data Models

After you create a data model, you may find it necessary to add, delete, activate, or deactivate relationships in the data model. To do so, return to the **Manage Relationships** dialog box shown in **Figure 85**, choose the relationship of interest, and click the appropriate button on the right side of the dialog box.

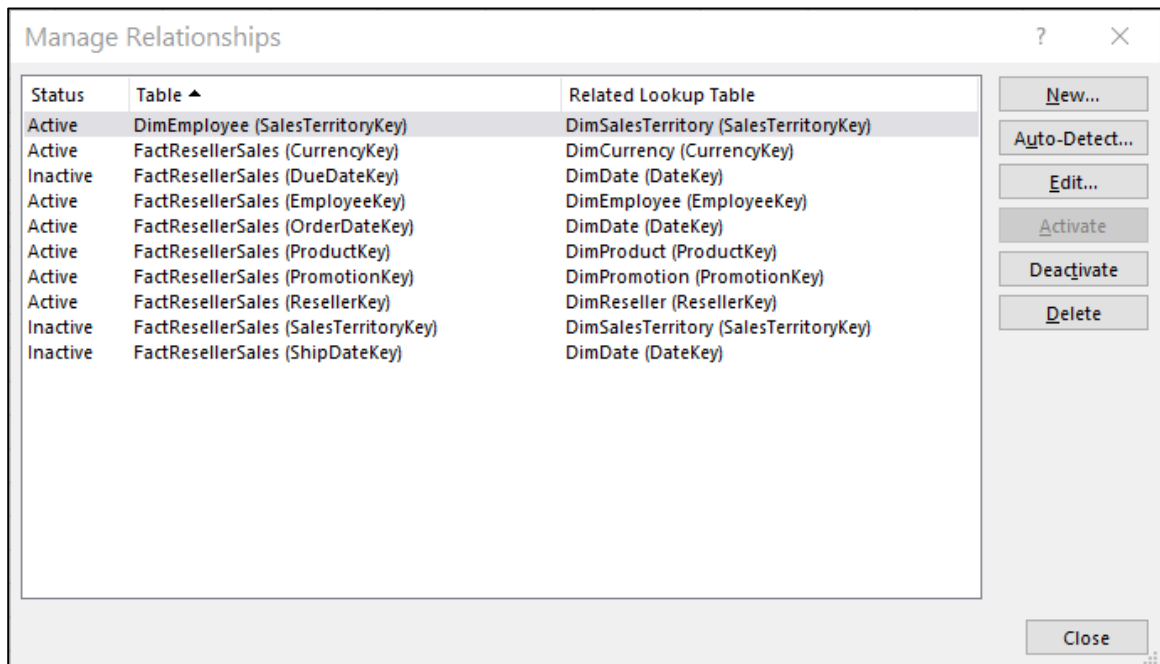


Figure 85 - Managing Relationships in a Data Model

As shown in Figure 85, a single table can have multiple relationships to other tables in the same data model. For example, the FactResellerSales table has three relationships to the DimDate table – the DueDateKey, the OrderDateKey, and the ShipDateKey fields in the FactResellerSales table all relate to the DateKey field in the DimDate table. This will often be the case when you import tables that already have multiple relationships defined in the original data source. In cases such as this, only one of the relationships can serve as the active, current relationship; Excel automatically marks the others as inactive. In Figure 85, the relationship between the OrderDateKey in the FactResellerSales table and the DateKey field in the DimDate table is the active relationship between those two tables, and the other two relationships have been marked as inactive. If you wanted the relationship between the two tables to be based on the DueDateKey field in the FactResellerSales table, you would need to select the existing active relationship and then click the **Deactivate** button in the **Manage Relationships** dialog box. After doing so, select the inactive relationship between the DueDateKey field and the DateKey field and click **Activate**. Be aware that changes such as these can cause calculations that are based on the data in the data model to change, so it is a good idea to scrutinize the results of changes such as this carefully to ensure the continued accuracy of your reports.

A Summary Example of Simple Data Models

Many businesses report on fiscal years, not on calendar years. However, the built-in grouping options in PivotTables do not support groupings based on fiscal reporting periods, leading to untold frustration and time lost trying to implement manual workarounds. Fortunately, this is an ideal situation in which we can use data models to facilitate reporting on fiscal years. In this example, we will work with data in two tables – *Sales* and *Dates* – and create a data model from which we will construct a PivotTable that groups information on fiscal reporting periods for a company that reports on 4-4-5 reporting periods.

Figure 86 illustrates a selection of the 18,271 rows of transactional data in the Sales table on which we need to report by fiscal period.

	A	B	C	D	E
1	Date	Location	Sales		
14602	1/1/2016	Baton Rouge Store No 1	6,905.52		
14603	1/1/2016	Baton Rouge Store No 2	6,500.58		
14604	1/1/2016	Jackson Store	5,932.37		
14605	1/1/2016	Slidell Store	5,325.62		
14606	1/1/2016	Hammond Store	5,120.09		
14607	1/1/2016	Lafayette Store	5,681.53		
14608	1/1/2016	Biloxi Store	5,174.15		
14609	1/1/2016	New Orleans Store No 1	6,578.25		
14610	1/1/2016	New Orleans Store No 2	6,821.27		
14611	1/1/2016	Shreveport Store	5,919.57		
14612	1/2/2016	Baton Rouge Store No 1	6,906.54		
14613	1/2/2016	Baton Rouge Store No 2	6,501.53		
14614	1/2/2016	Jackson Store	5,933.24		

Figure 86 - Sales Data to Summarize by Fiscal Year

In **Figure 87**, notice how the data in the Dates table “maps” the data in column A into the appropriate **Fiscal Week**, **Fiscal Month**, **Fiscal Quarter**, and **Fiscal Year**. Instead of grouping the PivotTable based the date in column A of each transaction, we will use the data in columns B through E for grouping purposes.

	A	B	C	D	E
1	Date	FWeek	FMonth	FQuarter	FYear
1455	12/25/2015	W52	Dec	Q4	2015
1456	12/26/2015	W52	Dec	Q4	2015
1457	12/27/2015	W52	Dec	Q4	2015
1458	12/28/2015	W01	Jan	Q1	2016
1459	12/29/2015	W01	Jan	Q1	2016
1460	12/30/2015	W01	Jan	Q1	2016
1461	12/31/2015	W01	Jan	Q1	2016
1462	1/1/2016	W01	Jan	Q1	2016
1463	1/2/2016	W01	Jan	Q1	2016
1464	1/3/2016	W01	Jan	Q1	2016
1465	1/4/2016	W02	Jan	Q1	2016
1466	1/5/2016	W02	Jan	Q1	2016
1467	1/6/2016	W02	Jan	Q1	2016

Figure 87 - Excerpted Data from Dates Table

Begin the PivotTable creation process by selecting **Insert, PivotTable** to open the **Create PivotTable** dialog box. Enter the name of the first table and make sure to check **Add this data to the Data Model** as shown in **Figure 88**. Click **OK** to display the PivotTable placeholder and Field List.

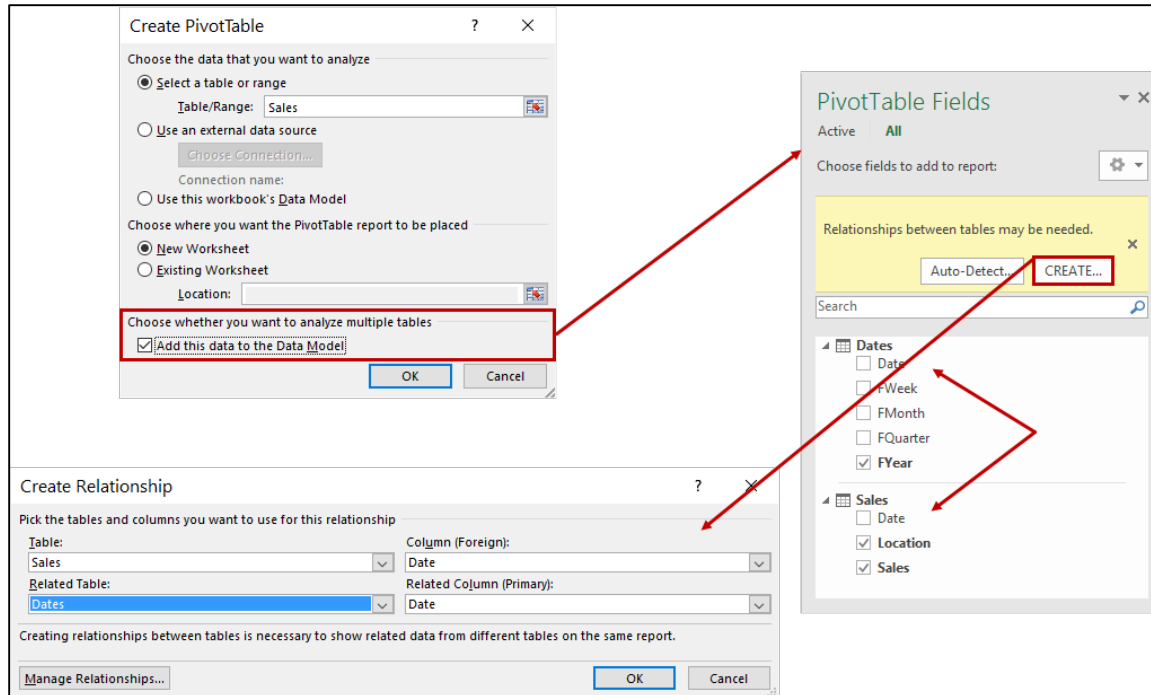


Figure 88 – Creating a PivotTable Using the Excel Data Model

When a table is added to the Data Model, the PivotTable Field List can display *all* fields from *all* tables in a workbook. If the tables are not related, you can create relationships based on a common field directly from the Field List as shown in Figure 88.

After establishing one or more relationships, you can construct the PivotTable which will summarize the data based on the appropriate fiscal reporting period. To do so, simply drag the FWeek, FMonth, FQuarter, and FYear fields to the PivotTable. **Figure 89** provides a sample of the completed PivotTable. In particular, notice the omission of the Date field; instead, see how the FWeek, FMonth, FQuarter, and FYear fields are used to summarize the data in the PivotTable by fiscal reporting period.

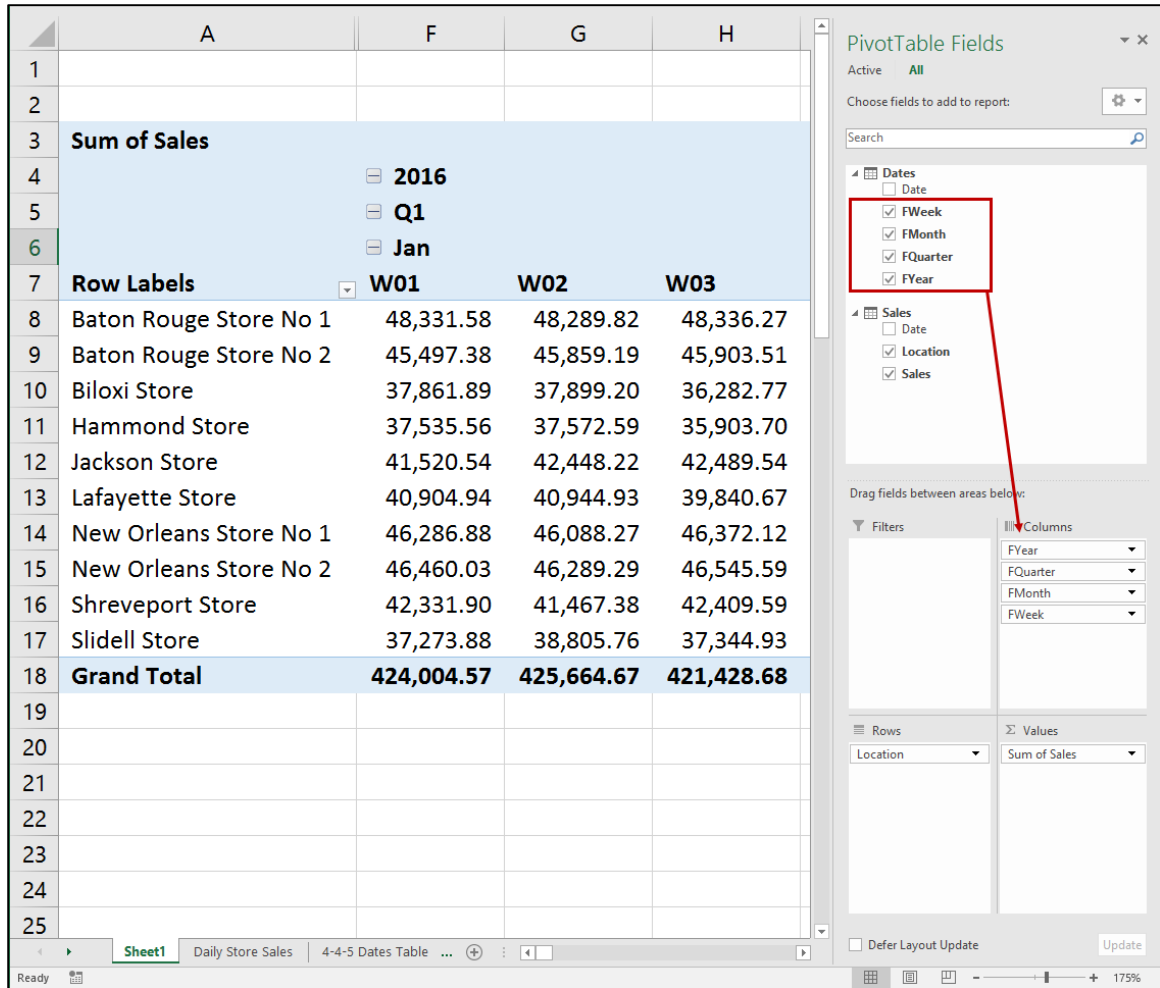


Figure 89 - PivotTable Based on Fiscal Reporting Periods

Advanced Data Model Tools and Techniques

With a fundamental understanding of data models in place, let us now turn our attention to some of the more advanced tools and techniques associated with this feature. Included in these are the ability to create data models through importing and linking processes, building calculations in data models, and graphically managing relationships in data models.

Learning Objectives

Upon completing this chapter, you should be able to:

- Build data models by linking data from external data sources;
- Utilize data obtained through Power Query in data models;
- Define Data Analysis Expressions (DAX) and create calculations in data models using DAX;
- Add special features to data models such as Perspectives and Hierarchies; and
- Manage data model relationships using the Diagram View.

Creating Data Models from External Data Sources

You learned how to create data models by relating together multiple tables that already reside in an Excel workbook. However, in some cases, that method may not be the best option for creating data models. More specifically, if the data you want to summarize through a data model resides elsewhere – your ERP database, for example – it would probably be preferable to link that data directly into a data model. Fortunately, Excel facilitates links such as these, allowing you to create connections to external databases and populate your data model with data linked from these external databases. Through Open Database Connectivity (ODBC), you can link data into Excel from virtually any data source.

To illustrate, assume the data that you need to analyze is stored in a SQL Express database, and you would like to link it into a data model. Begin the process by clicking **From Other Sources** on the **Data** tab of the Ribbon and choosing the **From SQL Server** option to open the **Data**

Connection Wizard dialog box pictured in **Figure 90**. In the dialog box, enter the name of the server where your data resides and your authentication credentials. Click **Next** to continue.

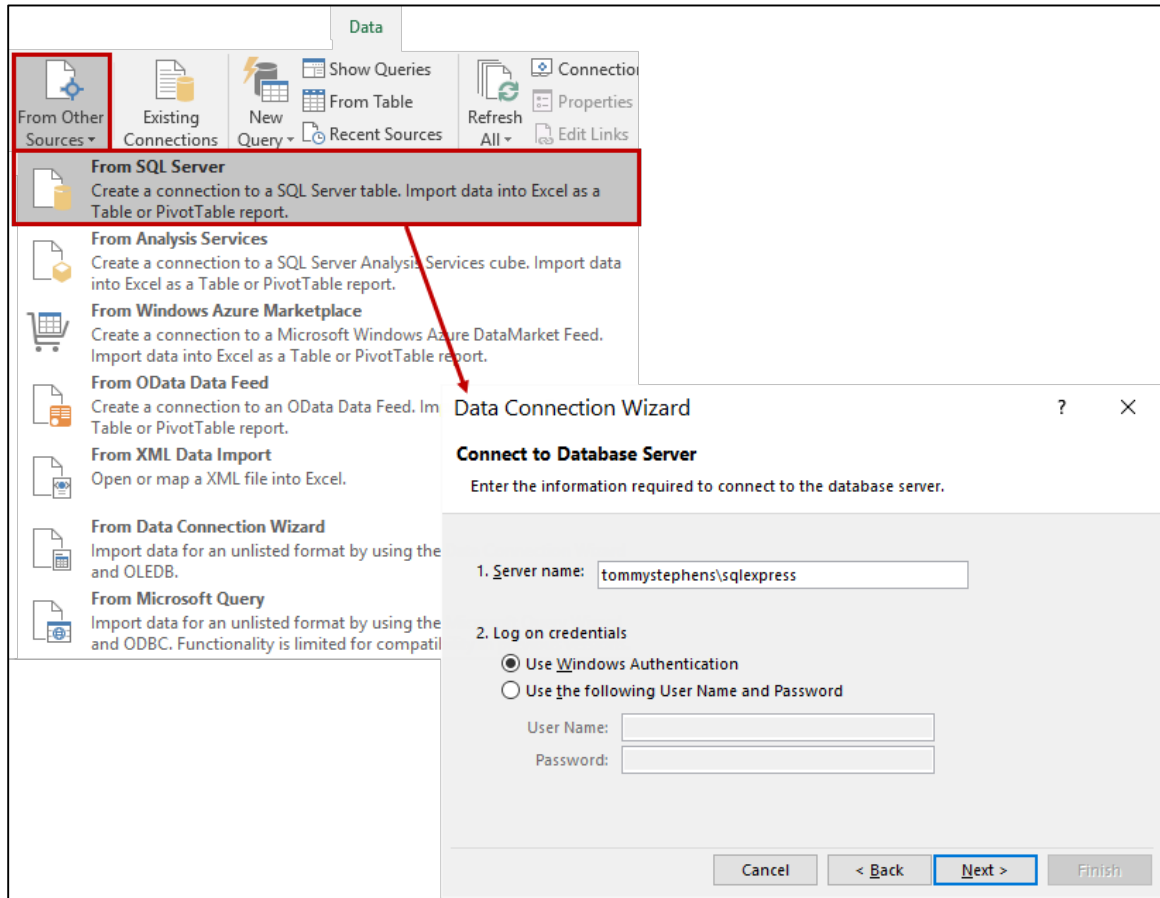


Figure 90 - Establishing a Data Connection to a SQL Server Database

There are at least five considerations to explore in the ensuing dialog box pictured in **Figure 91**.

1. Near the top of the dialog box, you will need to select the name of the database that contains the data you want to link into your data model.
2. To select multiple tables, check the box next to **Enable selection of multiple tables**. This will add a check box next to each table listed in the dialog box, allowing you to choose specifically which tables you will link into the data model.
3. Check the box next to each table you want to link into the data model.
4. Check the box next to **Import relationships between selected tables** to ensure that any relationships already established in the underlying database carry over into the data model.

5. Once you select a table to link into a data model, you can choose to have Excel link all related tables from the database into the data model. To do this, select the table of interest and then click **Select Related Tables** in the lower right corner of the dialog box.

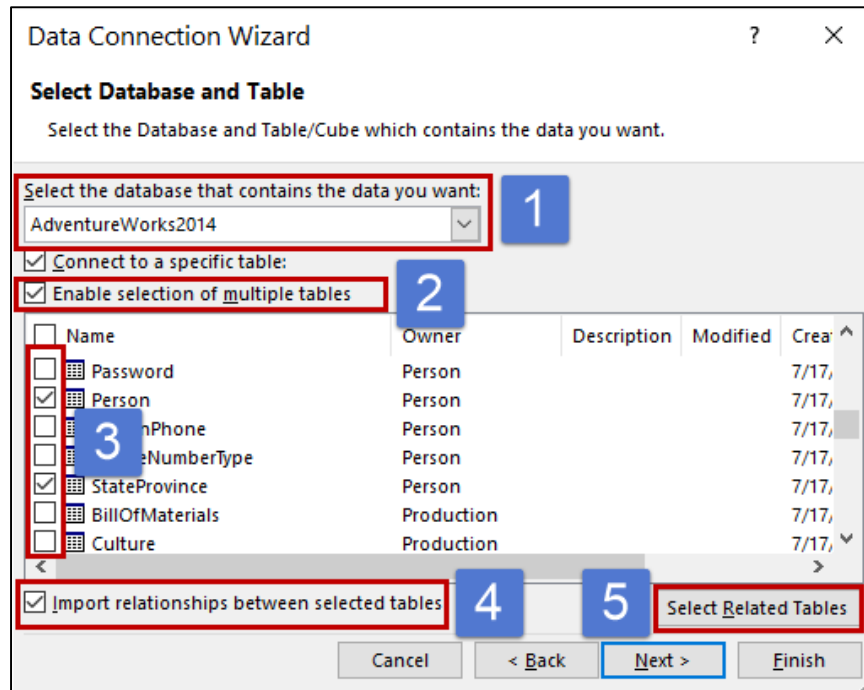


Figure 91 - Selecting Data to Link into an Excel Data Model

Upon clicking **Finish** in Figure 91, Excel opens the **Import Data** dialog box pictured in **Figure 92**. Most of the options presented in the Import Data dialog box are self-explanatory, indicating which type of object to which Excel will link the data.

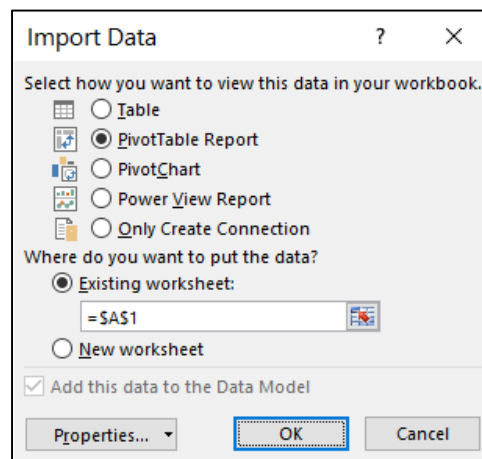


Figure 92 - Excel's Import Data Dialog Box

Three important considerations exist with respect to linking data into a data model. First, in all circumstances, Excel creates a data model from the data you are linking, even if you choose to

link data from only one table. Second, if you choose the **Only Create Connection** option, you will see the data only by viewing the data model; of course, if you choose any of the other options, you can see the data in the other object created in addition to seeing it in the data model. Third, Excel creates an **Office Data Connection (ODC)** file “behind the scenes” that saves the settings for this data connection. You can later run the same query in any workbook by simply launching the ODC file associated with the query. To do so, choose **Existing Connections** from the **Data** tab of the Ribbon, scroll through the list of ODC files, select the ODC file you wish to run, and click **Open**.

Once you link the data into the data model, you can modify some of the parameters of the query, including intervals at which the data will refresh. To do so, click **Connections** from the **Data** tab of the Ribbon to open the **Workbook Connections** dialog box. In the Workbook Connections dialog box, select the query you want to modify and click **Properties** to open the **Connection Properties** dialog box. There, change any properties necessary to tailor the query more precisely to your needs. **Figure 93** illustrates this process.

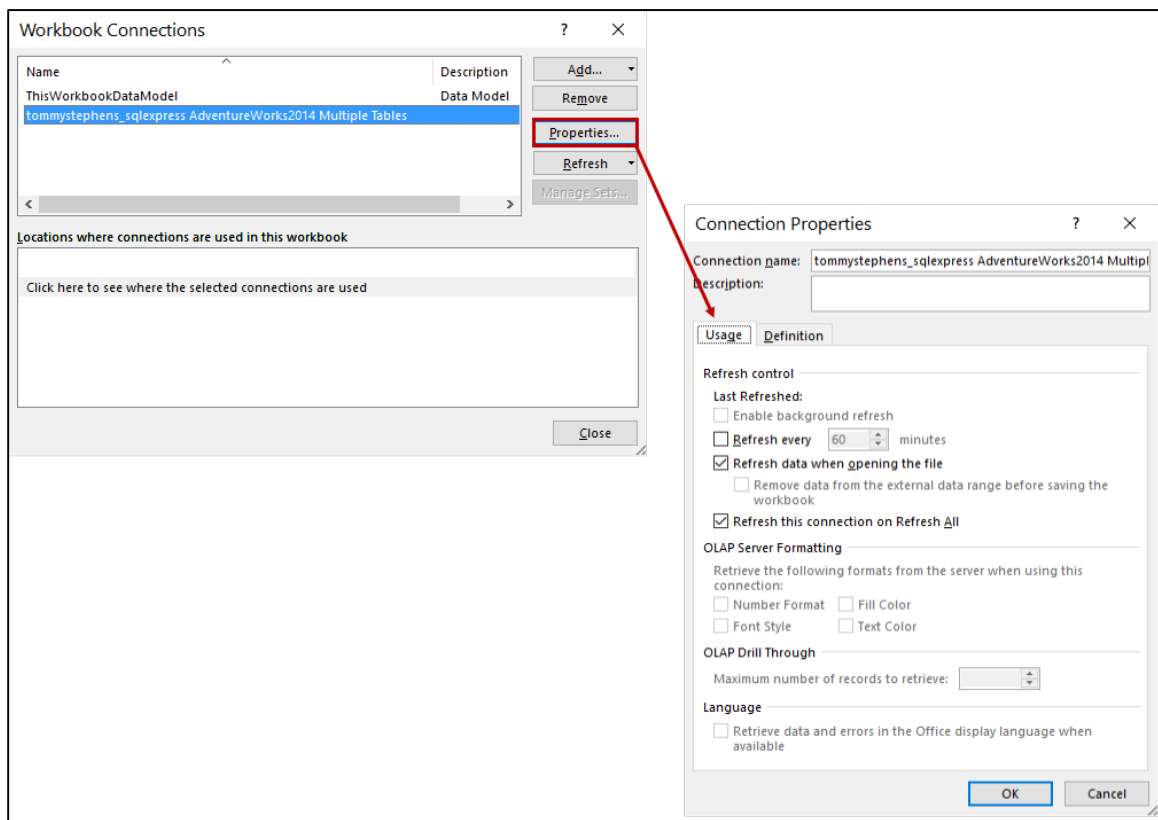


Figure 93 - Modifying the Properties of a Data Query Linking to a Data Model

It would be difficult, if not impossible, to overstate the importance of being able to link data from external data sources into a data model. With this capability, you are relieved of the tedious and time-consuming tasks associated with copying and pasting data from external data sources into Excel. This means that your data models – and by extension, all reports you build based on these data models – can summarize what effectively is real-time data with no additional effort

required of you once you build the data models and related reports. For many, this is a “game-changer” with respect to financial and operational reporting!

Building and Adding to Data Models with Power Query

As powerful as the ability to link data from external data sources into Excel data models is, many users will find the process to be somewhat challenging and perhaps even intimidating. To overcome this obstacle, you may choose to use Excel’s **Power Query** tool, available for those who run Excel 2010 or newer. Power Query is designed to make connecting to external data sources relatively easy, even for novice users. Power Query facilitates end users making connections to virtually all types of data sources and linking the data from those sources into Excel. Once you link the data into Excel, you can take virtually any action on the data, including creating data models, summarizing the data with formulas, building PivotTables from the data, and constructing charts and other visualizations.

Before you can work with Power Query, you may need to download and install it, as indicated below.

- If you are running Excel 2016, the functionality provided by Power Query is available in the **Get & Transform** group on the **Data** tab of the Ribbon. Microsoft supports Power Query in all versions of Excel 2016, regardless of whether you are running a 32-bit or a 64-bit version of Excel.
- If you are running Excel 2010 or Excel 2013 (and Power Query does not appear as a separate tab on the Ribbon), proceed to the following site to download the tool: <https://www.microsoft.com/en-us/download/details.aspx?id=39379>. Additionally, follow the installation instructions provided on the site. After installing and activating the add-in, Power Query will appear as a separate tab on the Ribbon. As with Excel 2016, Microsoft supports Power Query in all versions of Excel 2013, in both 32-bit and 64-bit versions of Excel. If you are running Excel 2010, Microsoft supports Power Query in 32-bit and 64-bit versions of Excel provisioned through Microsoft Office 2010 Professional Plus with Software Assurance.

Power Query facilitates queries from the thirty data sources indicated in **Table 2**. In particular, notice the “non-standard” data sources such as Facebook, Exchange, and Salesforce.

• Excel	• CSV	• XML	• Text	• Folders
• SQL Server	• Microsoft Office Access	• SQL Server Analysis Services	• Oracle	• IBM DB2
• MySQL	• PostgreSQL	• Sybase	• Teradata	• Microsoft Azure SQL Database
• Microsoft Azure Marketplace	• Microsoft Azure HDInsight	• Microsoft Azure Blob Storage	• Microsoft Azure Table Storage	• Web
• SharePoint List	• OData Feed	• Hadoop File	• Active Directory	• Dynamics CRM Online
• Microsoft Exchange	• Facebook	• Salesforce Objects	• Salesforce Reports	• ODBC

Table 2 - Potential Data Sources for Power Query

As shown in **Figure 94**, to launch a new query with Power Query, click **New Query** and choose your data source; in this example, the data source will be an SQL Server database that serves an organization's ERP system.

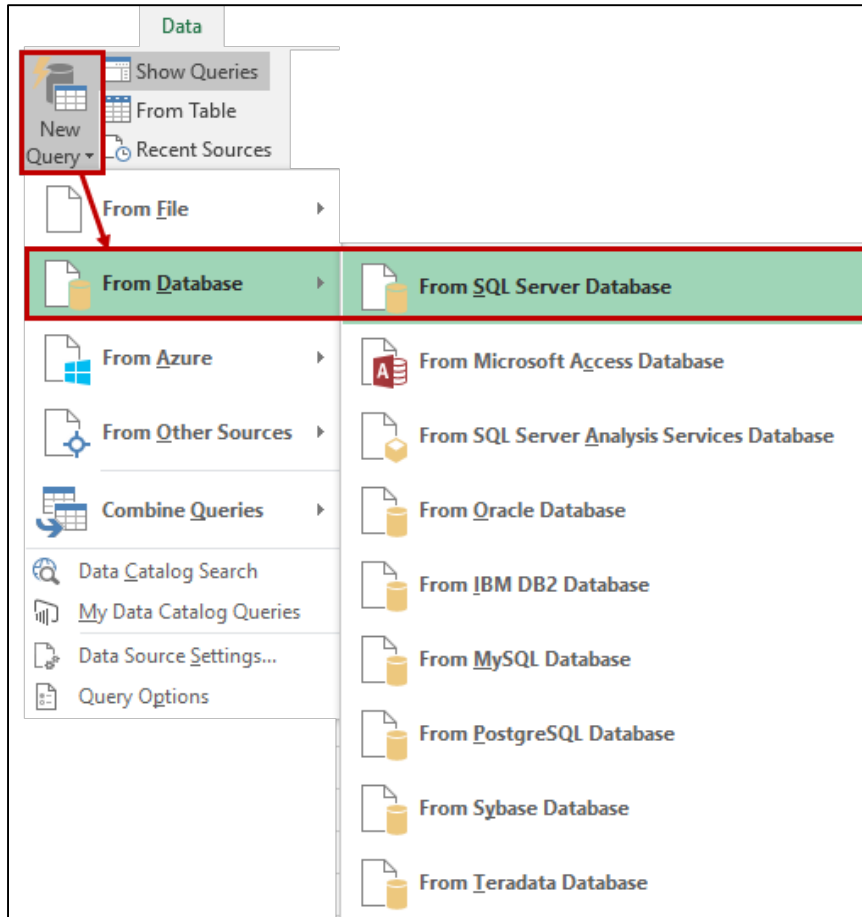


Figure 94 - Initiating a New Query Using Power Query in Excel 2016

In the ensuing dialog box pictured in **Figure 95**, enter the necessary details to identify the correct data source for the query. Click **OK** to continue.

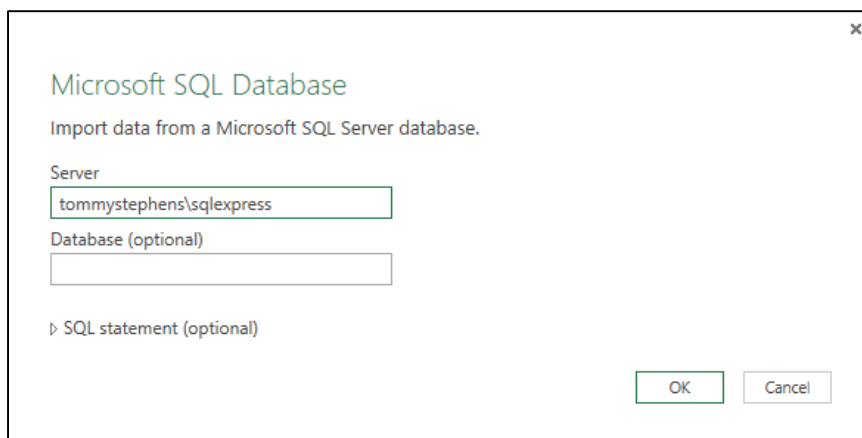


Figure 95 - Specifying the Data Source for Power Query

Next, expand the name of the database you wish to query so that you can see all of the data available for query. Then, select the items to include in your query and choose which **Load** option you wish to launch in the bottom right corner of the **Navigator** window as shown in **Figure 96**. When choosing Load options, you can elect to add the data to a table(s) in the workbook or to create only the data connection. Regardless of which choice you make, you can also specify whether the data should populate the workbook's data model.

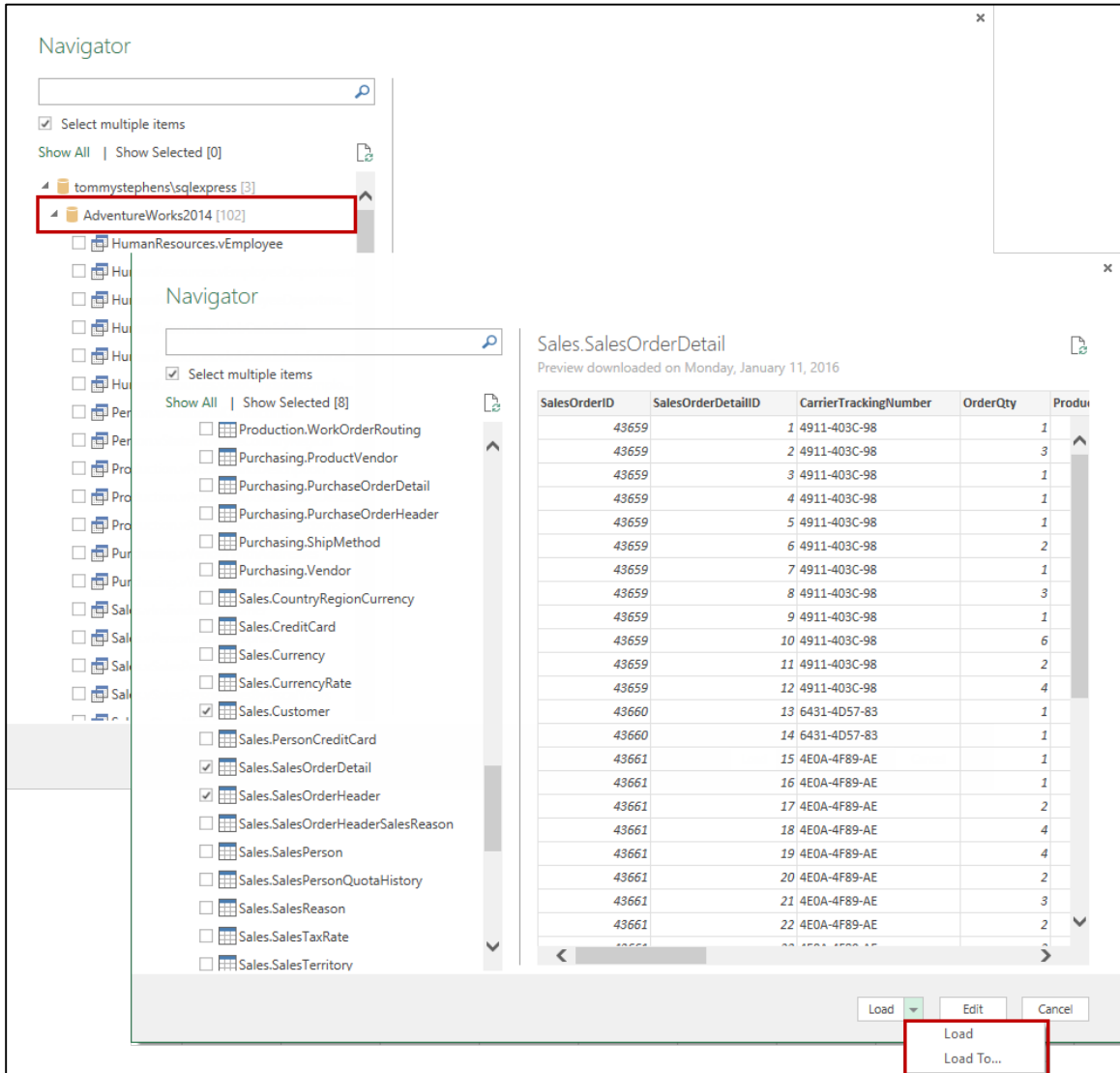


Figure 96 - Select Object for Query in Power Query

In the example at hand, eight tables were selected for query, and the election was made to load the data to visible worksheets in the workbook and also to the workbook's data model. **Figure 97** presents a snapshot of the query's results.

SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID
120545	58936	68333 A683-4B49-88	1	958	1
120546	58937	68337 858A-41E6-B7	1	979	1
120547	58937	68338 858A-41E6-B7	1	965	1
120548	58939	68346 5A15-45C3-B0	1	961	1
120549	58939	68347 5A15-45C3-B0	1	965	1
120550	58945	68446 EE31-47E0-81	1	961	1
120551	58946	68463 F9DD-426F-9A	1	959	1
120552	58948	68486 1534-4AB5-81	1	962	1
120553	58948	68494 1534-4AB5-81	1	965	1
120554	58948	68498 1534-4AB5-81	1	958	1
120555	58948	68503 1534-4AB5-81	1	978	1
120556	58948	68505 1534-4AB5-81	1	961	1
120557	58958	68672 BC3F-443F-B5	1	963	1
120558	58961	68780 0848-4E9B-81	1	960	1

Figure 97 - Results of Power Query of Eight Tables

If necessary, you can edit a particular query once you create it by right-clicking on the query in the **Workbook Queries** pane and choosing the **Edit** option as shown in **Figure 98**.

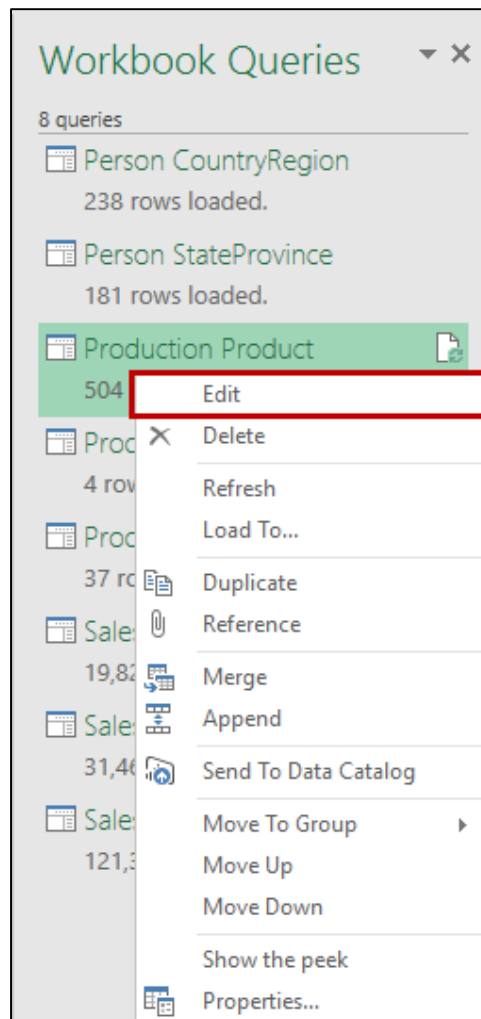


Figure 98 - Choosing to Edit a Power Query

In addition to the Edit option highlighted in Figure 98, another interesting choice is the **Send To Data Catalog** option. Clicking this option allows you to save the query so that you can reuse it in other workbooks. If you choose to save a query to your data catalog and later want to reuse that query, click **New Query**, choose **My Data Catalog Queries** to open the **Shared Queries Pane**, and then select the query you want to add to the current workbook from that pane. **Figure 99** illustrates the process of reusing a query saved in a data catalog.

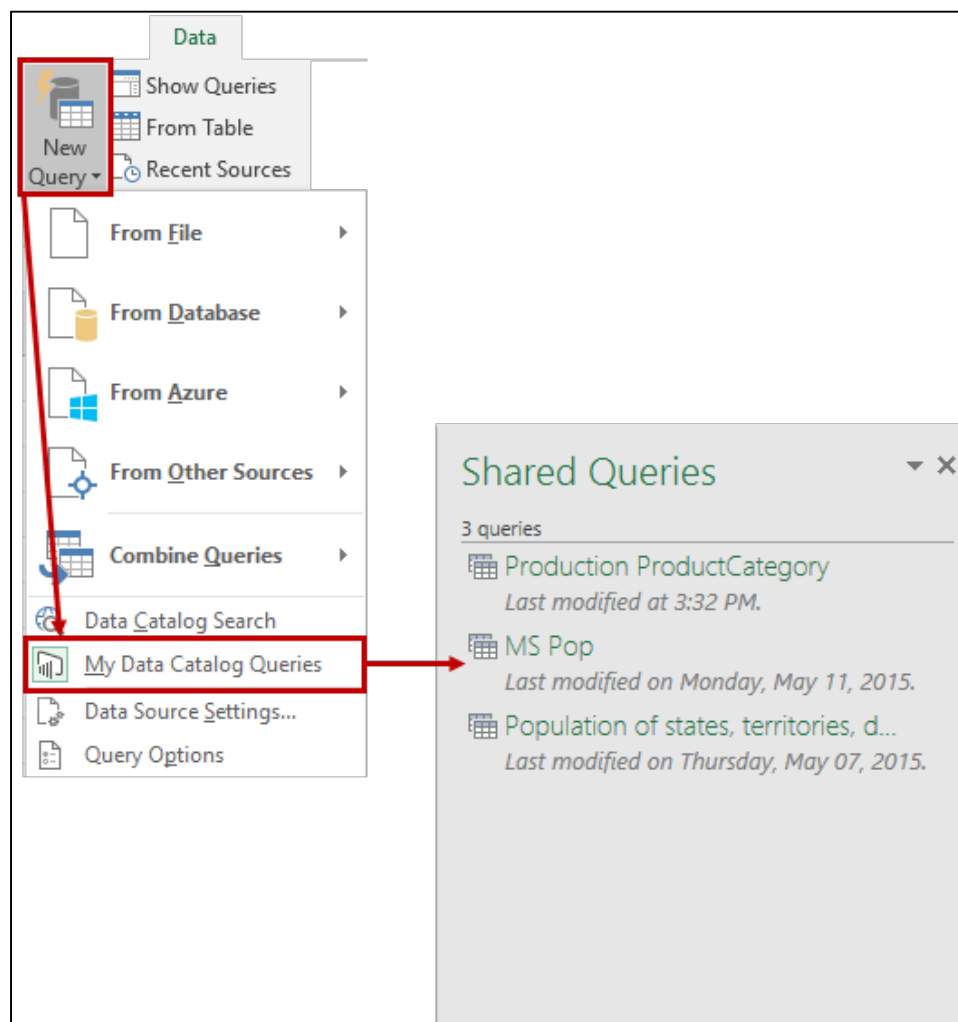


Figure 99 - Reusing a Query Stored in a Data Catalog

If you choose to load the data to the workbook's data model, you can access and manage the data model by clicking **Data Model** on the Data tab of the Ribbon or by accessing **Power Pivot**. Further, you can summarize the data in the data model with Power Pivot or create visualizations of the data with **Power View** or **Power Map** (also known as **3-D Maps**).

Using Power Pivot to Query Data into a Data Model

Notwithstanding the query tools available in Excel or Power Query, you can query data directly into a data model from within Power Pivot. As indicated on the **Home** tab of Power Pivot's

Ribbon and shown in **Figure 100**, the query capabilities are virtually the same as if you were to execute the query from within Excel or Power Query.

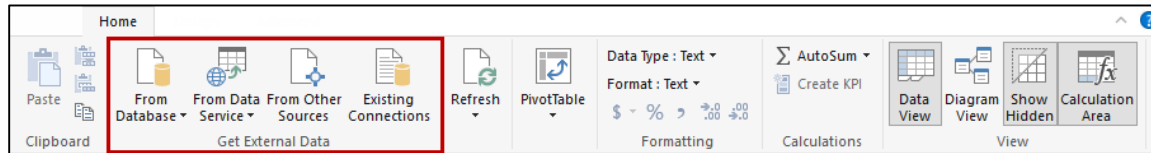


Figure 100 - Query Tools Available in Power Pivot

Table 3 indicates all of the data sources available from each Ribbon option in Power Pivot.

Ribbon Option	Available Data Sources
From Database	SQL Server, Access, Analysis Services, Power Pivot
From Data Service	Microsoft Azure Marketplace, Suggest Related Data, From OData Data Feed
From Other Sources	SQL Server, SQL Azure, SQL Server Parallel Data Warehouse, Access, Oracle, Teradata, Sybase, Informix, IBM DB2, OLEDB/ODBC, Analysis Services, Report, Azure Marketplace, Suggest Related Data, Other Feeds, Excel File, Text File

Table 3 - Available Data Sources for Each Ribbon Option in Power Pivot

Working with Data in a Data Model

Once you load your data into a data model, you have multiple options for how you view it, manage it, and even alter it. Of all of the options available, perhaps the following four are the most critical:

1. Managing data and relationships in a data model;
2. Building formulas with Data Analysis Expressions;
3. Creating hierarchies to manipulate reports more quickly; and
4. Using perspectives to control the amount of data available.

You will learn about each of these in the following sections.

Managing Data and Relationships in a Data Model

You learned how to create and manage relatively simple relationships in a data model. In some cases, you may need a more thorough set of tools you can use to manage these relationships. Fortunately, these tools are available and relatively easy to use.

To access tools for managing relationships in your data model, click **Manage Data Model** on the **Data** tab of the Ribbon or **Manage** on the **Power Pivot** tab of the Ribbon to open the data model in Power Pivot. Then, click **Diagram View** on the **Home** tab of the Ribbon to open the visual perspective of the data model pictured in **Figure 101**.

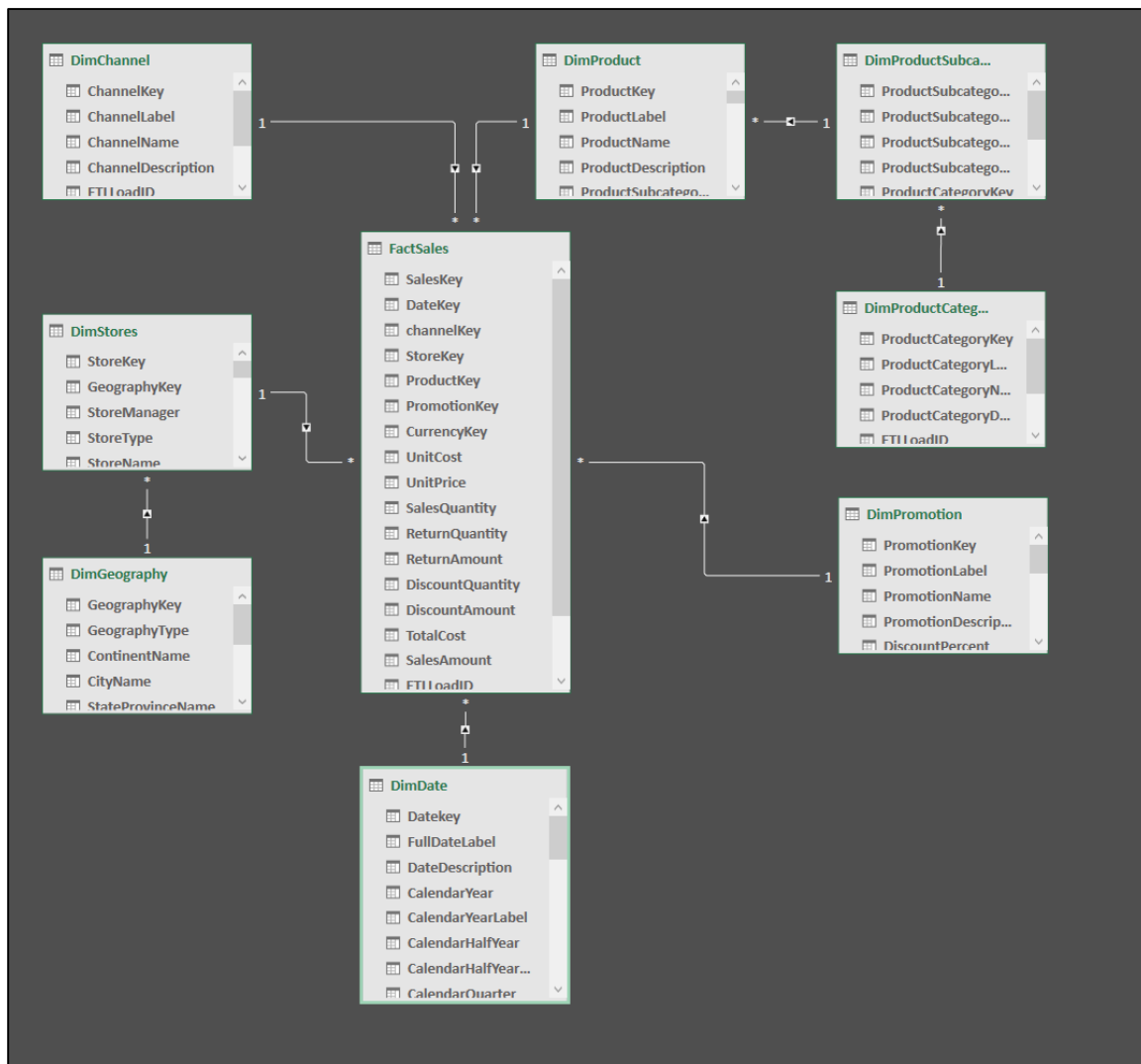


Figure 101 - Diagram View of a Data Model

Many users will find Diagram View to be a useful tool when working with data models because the visualization clearly depicts the established relationships in the data model. Regarding relationships, you will notice that many databases contain one or more “Fact” tables and multiple “Dim” tables. *Fact* tables store the data that is to be summarized – transactional data, for instance – and *Dim* tables store the various dimensions by which you may choose to summarize the data in the Fact table – product category, regions, and salesperson for instance. Because of this, typically you should expect to see “one-to-many” relationships when examining your data models, with Dim tables on the “one” side of the relationship and Fact tables on the “many” side of the relationship – many sales transactions attributable to one salesperson, for example.

In Diagram View, you can make edits to the relationships in place. For example, you can right-click on a relationship and choose to mark it as inactive, edit it, or delete it as shown in **Figure 102**.

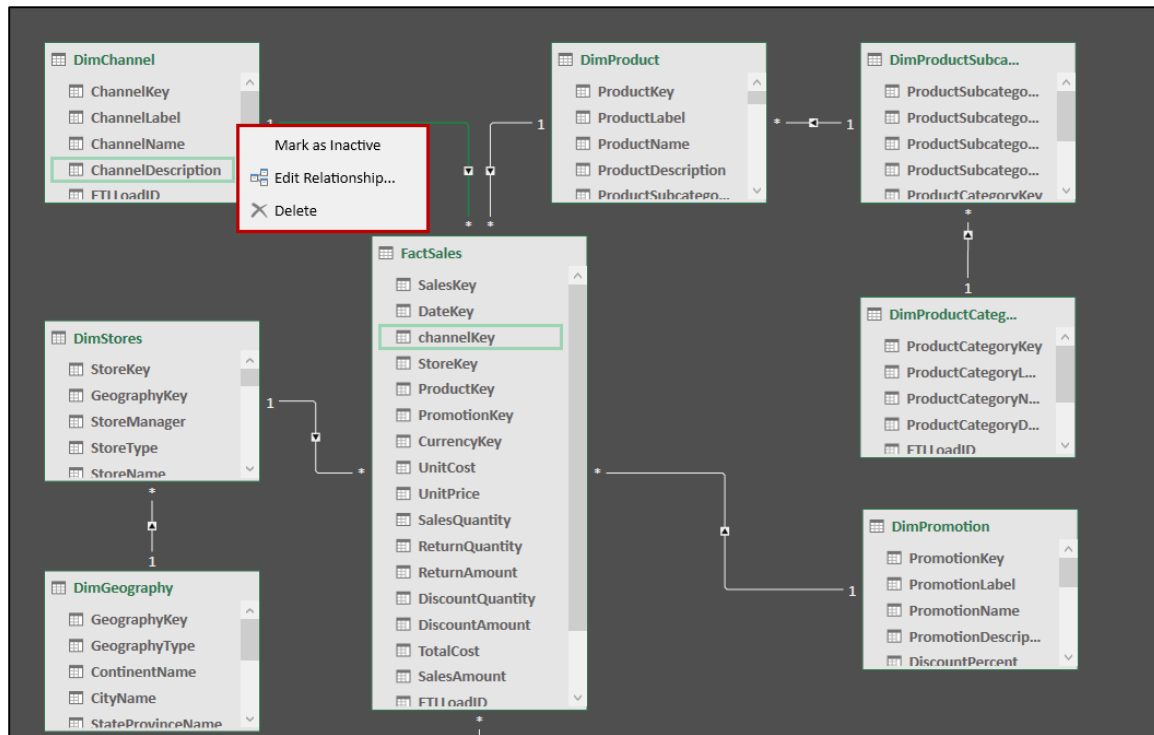


Figure 102 - Managing Relationships from Diagram View

Also, you can right-click on a field in an unrelated table and choose the **Create Relationship** option. This will open the **Create Relationship** dialog box pictured in **Figure 103**. In the lower portion of this dialog box, choose the table to which you want to establish a relationship and the field on which the relationship is to be based. Click **OK** to complete the process of relating the tables, and, upon doing so, Excel creates the relationship and automatically determine whether it is a one-to-one or one-to-many relationship. Click **Data View** on the **Home** tab of the Ribbon to close Diagram View.

Create Relationship?×

Select tables and columns that relate to one another.

DimChannel

ChannelDescription	ChannelKey	ChannelLabel	ChannelName	ETLLoadID	LoadDate	UpdateDate
Store	1	01	Store	1	7/11/2015 2:43:19 PM	7/11/2015 2:43:19 PM
Online	2	02	Online	1	7/11/2015 2:43:19 PM	7/11/2015 2:43:19 PM
Catalog	3	03	Catalog	1	7/11/2015 2:43:19 PM	7/11/2015 2:43:19 PM
Reseller	4	04	Reseller	1	7/11/2015 2:43:19 PM	7/11/2015 2:43:19 PM

FactSales

channelKey	CurrencyKey	DateKey	DiscountAmount	DiscountQuantity	ETLLoadID	LoadDate	ProductKey	PromotionKey
1	1	5/7/2016 12:00:00 AM	0	0	1	1/1/2010 12:00:00 AM	375	1
1	1	5/22/2018 12:00:00 AM	0	0	1	1/1/2010 12:00:00 AM	375	1
1	1	9/30/2018 12:00:00 AM	0	0	1	1/1/2010 12:00:00 AM	375	1
1	1	10/31/2017 12:00:00 AM	0	0	1	1/1/2010 12:00:00 AM	375	1
1	1	6/26/2018 12:00:00 AM	0	0	1	1/1/2010 12:00:00 AM	375	1

☒ Active

OKCancel

Figure 103 - Creating Relationships through the Create Relationship Dialog Box

While working in Data View, you can also access tools to manage relationships. To do so, click either **Create Relationship** or **Manage Relationships** on the **Design** tab of the Ribbon. If you choose Create Relationship, Excel presents the same dialog box illustrated in Figure 103. If you choose Manage Relationships, the dialog box Excel presents is slightly different as shown in Figure 104.

you to insert calculations directly into your Power Pivot-based PivotTables. These functions are quite powerful and truly extend the usefulness of Power Pivot for all users.

While many of the functions used for creating calculated columns work the same in Power Pivot as they do in traditional Excel applications, some differences exist, including the following.

- DAX renames Excel's **TEXT** function to **FORMAT**, **SUMIFS** to **CALCULATE**, and **CHOOSE** to **SWITCH**.
- DAX adds a new **BLANK** function used most often in tandem with IF to exclude certain rows from calculations.
- DAX uses **RELATED** instead of **VLOOKUP** to perform queries on data in a PivotTable.

In addition, according to Microsoft, some general guidelines regarding using DAX functions include those listed below.

- Many DAX functions have the same name and the same general behavior as Excel functions but have been modified to take different types of inputs and, in some cases, might return a different data type. Generally, you cannot use DAX formulas in an Excel workbook or use Excel formulas in a Power Pivot workbook without some modification.
- DAX functions never take a range of cells or a range as reference; instead, DAX functions take a column or table as reference.
- DAX date and time functions return a *datetime* data type. In contrast, Excel date and time functions return an integer that represents a date as a serial number.
- Many of the new DAX functions either return a table of values or make calculations based on a table of values as input. In contrast, Excel has no functions that return a table, but some functions can work with arrays. The ability to reference complete tables and columns easily is a new feature in Power Pivot.
- DAX provides new look up functions that are similar to the array and vector look up functions in Excel. However, the DAX functions require that a relationship be established between the tables.
- DAX does not support the variant data type found in Excel. The data in a column is always expected to be of the same data type. If the data is not the same type, DAX changes the entire column to the data type that best accommodates all values.

Creating Calculated Columns with DAX

Calculated columns are columns that you add to an existing data model. Instead of pasting or importing values in the column, you create a DAX formula that defines the values in the column. Once you create a calculated column, you can use the formula results in a calculated column the same way you would use data from any other column.

Formulas that you enter to create calculated columns are much like the formulas that you enter into Excel. However, unlike in Excel, you cannot create a different formula for different rows in a table; instead, the DAX formula is automatically applied to all rows in the entire column.

You can create calculated columns that are based on measures and other calculated columns. However, you should avoid using the same name for a calculated column and a measure because this potentially would lead to confusing results.

To illustrate the power of calculated columns built using DAX expressions, consider the extraction of data illustrated in **Figure 105**; the extraction shows 10 records of a population of 34,169 records of time and billing data for a consulting firm. The firm operates on a fiscal year starting on July 1 and ending on June 30. As discussed beginning on page 77, PivotTables – including those built from Power Pivot and from data models – have the inherent capability to group date data automatically only on calendar years/quarters/months; fiscal year groupings can be accomplished but involve manual workarounds.

Trans ...	Region	Account Category	Service Line	Sector	Hours	Net Revenue	Add Column
1	7/16/2016	Asia	Strategic Account	IT Consulting	Aerospace & Defense	90.61	8509.39
2	7/18/2016	Asia	Strategic Account	Training	Aerospace & Defense	13.58	2490.63
3	7/24/2016	Asia	Regional	IT Consulting	Aerospace & Defense	0.81	1188
4	8/10/2016	Asia	Regional	IT Consulting	Aerospace & Defense	3.6	1242.78
5	8/19/2016	Asia	Strategic Account	IT Consulting	Aerospace & Defense	17.9	7380.15
6	9/1/2016	Asia	Regional	HR Management	Aerospace & Defense	502.03	94125.17
7	9/3/2016	Asia	Regional	HR Management	Aerospace & Defense	26.85	5177.18
8	9/5/2016	Asia	Global Account	Advisory Services	Aerospace & Defense	473.4	107177.27
9	9/7/2016	Asia	Strategic Account	IT Consulting	Aerospace & Defense	310.91	100276.66
10	9/22/2016	Asia	Strategic Account	IT Consulting	Aerospace & Defense	13.07	3211.15

Data

Record: 1 of 34,169

Figure 105 - Extraction of Time and Billing Data to be Used in Creating Calculated Columns with DAX

In this example, a relatively easy-to-implement workaround will be to use three DAX functions – **MONTH**, **YEAR**, and **IF** – to create calculated columns in the data model to identify the appropriate fiscal quarter and fiscal year for the time and billing records. These three functions operate almost exactly as DAX functions as they do in a traditional Excel environment.

To begin, double-click at the top of a blank column in the data model and enter *Month* as the column header. Repeat that process two times, adding *Fiscal Quarter* and *Fiscal Year*. On the first row of the *Month* column, enter the formula shown below.

=Month([Trans Date])

This formula simply identifies the month number (1 through 12) based on the data in the current row of the *Trans Date* column in the data model.

Next, enter the following formula in the first row of the *Fiscal Quarter* column.

=IF([Month]<4,3,IF([Month]<7,4,IF([Month]<10,1,2)))

This formula checks to see if the month is January, February, or March (month numbers 1 through 3) and, if so, it assigns the value of 3 to that row of the *Quarters* column, indicating that the month falls into the third fiscal quarter. If the month is April, May, or June (month numbers 4 through 6), the formula assigns the value of 4 to that row of the *Quarters* column, indicating that the month falls into the fourth fiscal quarter. If the month is July, August, or September (month numbers 7 through 9), the formula assigns a value of 1 to that row of the *Quarters* column, indicating that the month falls into the first fiscal quarter. Finally, if the month is October, November, or December (month numbers 10 through 12), the formula assigns a value of 2 to that row of the *Quarters* column, indicating that the month falls into the second fiscal quarter.

Finally, enter the following formula into the *Fiscal Year* column.

=If([Month]<7,Year([Trans Date]),Year([Trans Date])+1)

This formula identifies the appropriate fiscal year for the transaction. It does so by checking to see if the month is January through June and, if it is, assigns the transaction to the year found in the *Trans Date* field. Otherwise, if the month is July through December, the transaction is deemed to be in the following fiscal year. **Figure 106** displays the modified data model, now with three newly added calculated columns based on MONTH, YEAR, and IF, three DAX functions.

Trans Date	Region	Account Category	Service Line	Sector	Hours	Net Revenue	Month	Fiscal Quarter	Fiscal Year
7/1/2016	North A...	Global Account	IT Consulting	Defense...	40.4	8847.68	7	1	2017
7/1/2016	North A...	Regional	IT Consulting	Life Scie...	66.3	14558.58	7	1	2017
7/1/2016	North A...	Strategic Account	HR Manageme...	Life Scie...	139	22589.77	7	1	2017
7/1/2016	North A...	Regional	IT Consulting	Life Scie...	197.4	36884.74	7	1	2017
7/1/2016	North A...	Global Account	HR Manageme...	Oil & Gas	3461.5	680472.43	7	1	2017
7/1/2016	North A...	Regional	IT Consulting	Pharmac...	73.5	19500	7	1	2017
7/1/2016	North A...	Regional	IT Consulting	Pharmac...	30.7	8623.04	7	1	2017
7/1/2016	North A...	Regional	HR Manageme...	Provider...	557.5	88500.25	7	1	2017
7/1/2016	North A...	Regional	IT Consulting	Services	251.6	15338.3	7	1	2017
7/1/2016	South A...	Regional	IT Consulting	Life Scie...	16.58	3639.65	7	1	2017
7/1/2016	South A...	Strategic Account	HR Manageme...	Life Scie...	26.41	4292.06	7	1	2017

Figure 106 – Creating Calculated Columns in the Data Model with DAX Expressions

With the calculated columns in place, all that is left to do is to complete the PivotTable. From Power Pivot's **Home** tab, click **PivotTable**. In the PivotTable field list, drag *Region* and *Service Line* to the *Rows* quadrant, *Fiscal Year* and *Fiscal Quarter* to the *Columns* quadrant, and *Net Revenue* to the *Values* quadrant. The completed report, with formatting added, might resemble that pictured in **Figure 107**.

	B	C	D	E	F	G	H	I
3	Summary of Revenue by Region by Service Line							
4		2017				2017 Total	2018	Grand Total
5		Q1	Q2	Q3	Q4			
6	Asia	10,975,622.98	12,066,996.39	10,786,916.04	11,036,856.45	44,866,391.86	68,654,948.52	113,521,340.38
7	Advisory Services	1,658,032.94	1,576,453.77	1,301,879.93	1,892,261.05	6,428,627.69	12,360,427.55	18,789,055.24
8	HR Management	4,520,169.49	5,137,624.93	3,906,747.95	3,284,249.64	16,848,792.01	25,869,617.77	42,718,409.78
9	IT Consulting	4,121,422.18	4,625,545.13	4,924,517.10	5,449,134.85	19,120,619.26	25,900,438.80	45,021,058.06
10	Training	675,998.37	727,372.56	653,771.06	411,210.91	2,468,352.90	4,524,464.40	6,992,817.30
11	Europe						189,863,535.20	189,863,535.20
12	Advisory Services						32,556,283.45	32,556,283.45
13	HR Management						83,009,493.19	83,009,493.19
14	IT Consulting						62,567,734.62	62,567,734.62
15	Training						11,730,023.94	11,730,023.94
16	North America	91,318,839.23	95,165,462.14	89,099,541.53	91,518,465.70	367,102,308.60	565,576,194.20	932,678,502.80
17	Advisory Services	12,306,107.21	11,758,282.30	9,985,846.06	14,550,079.51	48,600,315.08	93,790,984.33	142,391,299.41
18	HR Management	43,044,885.85	44,382,045.00	37,769,699.45	31,831,110.70	157,027,741.00	238,928,484.35	395,956,225.35
19	IT Consulting	30,640,819.14	34,164,260.01	36,674,269.90	42,187,684.74	143,667,033.79	198,726,768.70	342,393,802.49
20	Training	5,327,027.03	4,860,874.83	4,669,726.12	2,949,590.75	17,807,218.73	34,129,956.82	51,937,175.55
21	South America	47,877,458.63	49,879,946.85	40,996,879.91	46,931,050.73	185,685,336.12	183,850,164.29	369,535,500.41
22	Advisory Services	6,529,300.05	5,802,681.90	3,151,025.60	7,354,537.67	22,837,545.22	31,050,687.45	53,888,232.67
23	HR Management	24,838,193.69	24,784,377.11	19,747,035.21	15,501,797.02	84,871,403.03	80,532,819.96	165,404,222.99
24	IT Consulting	13,908,957.85	16,457,588.23	15,400,544.65	22,633,945.64	68,401,036.37	60,822,949.84	129,223,986.21
25	Training	2,601,007.04	2,835,299.61	2,698,274.45	1,440,770.40	9,575,351.50	11,443,707.04	21,019,058.54
26	Grand Total	150,171,920.84	157,112,405.38	140,883,337.48	149,486,372.88	597,654,036.58	1,007,944,842.21	1,605,598,878.79

Figure 107 - PivotTable Using Calculated Columns for Fiscal Period Groupings

Creating Measures in a PivotTable using DAX

Consider the need for advanced data analysis capabilities in a PivotTables built from data models. For instance, the PivotTable pictured in **Figure 108** presents data by region, year, and service line, yet it paints an incomplete picture relative to revenue because it does not show the number of revenue-producing sectors in each region. In other words, a simple comparison of revenue between regions does not provide the depth of analysis necessary if the South America region does not operate in all of the industry sectors that the North America region does.

	A	B	C	D
1	Total Revenue			
2		Asia	North America	South America
3	2016	23,042,619.37	186,484,301.37	97,757,405.48
4	Advisory Services	3,234,486.71	24,064,389.51	12,331,981.95
5	HR Management	9,657,794.42	87,426,930.85	49,622,570.80
6	IT Consulting	8,746,967.31	64,805,079.15	30,366,546.08
7	Training	1,403,370.93	10,187,901.86	5,436,306.65
8	2017	59,213,911.76	492,164,199.88	185,310,610.10
9	Advisory Services	10,883,224.06	83,356,345.18	29,943,433.95
10	HR Management	20,569,737.15	196,904,075.29	75,934,567.15
11	IT Consulting	24,525,509.86	187,652,645.84	69,775,308.18
12	Training	3,235,440.69	24,251,133.57	9,657,300.82
13	2018	31,264,809.25	254,030,001.55	86,467,484.83
14	Advisory Services	4,671,344.47	34,970,564.72	11,612,816.77
15	HR Management	12,490,878.21	111,625,219.21	39,847,085.04
16	IT Consulting	11,748,580.89	89,936,077.50	29,082,131.95
17	Training	2,354,005.68	17,498,140.12	5,925,451.07
18	Grand Total	113,521,340.38	932,678,502.80	369,535,500.41

Figure 108 – Sample PivotTable Analyzing Revenue by Region, Year, and Service Line

To paint a more complete picture, you can add a *measure* to the PivotTable built from the data model. A measure is a formula that is created specifically for use in a PivotTable (or PivotChart) that uses Power Pivot data. You can create measures based on standard aggregation functions, such as COUNT or SUM, or you can define your own formula by using DAX expressions. You always use measures in the *Values* area of a PivotTable. If you need calculated results to be placed elsewhere in a PivotTable – such as a row or column – you should use a calculated column instead of a measure.

When you define a formula for an explicit measure, nothing happens until you add the measure into a PivotTable. When you add the measure, the formula is evaluated for each cell in the Values area of the PivotTable. Because a result is created for each combination of row and column headers, the result for the measure can be different in each cell. The definition of the measure that you create is saved with its source data table. It appears in the PivotTable Fields list and is available to all users of the workbook.

Returning to the example presented in Figure 108, you can use a function – **DISTINCTCOUNT** – that is only available using DAX to create a deeper analysis of the data. **DISTINCTCOUNT** identifies the number of distinct values that a PivotTable summarizes at each unique intersection of a column and a row. To add this calculation to the PivotTable shown in Figure 108, begin by clicking the arrow under Measures on the **Power Pivot** Ribbon and then choose **New Measure** to open the **Measure** dialog box pictured in **Figure 109**. In the dialog box, add the formula, using the **DISTINCTCOUNT** function and referring to the table and the column header in the table on which you want to base the calculation.

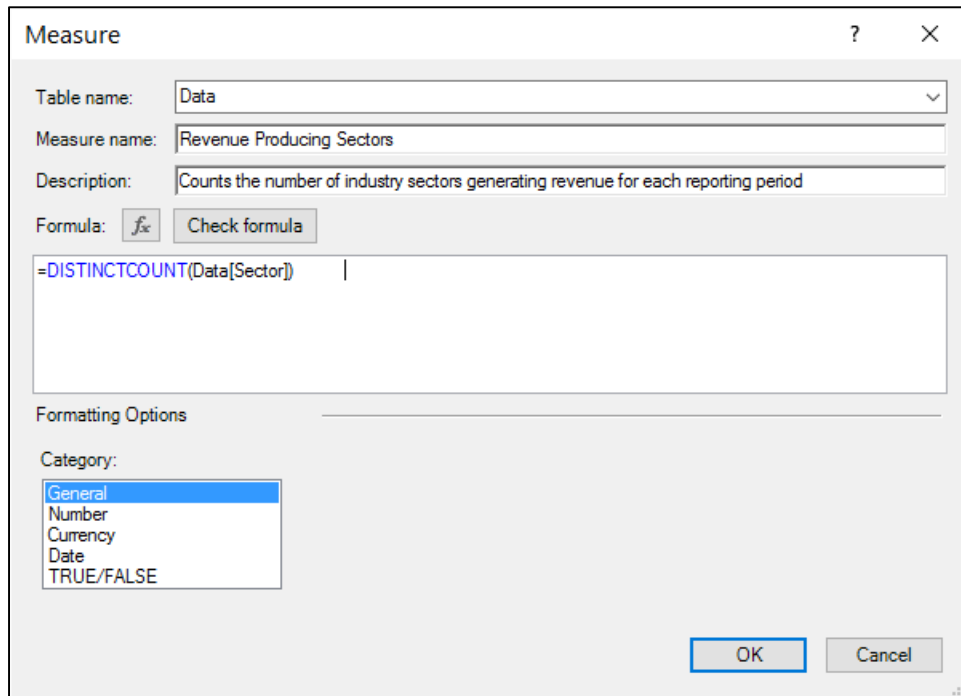


Figure 109 – Adding a Measure in Power Pivot’s Measure Dialog Box

Once you create the formula and click **OK**, Excel adds the measure to the PivotTable. Once it is added, you can use it in any other calculation in the PivotTable. **Figure 110** provides an illustration of the updated PivotTable, complete with the newly appended measures that offer greater insight into the revenues than was available previously.

	A	B	C	D	E	F	G
1	Asia			North America		South America	
2		Total Revenue	Revenue Producing Sectors	Total Revenue	Revenue Producing Sectors	Total Revenue	Revenue Producing Sectors
3	2016	23,042,619.37	22	186,484,301.37	24	97,757,405.48	21
4	Advisory Services	3,234,486.71	13	24,064,389.51	15	12,331,981.95	12
5	HR Management	9,657,794.42	20	87,426,930.85	22	49,622,570.80	19
6	IT Consulting	8,746,967.31	22	64,805,079.15	24	30,366,546.08	21
7	Training	1,403,370.93	13	10,187,901.86	14	5,436,306.65	11
8	2017	59,213,911.76	22	492,164,199.88	24	185,310,610.10	21
9	Advisory Services	10,883,224.06	16	83,356,345.18	18	29,943,433.95	15
10	HR Management	20,569,737.15	20	196,904,075.29	22	75,934,567.15	19
11	IT Consulting	24,525,509.86	22	187,652,645.84	24	69,775,308.18	21
12	Training	3,235,440.69	19	24,251,133.57	21	9,657,300.82	18
13	2018	31,264,809.25	22	254,030,001.55	24	86,467,484.83	22
14	Advisory Services	4,671,344.47	16	34,970,564.72	18	11,612,816.77	16
15	HR Management	12,490,878.21	21	111,625,219.21	23	39,847,085.04	21
16	IT Consulting	11,748,580.89	22	89,936,077.50	24	29,082,131.95	22
17	Training	2,354,005.68	15	17,498,140.12	17	5,925,451.07	15
18	Grand Total	113,521,340.38	22	932,678,502.80	24	369,535,500.41	22

Figure 110 – PivotTable with Calculated Field Added by Power Pivot

Creating Hierarchies to Manipulate Reports More Quickly

Given the large volume of data that most data models will likely contain, many data models contain inherent hierarchies or groupings of data. For example, if you build your data model from sales data of a large retail organization, one of the inherent hierarchies of data might be *Product Category*, *Product Subcategory*, and *Product* as shown in **Figure 111**.

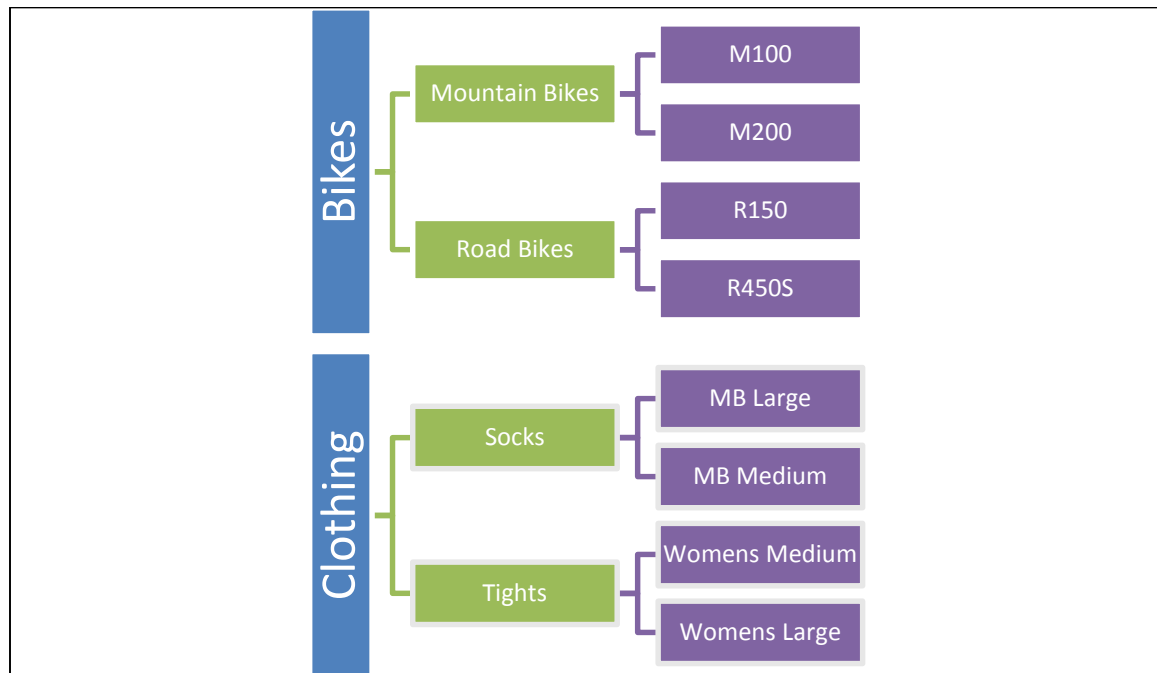


Figure 111 - Sample Product Hierarchy Inherent in Sales Data

In addition to the product-oriented hierarchy, you may also have an inherent geographic hierarchy along the lines of *Country*, *State/Province*, and *City* as depicted in **Figure 112**.

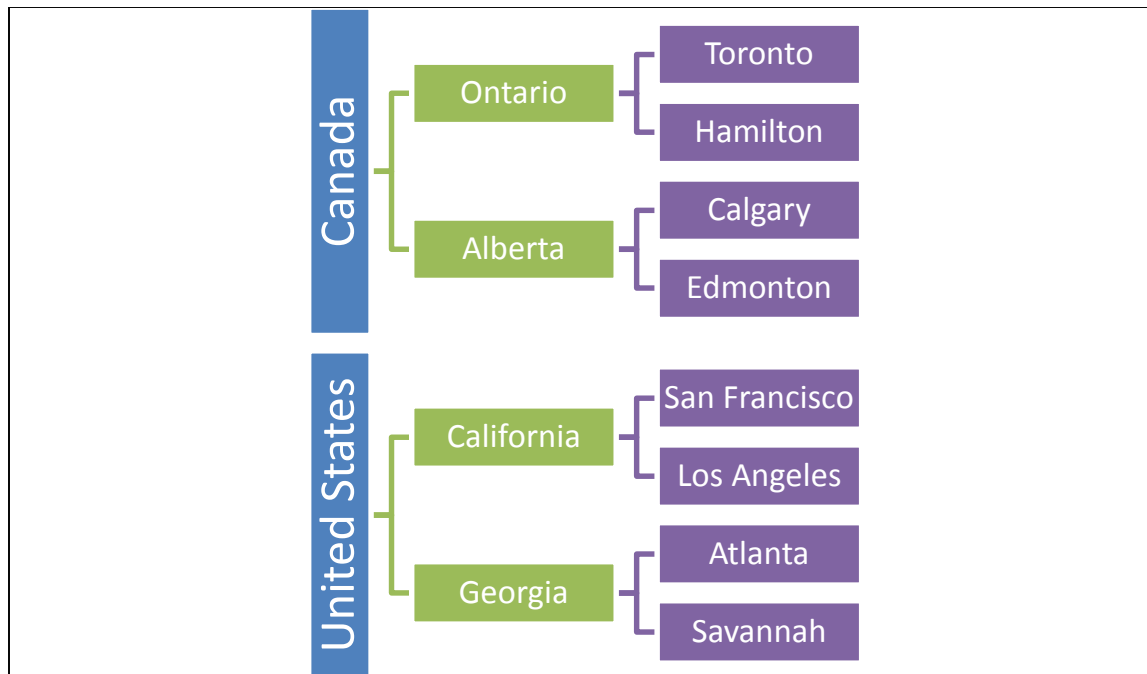


Figure 112 - Sample Geographic Hierarchy Inherent in Sales Data

Creating and storing one or more hierarchies for your data model can be useful because once you create a hierarchy, you can simply drag the hierarchy into your data model-based reporting object – Power Pivot, for example – instead of dragging all of the individual elements associated with the hierarchy. Not only is this process faster, but it helps to ensure you get consistent reporting across multiple reporting objects built from your data model.

Note that all of the fields you want to add to a hierarchy must belong to the same table in the data model. If you have fields from multiple tables from which you want to create a hierarchy, you can use the **RELATED** DAX function to “link” the fields together into a single table. For example, adding the DAX expressions pictured in **Figure 113** as calculated columns in the *Product* table links the *Name* fields from both the *Product Category* and *Product Subcategory* tables into the *Product* table; of course, once linked through the DAX expression, then building a hierarchy is possible.

<code>=RELATED(ProductCategory[Name])</code>
<code>=RELATED(ProductSubcategory[Name])</code>

Figure 113 - DAX Expressions Used to Link Product Data from Related Tables

To create and save the geographic hierarchy pictured in Figure 111, perform the following three steps.

1. In Power Pivot, click **Manage Data Model** and then open **Diagram View** from the **Home** tab of the Ribbon. Expand the Product table so that you can see all of the fields.

2. While pressing the **CTRL** key, click the *Name*, *ProductCategory*, and *ProductSubCategory* fields. With all three fields selected, right-click and choose **Create Hierarchy**.
3. Rename the resulting hierarchy to a name that is reflective of the nature of the data. In the example shown in **Figure 114**, *Product Hierarchy* is the new name assigned to the hierarchy.

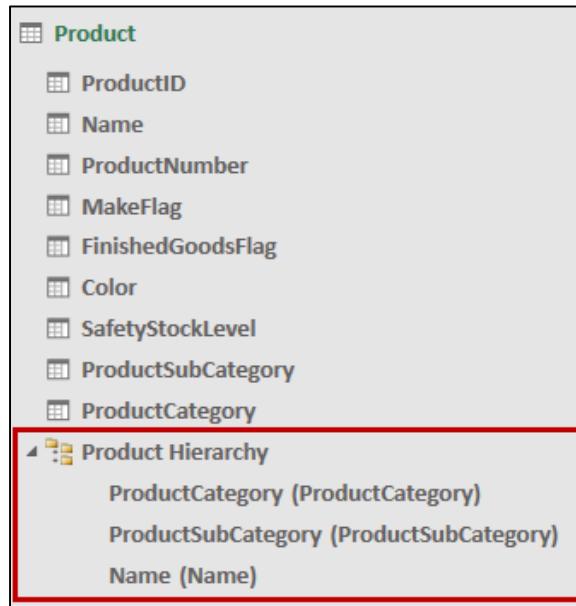


Figure 114 - Establishing a Product-oriented Hierarchy in a Data Model

To establish a geographic hierarchy, first add the two DAX expressions pictured in **Figure 115** to the *Address* table.

```
=RELATED(CountryRegion[Name])
=RELATED(StateProvince[Name])
```

Figure 115 - DAX Expressions Used to Link Geography Data from Related Tables

With the DAX expressions in place, you can create the geographic hierarchy by repeating the same three steps outlined previously, only this time acting on the *Country*, *State/Province*, and *City* fields in the *Address* table. The results of your efforts should resemble those shown in **Figure 116**.

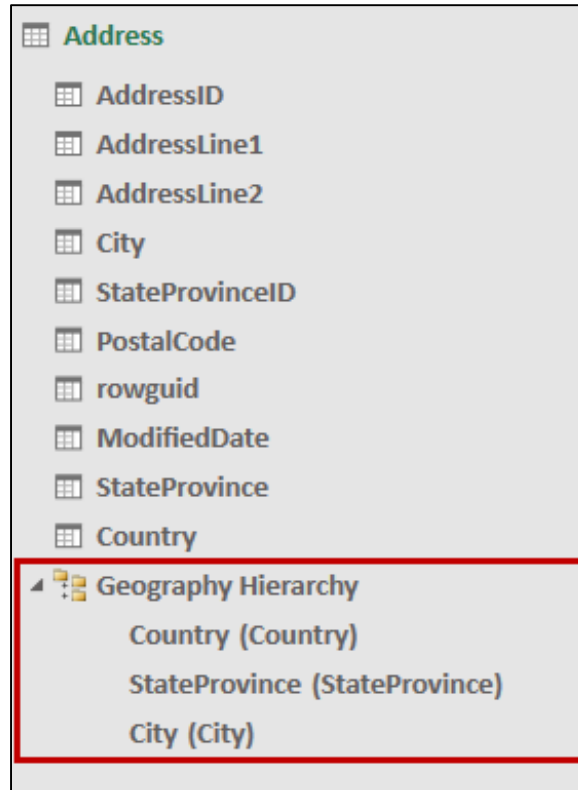


Figure 116 - Establishing a Geography-oriented Hierarchy in a Data Model

With both hierarchies in place, let us now see how you can use hierarchies to speed the process of creating and modifying PivotTables that use the data models where the hierarchies reside. Create a basic PivotTable by adding the *SubTotal* field from the *SalesOrderHeader* table to the *Values* quadrant in the PivotTable Field List. Also, add the *OrderDate* field from the *SalesOrderHeader* table to the *Columns* quadrant in the PivotTable Field List. Next, add the *Geography Hierarchy* from the *Address* table to the Rows quadrant to generate the PivotTable shown in **Figure 117**.

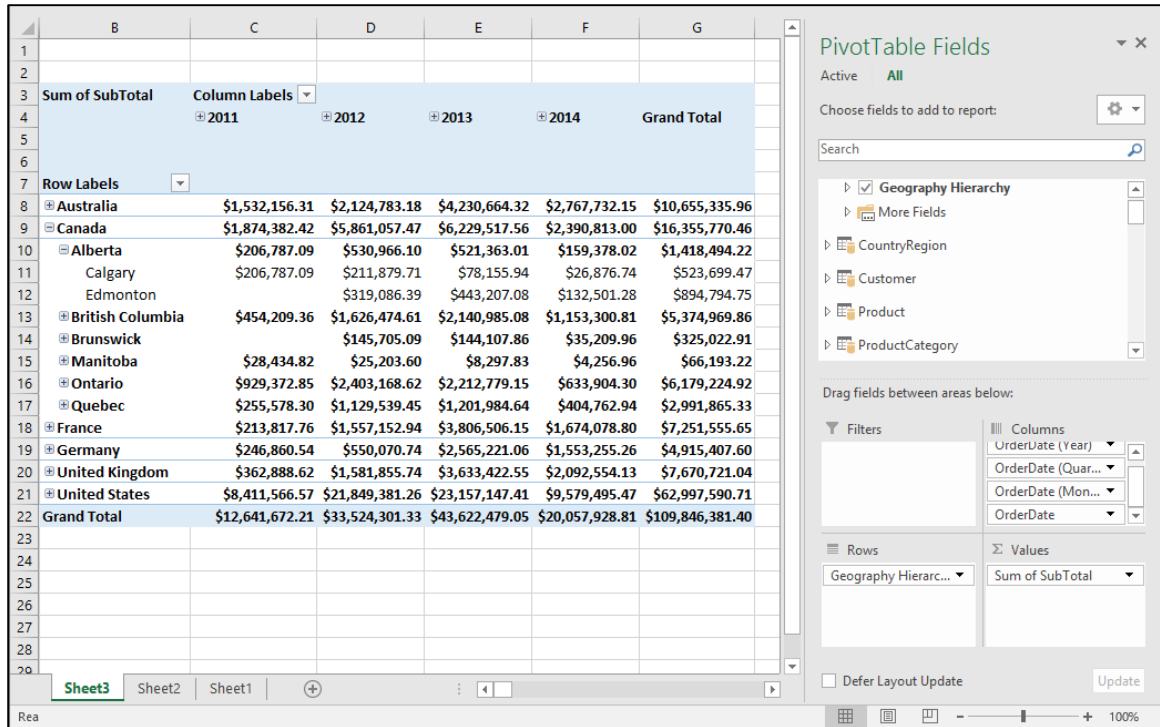


Figure 117 - PivotTable Including the Geography Hierarchy Created in the Data Model

Should your reporting needs change such that you need to see a product-oriented report, simply remove the *Geographic Hierarchy* from the rows quadrant and replace it with the *Product Hierarchy* from the *Product* table. Also, remove the *SubTotal* field from the values quadrant and replace it with the *LineTotal* field from the *SalesOrderDetail* table. Doing so generates the PivotTable pictured in **Figure 118**.

Note that the use of hierarchies is purely optional. However, adding hierarchies to your data models and using them in the reports you build from your data models can streamline the process of constructing and editing these reports substantially.

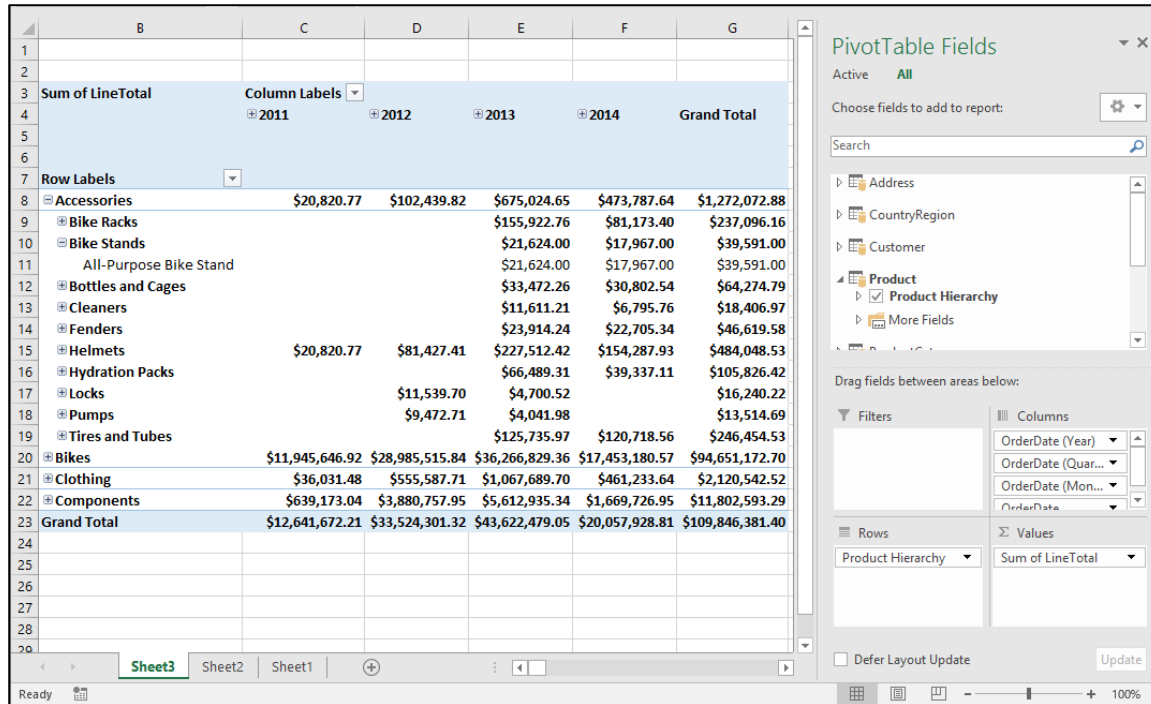


Figure 118 - PivotTable Including the Product Hierarchy Created in the Data Model

Using Perspectives to Control the Amount of Data Available

Perspectives offer another approach to managing data models. Using perspectives, you can create different views of your data to be used by different user groups or in different scenarios when working the data emanating from the data model. For example, you might want to create a perspective that can be used by Regional Sales Managers and a separate perspective that can be used by Product Managers. Likewise, you might create a perspective that can be used by the Accounting team and a separate perspective that is used by the Business Planning group.

To create a perspective, choose **Manage Data Model** and then click the **Advanced** tab of the Ribbon. On the left side of the Ribbon, click **Create and Manage** in the Perspectives group as shown in **Figure 119**.

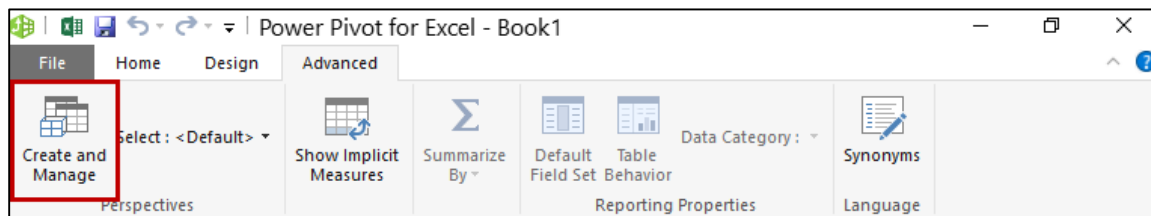


Figure 119 - Choosing to Create a New Perspective for a Data Model

In the resulting Perspectives dialog box, click **New Perspective**, enter a name for the perspective you are creating, and indicate which tables and fields within selected tables will be accessible when using this perspective. **Figure 120** depicts a new perspective entitled *Regional Sales* that

currently excludes access to the **Product** table. Of course, click **OK** to complete the process of adding a perspective.

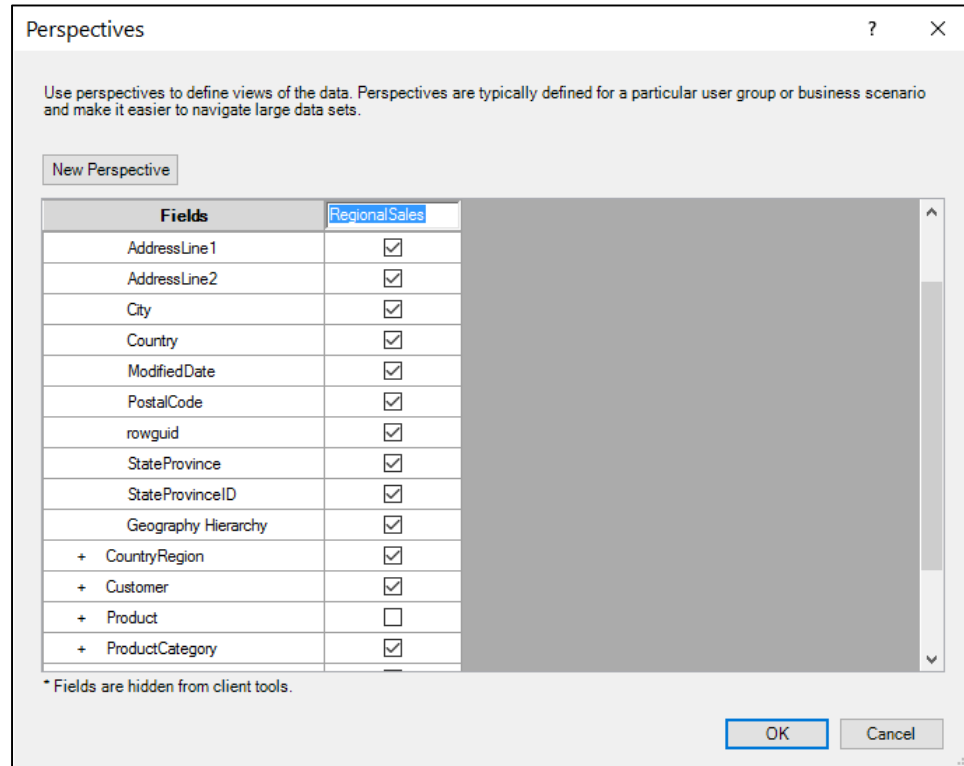


Figure 120 - Adding a Perspective to a Data Model

Repeat the process outlined above for as many perspectives as you need. **Figure 121** illustrates a scenario in which four perspectives are available in the data model, each with access to different tables and fields within those tables.

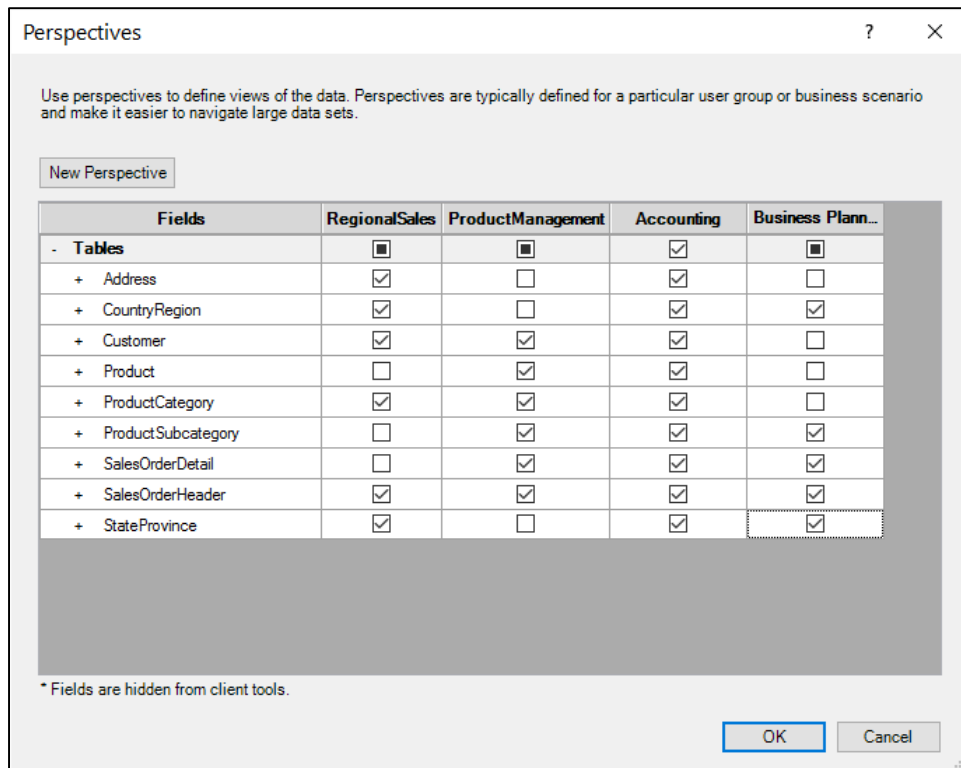


Figure 121 - Multiple Perspectives Available for a Data Model

With the perspectives defined, you can then indicate which perspective you want to use when working with the data model. To do so, in the **Perspectives** group on the Ribbon, click the drop-down arrow next to **Select** and choose the appropriate perspective for a particular task as shown in **Figure 122**. Note that as you select different perspectives, the tables and fields that are accessible will differ based on the definition of that perspective.

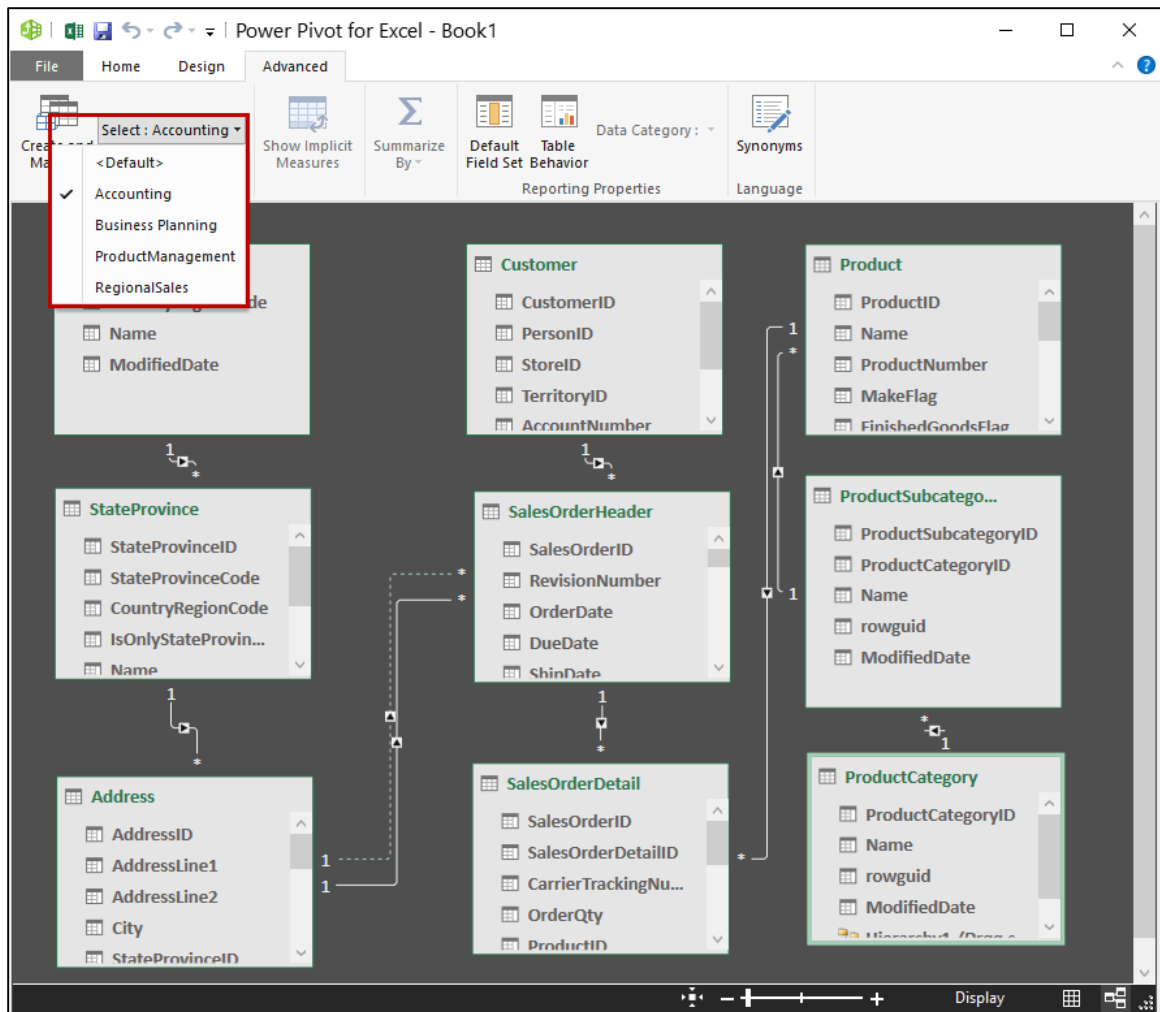


Figure 122 - Choosing among Available Perspectives When Working with Data Models

Creating Reports and Summaries from Data Models

As indicated in previous chapters, tables and data models provide exceptional capabilities for storing and managing data. From tables, you can build formulas, charts, PivotTables, and PivotCharts quickly and easily. However, many Excel users struggle with what they will do with data that resides in a data model, perhaps because this data is not readily accessible by traditional Excel formulas. In this chapter, you will learn how to transform the data in data models into meaningful and actionable reports.

Learning Objectives

Upon completing this chapter, you should be able to:

- Create interactive reports in Power View based on data models;
- Utilize data models to create geospatial visualizations using 3D Maps; and
- Construct PivotCharts that use data models as data sources.

Construct PivotTables and PivotCharts That Use Data Models as Data Sources

To build a data model using the tools available in Power Pivot, begin by choosing a data source from the **Get External Data** group on the **Home** tab of the Power Pivot window. In our first example, we will click the **From Database** and choose **From Access** as shown in **Figure 123**.

In the first pane of the wizard, click **Browse** and then select *ContosoSales.accdb* in the **Open** dialog box and click **Open**. Click **Next** to open the second pane of the wizard. Choose **Select from a list of tables and views to choose the data to import** and click **Next**. Select the following tables: **DimChannel**, **DimDate**, **DimProduct**, **DimProductSubcategory**, and **Fact Sales**. If you are unsure which tables contain the data elements that you need, click **Preview & Filter**, which will allow you to see the data within each table and also allow you to filter the data to just those records required. If you are unsure as to whether you may need data stored in

tables related to those selected, click **Select Related Tables**, and Power Pivot will select related tables automatically.

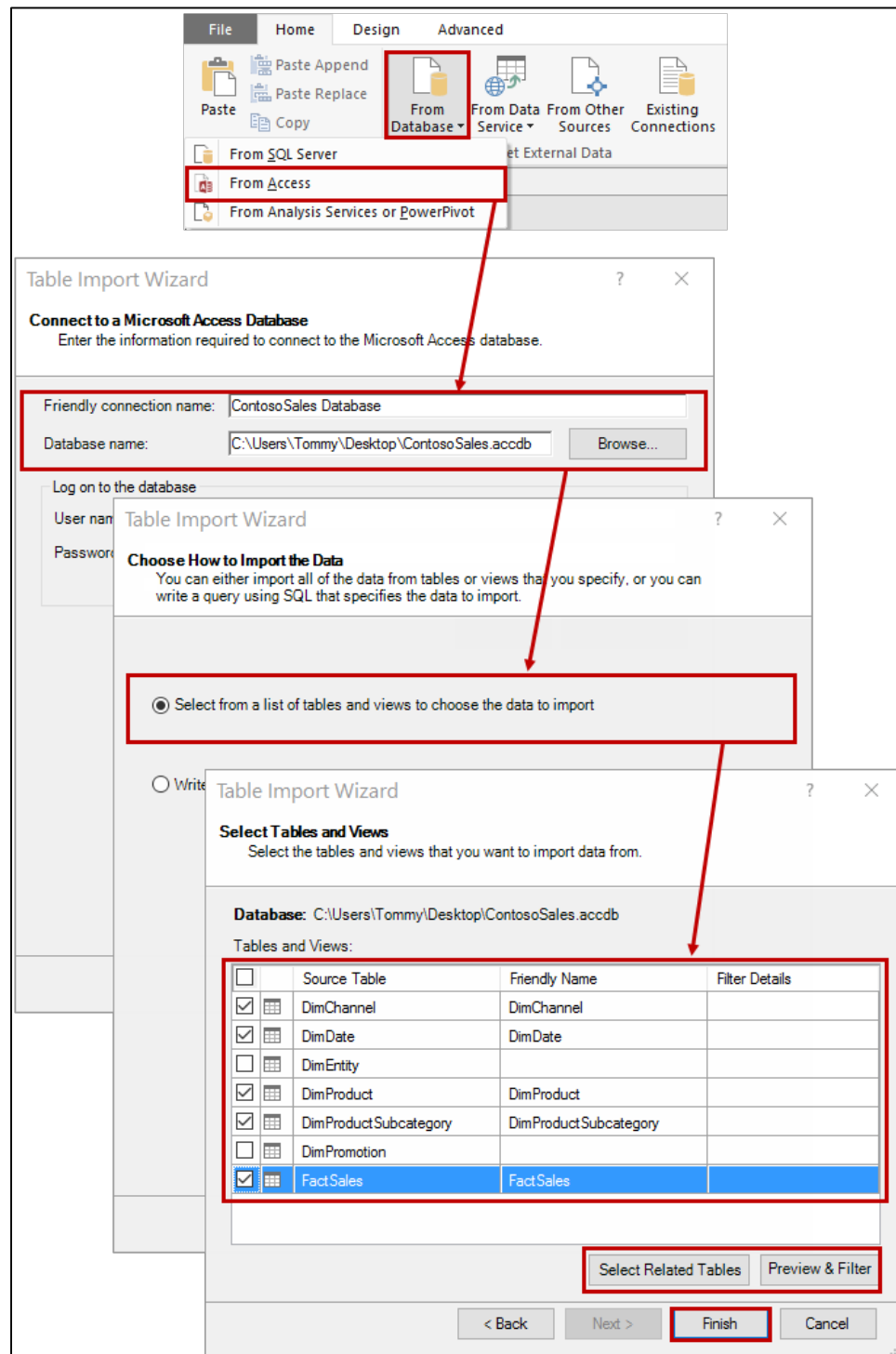


Figure 123 – Importing Data from an Access Database

Click **Finish** to open the final pane in the wizard – the **Table Import Wizard** dialog box – and begin the data import. When the data is imported successfully, click **Close** to display the tables in Power Pivot. Power Pivot displays each table in the data model on a separate tab as shown in **Figure 124**. Note that the *FactSales* table contains over 2.28 million records – roughly twice the data than would fit in an “ordinary” Excel workbook.

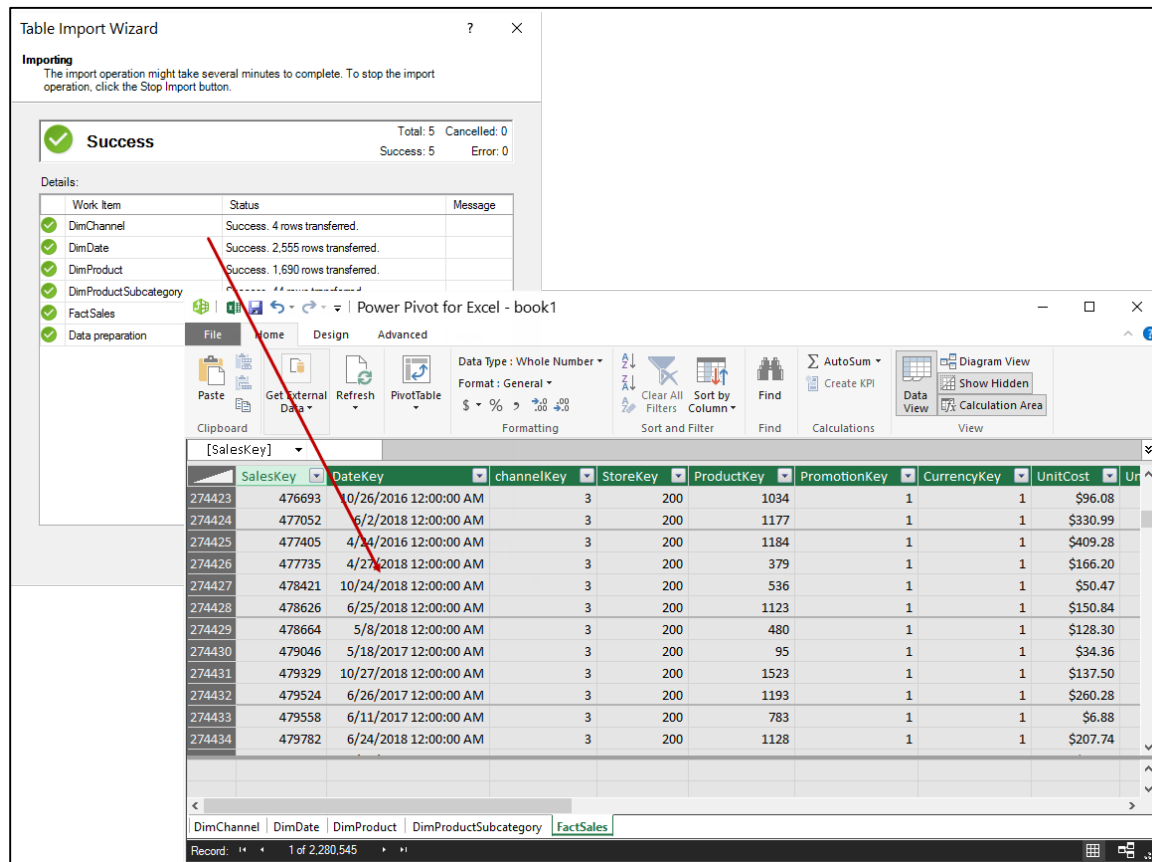


Figure 124 – Completing the Data Import

Once the data has imported successfully, click **Close** to display the tables in Power Pivot. Each table appears on a single tab as shown in Figure 124.

In this example, we are building a report that displays sales by product category by year. Unfortunately, the product category information is stored in another database, so we need to import more data. Having to build reports from data stored in two or more databases is a common problem in many companies. For example, it's not unusual for accounting information stored in the general ledger to be matched with production or scheduling information stored in another system. Power Pivot allows users to overcome this obstacle by allowing them to import and relate data from multiple sources.

Importing the second set of data will follow the same process as the one for the previous set of tables, except this time the name of the database to be imported is *ProductCategories.accdB*, which contains a single table. Once the data is imported, we need to relate the data so that the information in one table is logically linked to the others. To begin the process, click **Design**,

Create Relationships to open the **Create Relationships** dialog box as shown in **Figure 125**. Select the *DimProductCategory* table as *Table 1* and the *DimProductSubcategory* table as *Table 2*. Then, select the *ProductCategoryKey* from both tables to build the relationship between the tables. Click **OK** to complete the process.

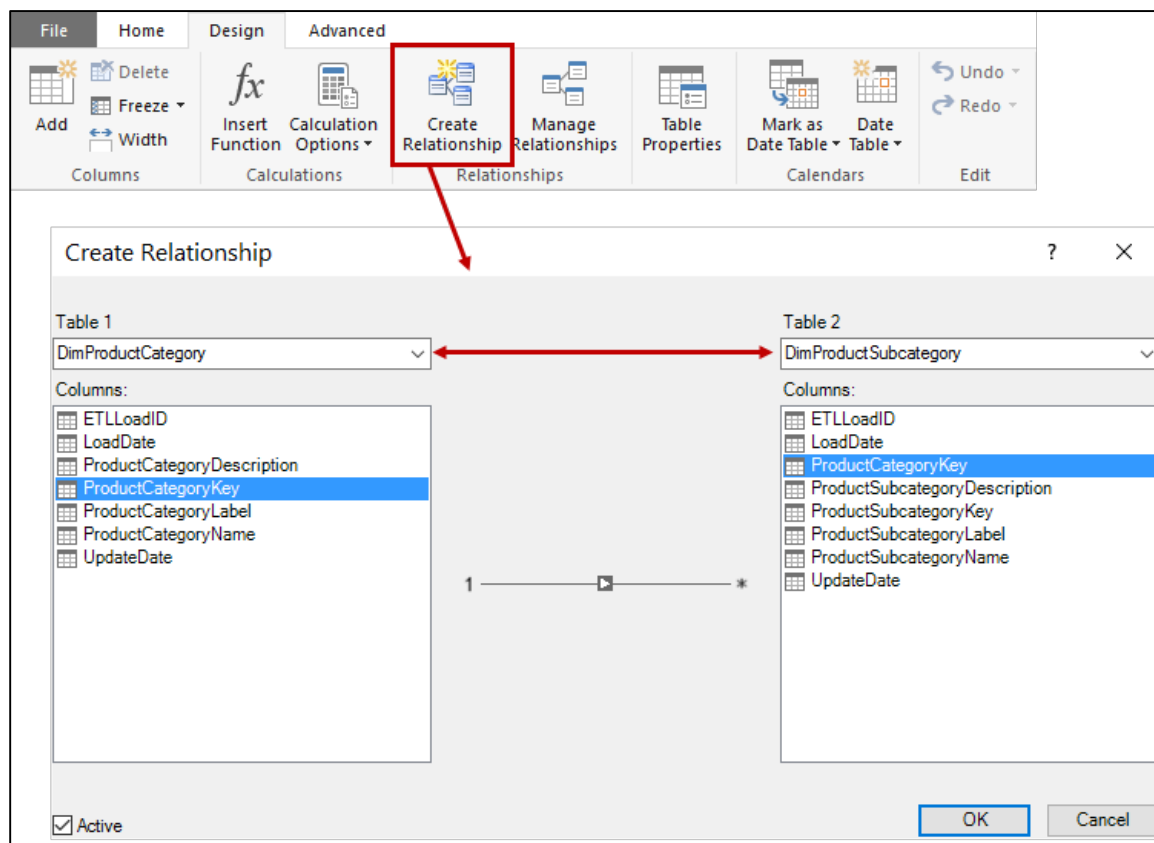


Figure 125 – Managing Table Relationships in Power Pivot

In fact, you can choose to create and manage relationships in the data model from several different entry points in Power Pivot. For instance, instead of choosing **Create Relationships**, you could opt to click **Manage Relationships** and create and manage relationships in that dialog box. Likewise, you could choose **Diagram View** from the **Home** tab of the PowerPivot window and manage relationships between tables using drag-and-drop functionality as pictured in **Figure 126**. Note that Power Pivot attempts to detect establish relationships among the tables in the data model automatically on import, minimizing the amount of time required to manage relationships.

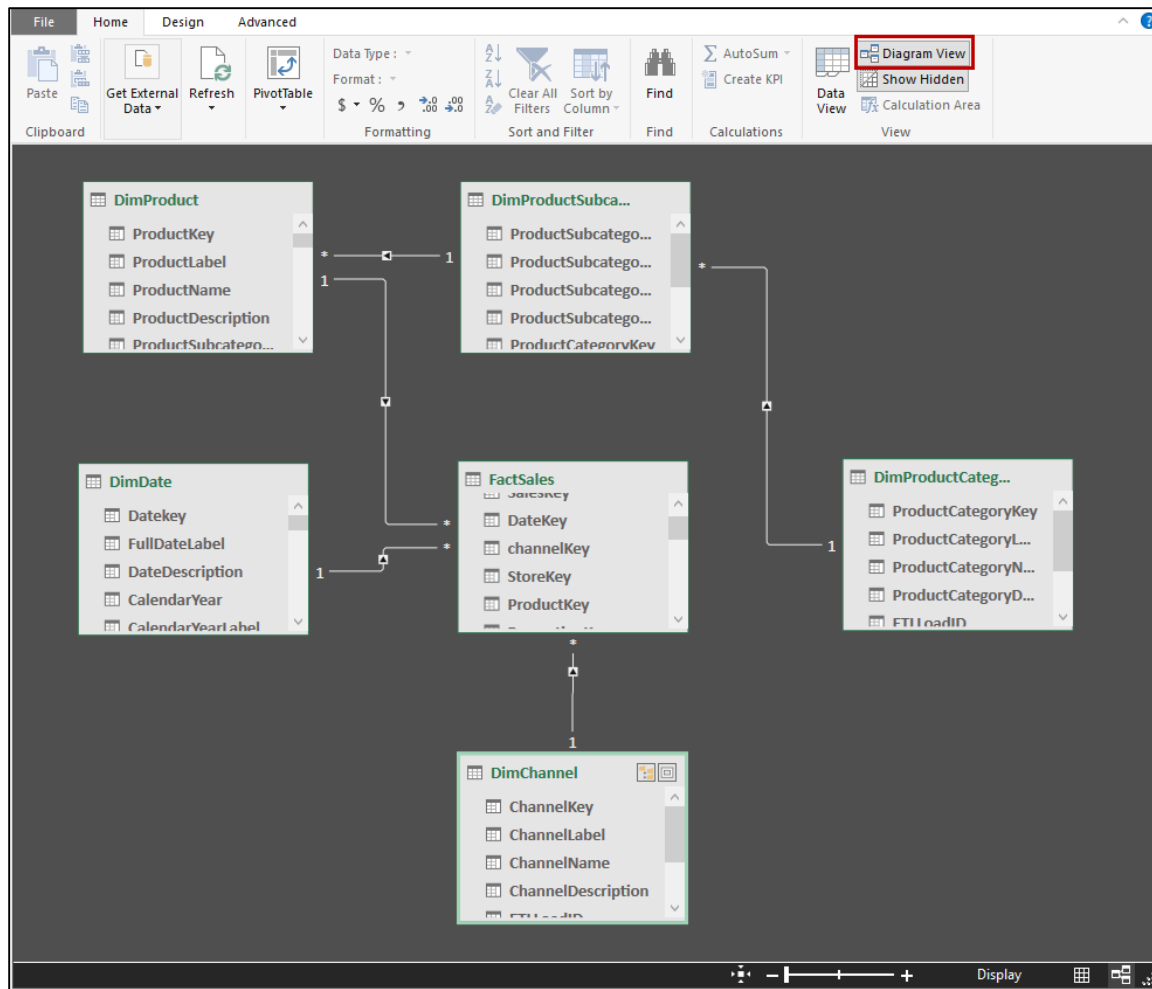


Figure 126 – Managing Relationships between Tables in Power Pivot’s Diagram View

Creating a PivotTable Report in Power Pivot

Now that the data has been imported and the tables related, it's time to create a PivotTable report. To create a PivotTable based on Power Pivot data, begin by clicking **PivotTable** on the **Home** tab of the Power Pivot Ribbon. Next, select what type of PivotTable or PivotChart object to create as shown in **Figure 127**.

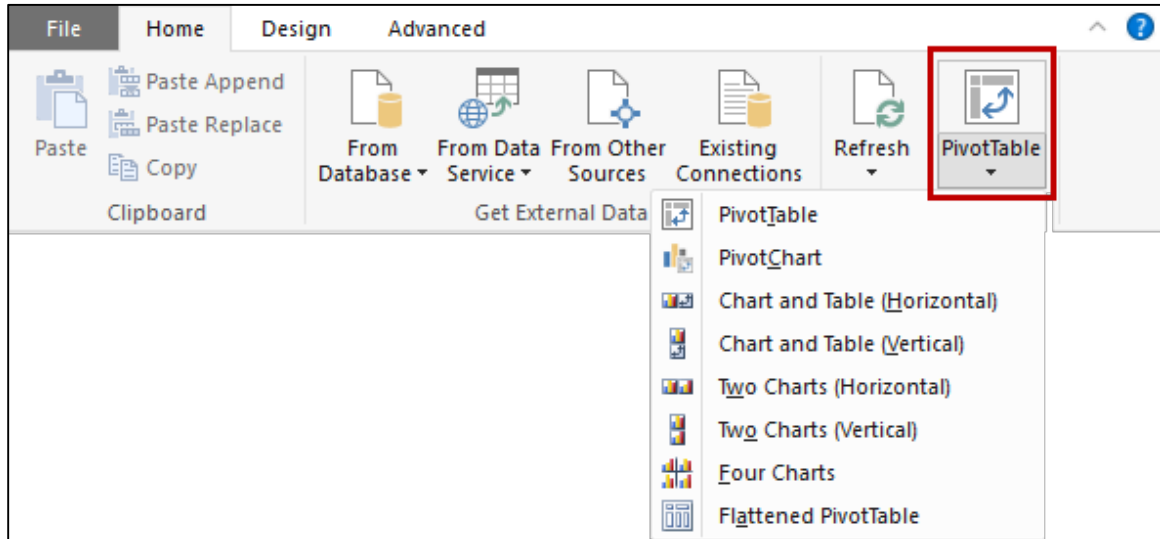


Figure 127 – Starting the PivotTable / PivotChart Process in Power Pivot

In this example, click **PivotTable** and then choose to put the report on a **New Worksheet**. Power Pivot then displays the familiar PivotTable placeholder and task pane in Excel. The process for building a PivotTable is virtually identical to that of building a PivotTable based on raw data. The only significant differences are that 1) the List contains fields from multiple tables as shown in **Figure 128** and that 2) certain functionality, such as the grouping of items available in Excel PivotTables, is disabled. To overcome this limitation, create calculated fields in the data model to provide any necessary data groupings.

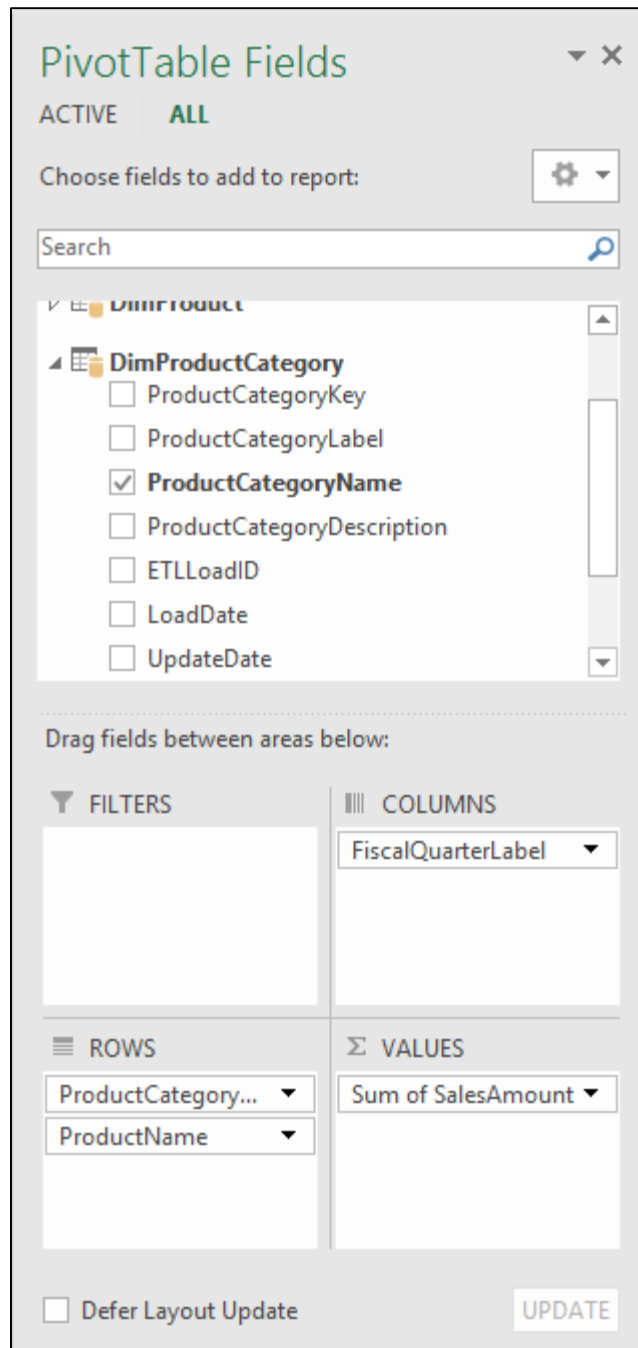


Figure 128 – PivotTable Field List in Power Pivot

To create the PivotTable from the data model, add the following fields to the identified drop-areas.

Table	Field	Drop Area
DimProductCategory	ProductCategoryName	Row Labels
DimProduct	ProductName	Row Labels
FactSales	SalesAmount	Values
DimDate	FiscalQuarterLabel	Column Labels
DimDate	FiscalYear	Slicer
DimChannel	ChannelName	Slicer

Add formatting, filtering, sorting, and slicers to the report as desired to complete the PivotTable. The completed report built on 2.28 million records and with slicers to add user interactivity should resemble the one in **Figure 129**.

FiscalYear		Top Five Selling Items Per Category					
			Q1	Q2	Q3	Q4	Grand Total
2016	Computers		\$ 3,725,721	\$ 2,687,700	\$ 2,097,100	\$ 4,007,702	\$ 12,518,223
2017	Contoso Projector 1080p X980 Black		\$ 410,920	\$ 688,500	\$ 376,610	\$ 835,839	\$ 2,311,868
2018	Contoso Projector 1080p X980 White		\$ 368,692	\$ 642,600	\$ 381,200	\$ 1,202,580	\$ 2,595,071
	Proseware Projector 1080p DLP86 Black		\$ 780,938	\$ 474,810	\$ 207,167	\$ 947,621	\$ 2,410,535
	Proseware Projector 1080p DLP86 White		\$ 851,284	\$ 399,840	\$ 618,503	\$ 488,305	\$ 2,357,931
	Proseware Projector 1080p LCD86 White		\$ 1,313,888	\$ 481,950	\$ 513,621	\$ 533,358	\$ 2,842,817
	Cameras and camcorders		\$ 2,045,951	\$ 1,110,700	\$ 1,112,012	\$ 1,568,868	\$ 5,837,531
	Fabrikam Independent Filmmaker 1" 25mm X400 Blue		\$ 348,300	\$ 162,000	\$ 345,870	\$ 456,516	\$ 1,312,686
	Fabrikam Trendsetter 1" 25mm X400 Grey		\$ 375,600	\$ 300,000	\$ 188,500	\$ 289,800	\$ 1,153,900
	Fabrikam Trendsetter 1/3" 8.5mm X200 Grey		\$ 374,949	\$ 189,620	\$ 176,846	\$ 373,052	\$ 1,114,467
	Fabrikam Trendsetter 1/3" 8.5mm X200 White		\$ 375,048	\$ 319,360	\$ 259,680	\$ 203,792	\$ 1,157,880
	Fabrikam Trendsetter 1/3" 8.5mm X200 Blue		\$ 572,054	\$ 139,720	\$ 141,117	\$ 245,708	\$ 1,098,598
	TV and Video		\$ 1,803,657	\$ 828,980	\$ 609,907	\$ 1,251,328	\$ 4,493,873
	Contoso Home Theater System 7.1 Channel M1700 Black		\$ 310,083	\$ 204,470	\$ 84,099	\$ 349,377	\$ 948,030
	Contoso Home Theater System 7.1 Channel M1700 Brown		\$ 350,088	\$ 168,910	\$ 272,656	\$ 162,865	\$ 954,519
	Contoso Home Theater System 7.1 Channel M1700 Silver		\$ 445,567	\$ 142,240	\$ 115,392	\$ 204,114	\$ 907,313
	Contoso Home Theater System 7.1 Channel M1700 White		\$ 349,244	\$ 213,360	\$ 31,560	\$ 310,972	\$ 905,135
	SV Car Video LCD7W M7080 Silver		\$ 348,675	\$ 100,000	\$ 106,200	\$ 224,000	\$ 778,875
	Cell phones		\$ 655,568	\$ 327,810	\$ 494,361	\$ 652,401	\$ 2,130,139
	The Phone Company PDA Handheld 4.7 inch L650 Black		\$ 222,508	\$ 50,570	\$ 59,906	\$ 150,699	\$ 483,683
	The Phone Company PDA Palm 4.7 inch L850 Black		\$ 96,137	\$ 47,760	\$ 131,539	\$ 114,385	\$ 389,821
	The Phone Company PDA Phone 4.7 inches L360 White		\$ 68,320	\$ 96,480	\$ 61,426	\$ 185,081	\$ 411,306
	The Phone Company PDA Wifi 4.7-inch L290 Black		\$ 115,691	\$ 41,800	\$ 120,688	\$ 148,580	\$ 426,759
	The Phone Company PDA Wifi 4.7-inch L290 White		\$ 152,912	\$ 91,200	\$ 120,802	\$ 53,656	\$ 418,570
	Music, Movies and Audio Books		\$ 603,846	\$ 249,391	\$ 325,427	\$ 414,918	\$ 1,593,582
	Contoso DVD 15-Inch Player Portable L200 Black		\$ 100,946	\$ 60,898	\$ 65,248	\$ 68,844	\$ 295,935

Figure 129 – Completed Power Pivot PivotTable

The PivotTable can be refreshed when data in the data source changes. Simply click the **Refresh** button on the **PivotTable Tools, Analyze** tab of the Ribbon. Beginning with Power Pivot in Excel 2013, you can also drill to detail. Double-click on the cell of interest to create a new worksheet that contains the detail records underlying the cell. Note that the data reported is created from a drill-through query to the table containing the **Values** field in the PivotTable.

As you can see even in this simple example, Power Pivot provides very powerful features for reporting and analyzing very large data sets without degradation in performance. However, the data retrieved and managed in the data model is stored for future reference and use in the Excel

workbook in which it was created. Retrieving unnecessary tables, fields, and records will expand the size of your workbooks significantly. Whenever you use Power Pivot, make sure to pare down your data model so that it contains only the data required for the report or analysis.

Power Pivot can produce a number of report and chart layouts. Earlier, when we chose to create a single PivotTable, we had the opportunity to choose from multiple PivotTables, a selection of PivotTable and PivotChart combination layouts, and a selection of multiple PivotChart layouts. **Figure 130** displays a simple PivotChart dashboard created from one of Power Pivot's built-in layouts. The headings and gridlines have been turned off to enhance the dashboard's appearance.

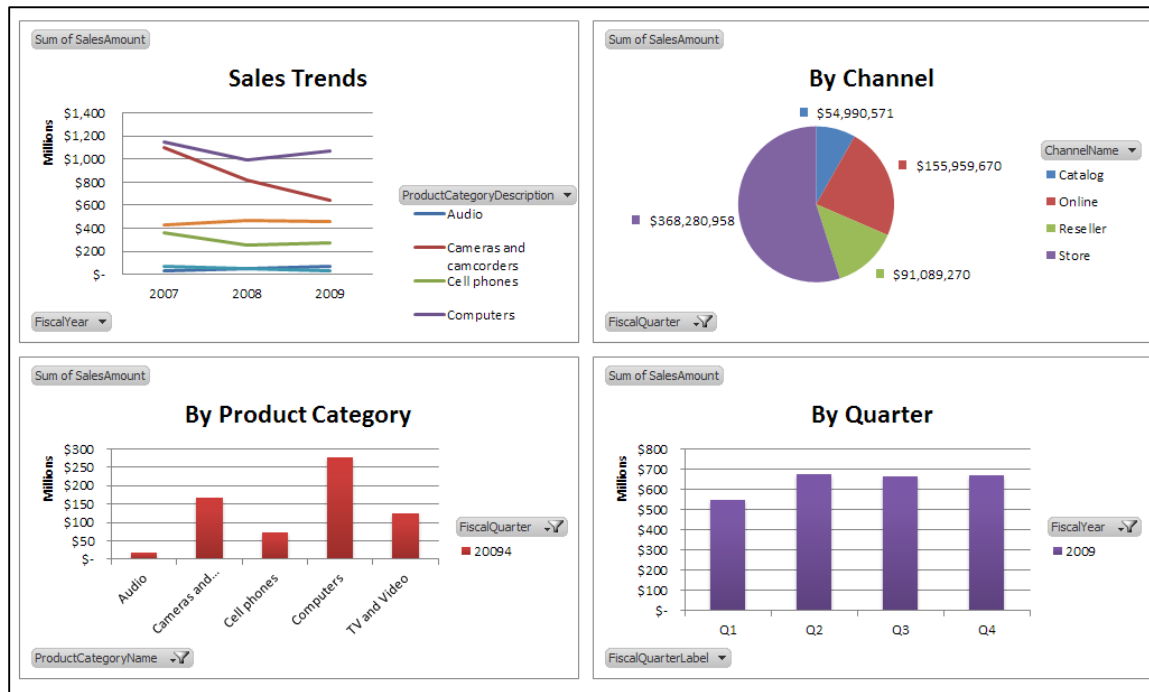


Figure 130 – Power Pivot Dashboard Built from Four PivotCharts

Create Interactive Reports Using Power View

Microsoft added **Power View**, a new reporting tool, to the 2013 release of Excel.⁴ Power View helps you to build presentation-quality, dynamic reporting objects that facilitate visual, interactive reporting on potentially very large data sets. Using Power View, you can create dashboard-style “widgets” to facilitate reporting, analysis, and business intelligence (BI) functions in your organization.

Organizing Your Data for Power View

Before you can build a Power View report, you must organize your data in tables. Your Power View report can incorporate data from a single table, or it can include data from multiple tables

⁴ Not all versions of Excel 2013 or Excel 2016 contain the Power View feature.

if you create relationships between those tables in the form of data models. Other than converting the source data for Power View to a table and creating relationships between the tables if multiple tables will serve as your data source, no other special preparation is necessary. The table pictured in **Figure 131** will serve as the basis for building a Power View dashboard; it contains over 400,000 rows of sales information from twelve cities.

	A	B	C	D	E	F	G	H
1	Date	City	Product	UnitCost	UnitPrice	SalesQuantity	TotalCost	SalesAmount
2	1/1/2017	Atlanta, Georgia	1749	36.11	109.00	5	180.55	545.00
3	1/1/2017	Atlanta, Georgia	1046	143.48	312.00	5	717.40	1,560.00
4	1/1/2017	Atlanta, Georgia	1329	7.81	16.99	11	85.91	186.89
5	1/1/2017	Atlanta, Georgia	1345	10.12	22.00	11	111.32	242.00
6	1/1/2017	Atlanta, Georgia	1454	91.51	199.00	5	457.55	995.00
7	1/1/2017	Atlanta, Georgia	1900	152.94	299.99	10	1,529.40	2,999.90
8	1/1/2017	Atlanta, Georgia	1498	137.96	300.00	16	2,207.36	4,800.00
9	1/1/2017	Atlanta, Georgia	661	75.54	228.00	22	1,661.88	5,016.00
10	1/1/2017	Atlanta, Georgia	176	58.36	126.90	10	583.60	1,269.00

Figure 131 - Sample Data for Power View

Creating a Simple Power View Dashboard

Your Power View dashboard will reside on a new worksheet in the file containing your data. To create the Power View worksheet, choose **Power View** from the **Insert** tab of the Ribbon in Excel 2013. However, if you are using Excel 2016, you will need to customize the Ribbon to add the **Power View** tab to the Ribbon and then choose **Power View** from the newly added tab in order to insert a Power View sheet into your workbook. Note that Power View sheets are different than regular Excel worksheets; they can only hold Power View dashboards. If desired, you may rename this sheet, move it, or delete it just as you would with any other worksheet.

As shown in **Figure 132**, upon creating your Power View worksheet, you will see a list of all available tables on the right side of the window in the **Power View Field List**.

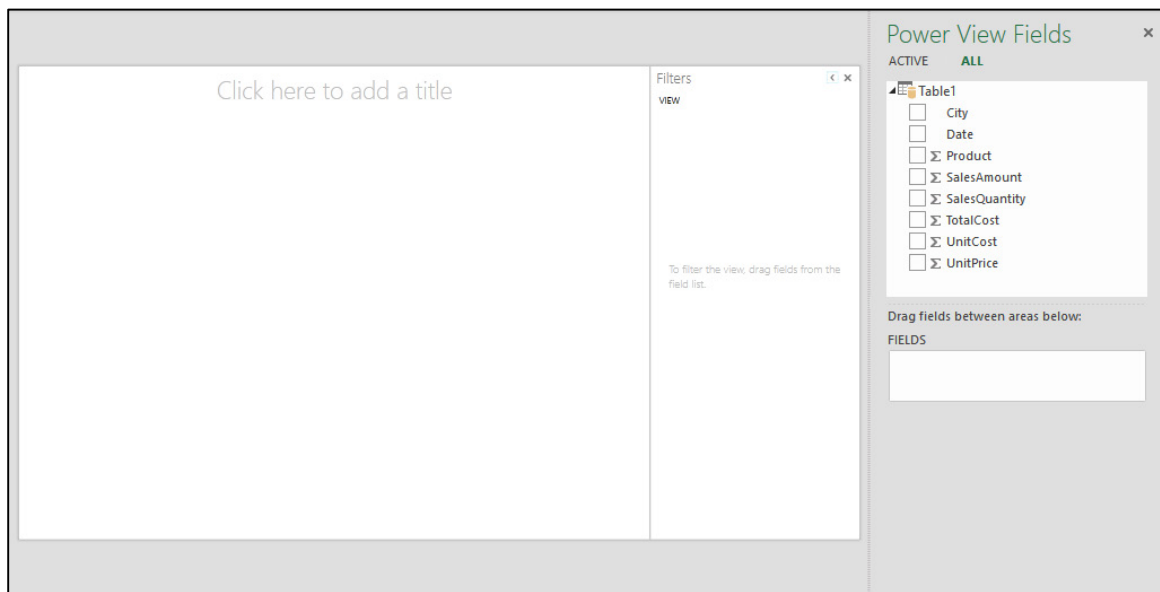


Figure 132 - Power View Field List

If you routinely work with PivotTables, the Power View Field List will resemble the PivotTable Field List in Excel. From the Power View Field List, select the fields you want to include in your dashboard. As you do so, Excel adds the values from the fields to the Power View Dashboard in a Power View **Table** as shown in **Figure 133**. You can resize the table as necessary by clicking and dragging on its edges.

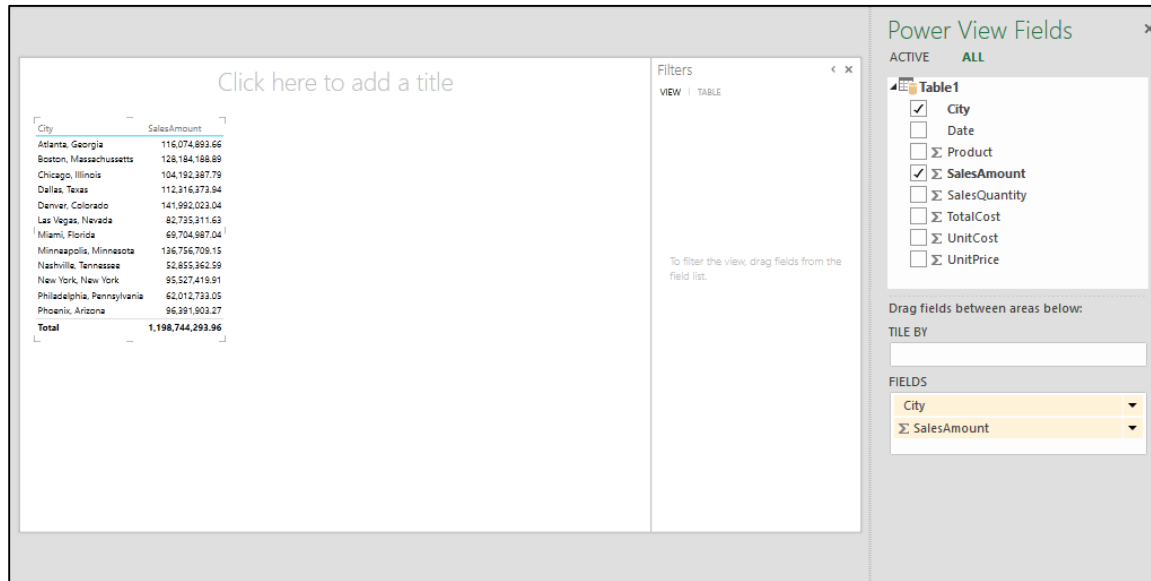


Figure 133 - Creating a Power View Table

Adding Visualization to a Power View Dashboard

After creating your table in Power View, you can convert it to another object, including a matrix view, a card view, a map, or any of nine chart types. For example, choosing to add the **Total Cost** field and choosing to convert the table to a **Clustered Column** chart causes Power View to display the data as shown in **Figure 134**.

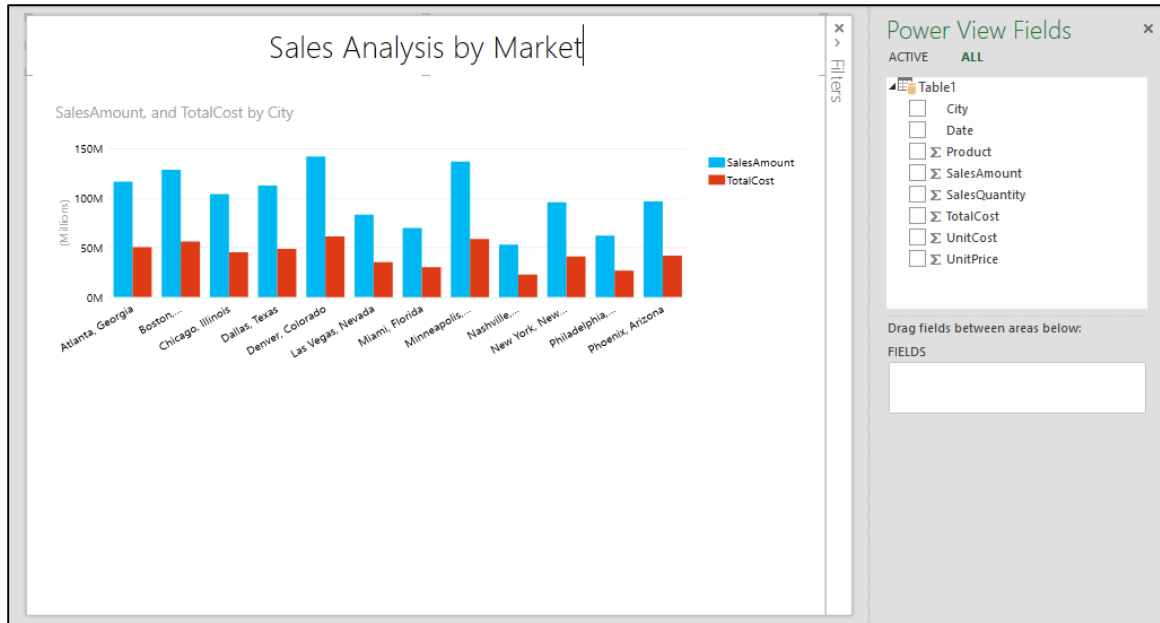


Figure 134 - Power View Table Converted to a Clustered Column Chart

Likewise, changing the visualization to a map causes Power View to display the information as shown in **Figure 135**.

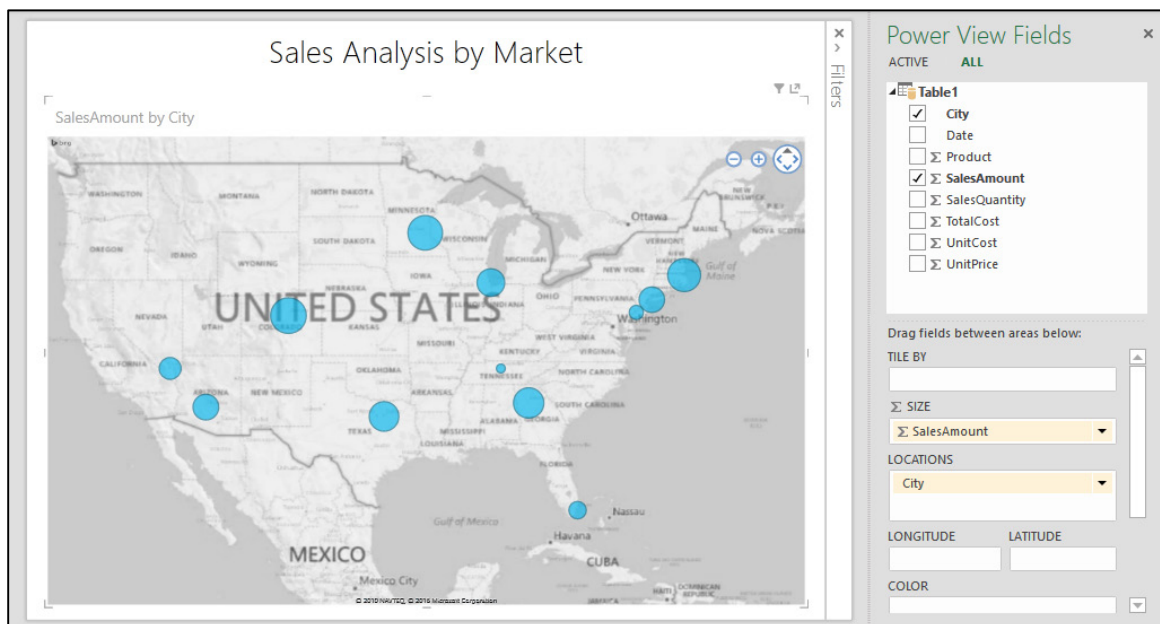


Figure 135 - Changing the Power View Visualization to a Map

Adding Additional Elements to the Dashboard

Power View does not limit you to a single element or “widget” on your dashboard. To add additional elements to the dashboard, just click and drag a field from the Field List to a blank area on the dashboard. For example, adding the *Date* and *SalesAmount* fields to the dashboard

and changing the visualization to a line chart causes the Power View worksheet to appear as shown in **Figure 136**.

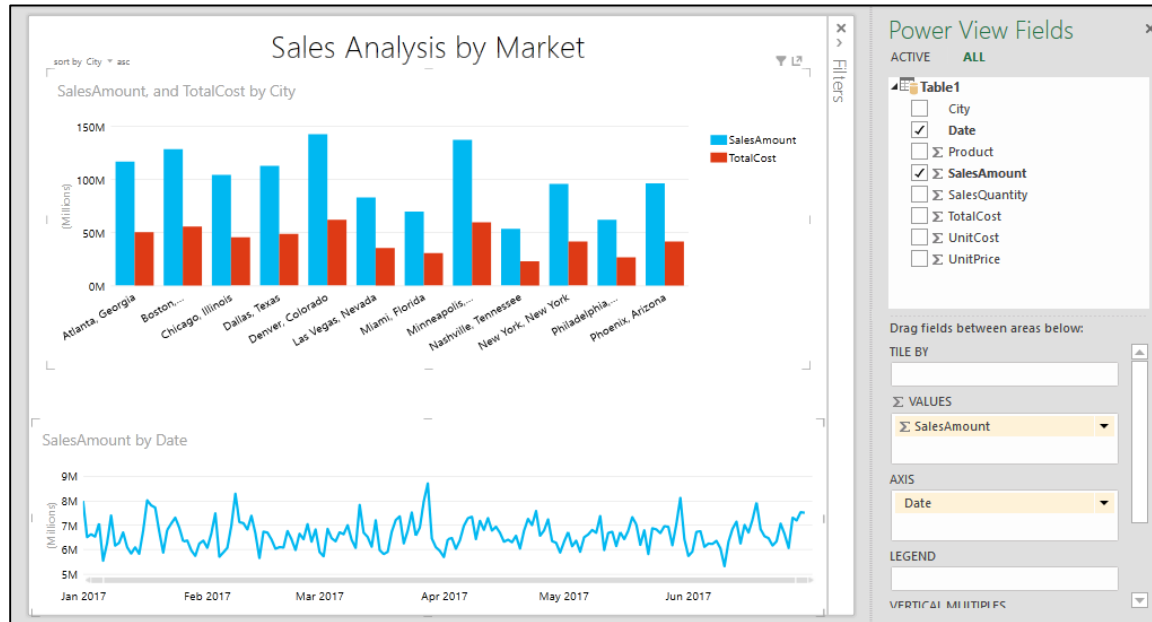


Figure 136 - Power View Worksheet with Multiple Elements

Filtering a Power View Dashboard

Once you create a Power View object such as a table, chart, or map, you can filter the data to specific items of interest. To do so, just click any item shown in the dashboard, and all elements in the dashboard instantly filter to that item. For example, clicking the “Nashville, Tennessee” element in the dashboard shown in **Figure 136** causes both the column chart and the line chart to filter results to only sales originating from the Nashville location as shown in **Figure 137**.

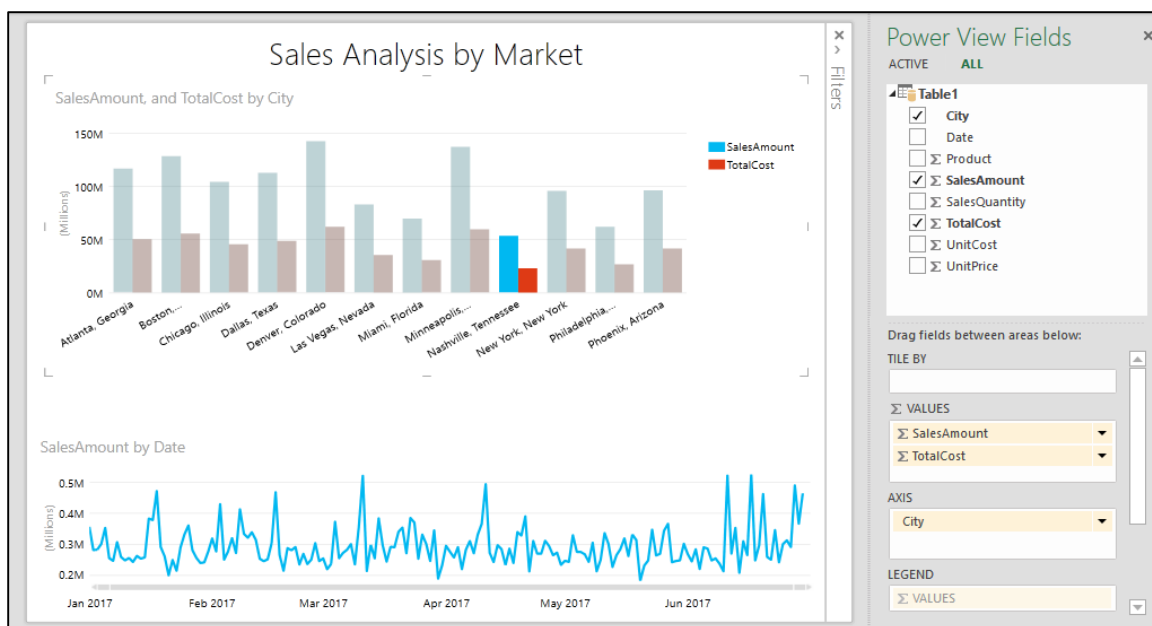


Figure 137 - A Power View Dashboard with a Filter Applied

Additionally, you can add **Slicer** filters to a Power View dashboard. To do so, drag any field to a blank area of the dashboard. Upon doing so, from the **Design** tab of the Ribbon, choose **Slicer**, and Power View converts the Table to a Slicer filter. **Figure 138** presents an illustration of a Power View dashboard with Slicer filters.

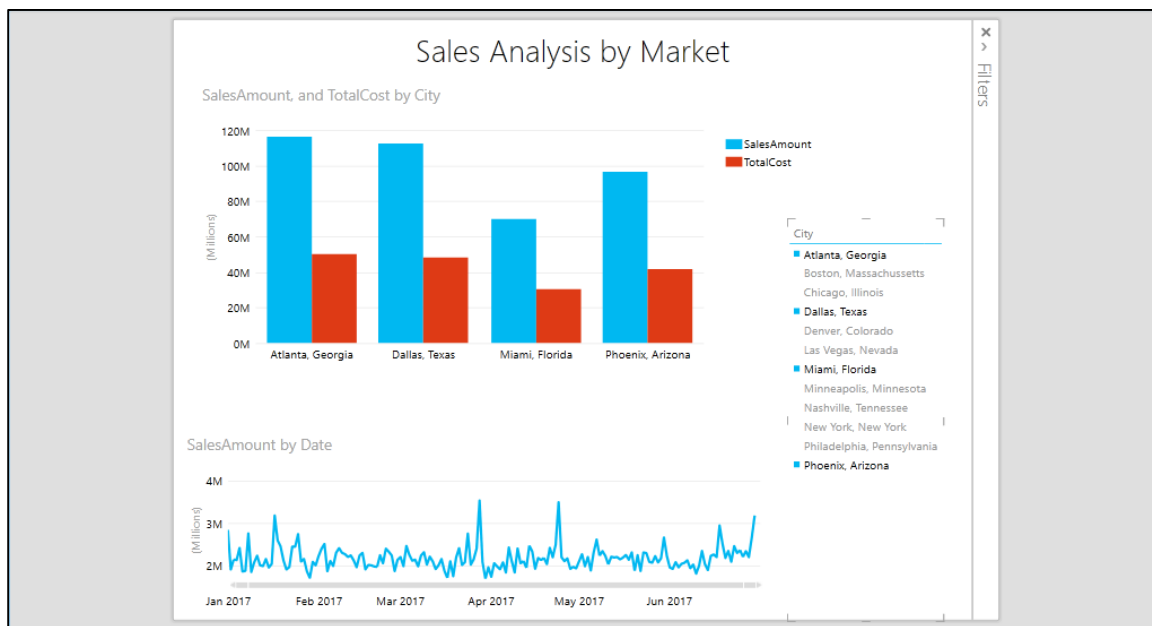


Figure 138 - A Power View Dashboard with Slicer Filters Applied

While far from an exhaustive discussion of all that you can accomplish using Power View, it should now be evident that this new tool provides a very powerful set of features from which

you can create engaging interactive reports from potentially very large data sets. Accordingly, for users of Excel 2013 and Excel 2016 who have access to Power View, building reports with this represents an Excel best practice.

Utilize Data Models in Conjunction with 3D Maps

3D Maps (formerly known as **Power Map**) is another tool that can use a data model as its data source. As implied by the name, 3D Maps plots data geospatially on an interactive map. You can plot data using a number of geospatial coordinates, including country, state/province, city, street address, and even latitude/longitude coordinates. Further, you can add time series elements to your data showing how the data changes over time.

To begin to uncover some of the power of working with 3D Maps, we will use a data model that has sales data in it, segregated by store, with address information available for each store in the data model. The purpose of our analysis of this data is to determine where our most valuable customers reside.

Click **3D Map** from the **Insert** tab of the Ribbon to begin working with the tool. Upon doing so, the window pictured in **Figure 139** appears.

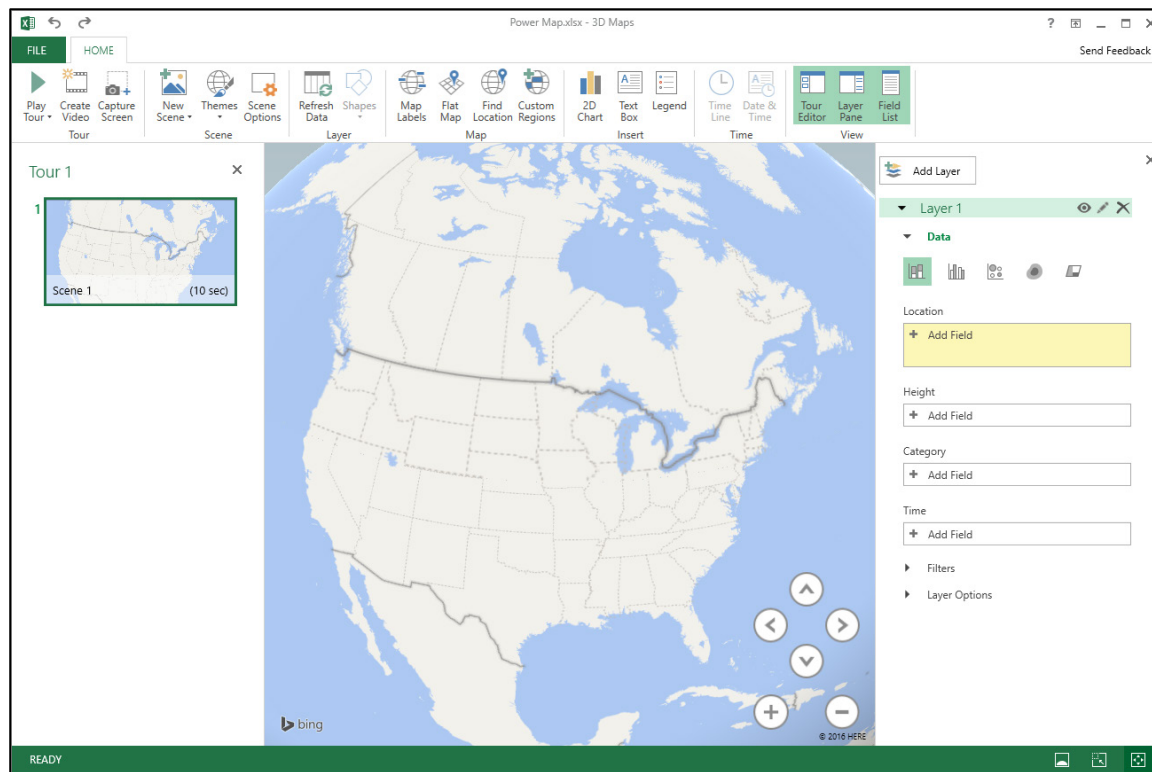


Figure 139 - Initial View of 3D Maps in Excel 2016

With the mapping tool now available, the first step is to add one or more location fields to the Location area in the task pane on the right side of the window. As shown in **Figure 140**, we are

adding *ZipCode* as a Location field in this example, and we have indicated to 3D Maps that *ZipCode* is a Postal Code.

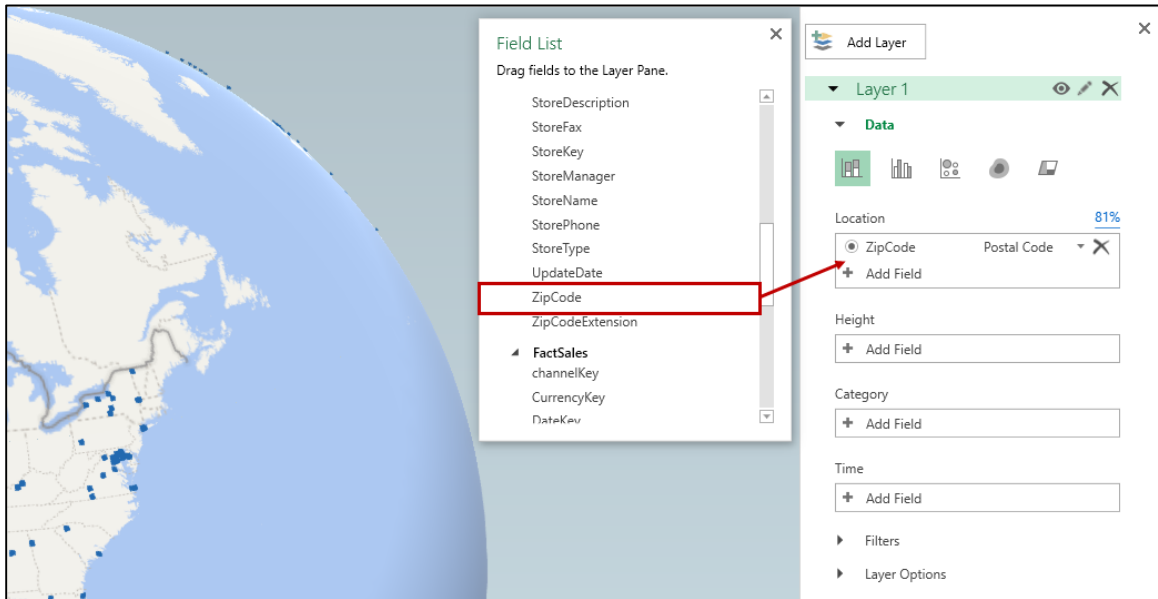


Figure 140 - Adding Location Fields to a 3D Map

Next, we add *SalesAmount* as a Height field, and 3D Maps immediately creates a column chart at the *ZipCode* level indicating relative sales per store. **Figure 141** provides a view of that map, indicating results for stores in the continental United States.

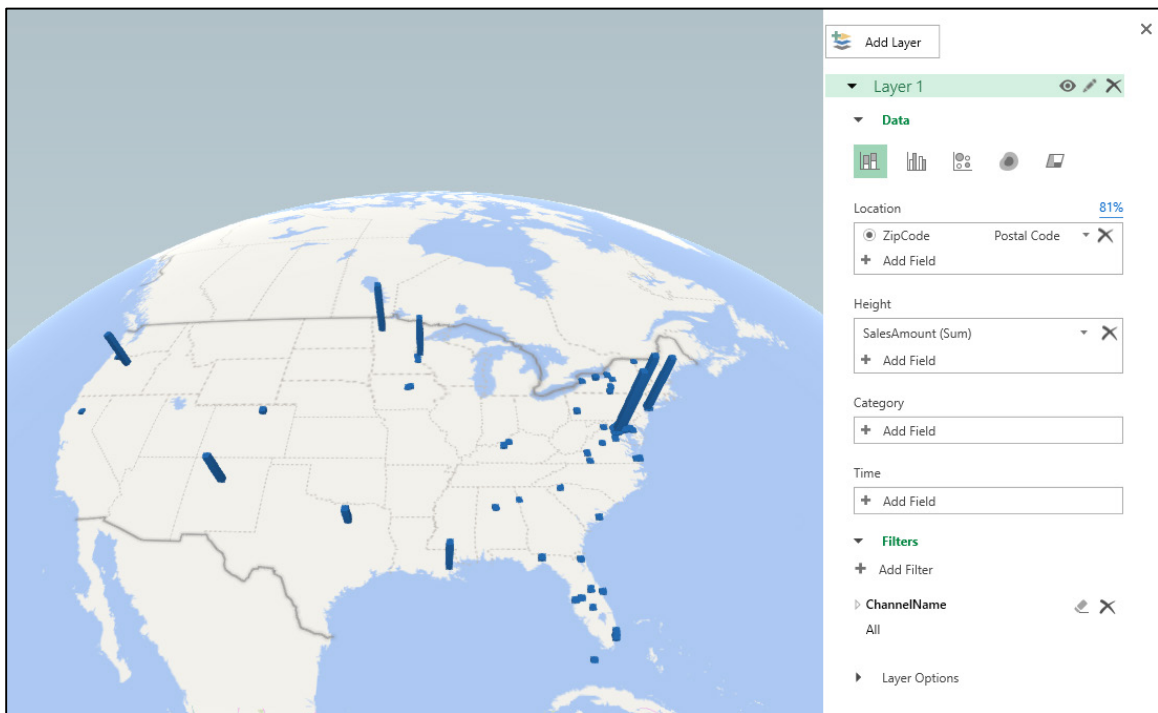


Figure 141 - Sales by Store Location in the Continental United States

We can continue by adding additional elements to the map, including additional items to plot – such as cost of goods sold – and filters – such as channel. Upon completion, our map might resemble that pictured in **Figure 142**.

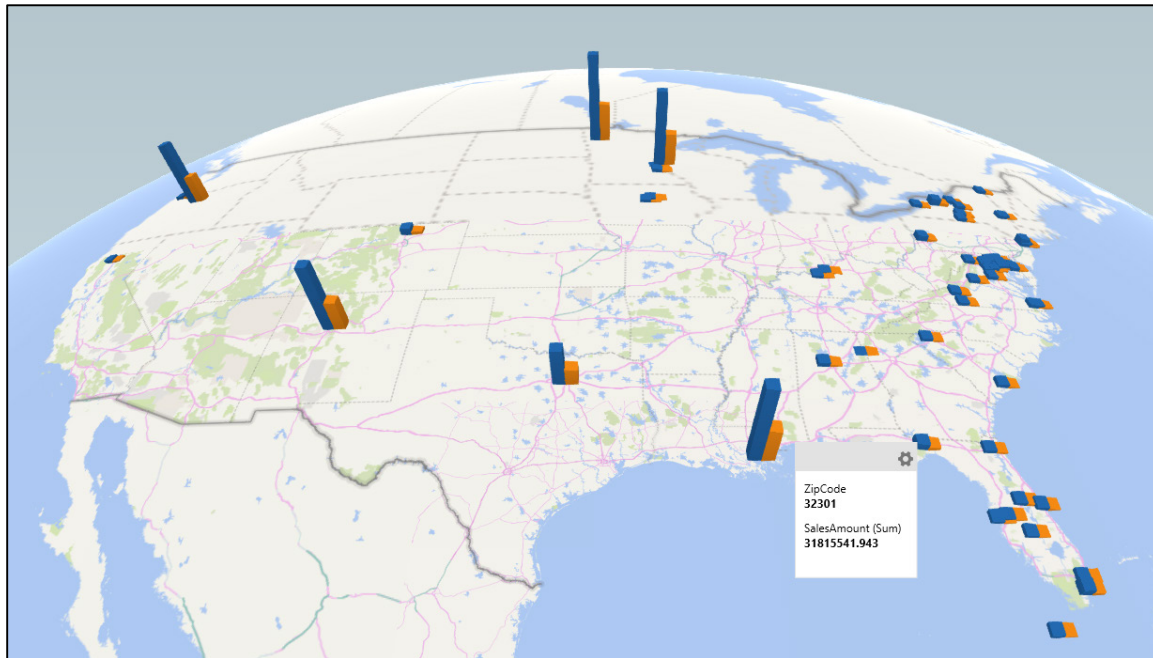


Figure 142 - Completed 3D Map, Including Cost of Goods Sold and Filtered to Display Results from the Store Channel Only

Data models can contain sophisticated relationships of data – potentially very large volumes of data. In many cases, summarizing this data via a PivotTable and/or PivotChart might be appropriate. In some cases, using Power View is advantageous because of the interactivity that tool provides to users. However, in cases where the data is geospatially-oriented, 3D Maps might be the best tools to help information consumers understand the true meaning of the data and the underlying business issues.

