Making Everything Easier!™

Microsoft® Excel® Power Pivot & Power Query

FOR DUMMIES®

A Wiley Brand

Learn to:

- **Extract data from databases and external files for use in Excel reporting**
- **Create powerful interactive reporting mechanisms and dashboards**
- **Integrate Power Query with Power Pivot to create truly robust Business Intelligence**

Michael Alexander
Author of Excel Macros For Dummies

Excel® Power Pivot & Power Query

FOR DUMMIES®

A Wiley Brand

Excel® Power Pivot & Power Query

FOR DUMMIES®

A Wiley Brand

by Michael Alexander

FOR DUMMIES®

A Wiley Brand

Excel® Power Pivot & Power Query For Dummies®

Published by: **John Wiley & Sons, Inc.,** 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

Library of Congress Control Number: 2016933854

ISBN 978-1-119-21064-1 (pbk); ISBN 978-1-119-21066-5 (ebk); ISBN 978-1-119-21065-8 (ebk)

Manufactured in the United States of America

10 9 8 7 6 5 4 3 2 1

Contents at a Glance

Table of Contents

Introduction

Over the past few years, the concept of self-service business intelligence (BI) has taken over the corporate world. Self-service BI is a form of business intelligence in which end users can independently generate their own reports, run their own queries, and conduct their own analyses, without the need to engage the IT department.

The demand for self-service BI is a direct result of several factors:

- **More power users:** Organizations are realizing that no single enterprise reporting system or BI tool can accommodate all of their users. Predefined reports and high-level dashboards may be sufficient for casual users, but a large portion of today's users are savvy enough to be considered power users. Power users have a greater understanding of data analysis and prefer to perform their own analysis, often within Excel.
- **Changing analytical needs:** In the past, business intelligence primarily consisted of IT-managed dashboards showing historic data on an agreed-upon set of key performance metrics. Managers now demand more dynamic predictive analysis, the ability to perform data discovery iteratively, and the freedom to take the hard left and right turns on data presentation. These managers often turn to Excel to provide the needed analytics and visualization tools.
- **Speed of BI:** Users are increasingly dissatisfied with the inability of IT to quickly deliver new reporting and metrics. Most traditional BI implementations fail specifically because the need for changes and answers to new questions overwhelmingly outpaces the IT department's ability to deliver them. As a result, users often find ways to work around the perceived IT bottleneck and ultimately build their own shadow BI (under the radar) solutions in Excel.

Recognizing the importance of the self-service BI revolution and the role Excel plays in it, Microsoft has made substantial investments in making Excel the cornerstone of its self-service BI offering. These investments have appeared starting with Excel 2007. Here are a few of note: the ability to handle over a million rows, tighter integration to SQL Server, pivot table slicers, and not least of all, the introduction of the Power Pivot and Power Query add-ins.

With the release of Excel 2016, Microsoft has aggressively moved to make Excel a player in the self-service BI arena by embedding both Power Pivot and Power Query directly into Excel.

For the first time, Excel is an integral part of the Microsoft BI stack. You can integrate multiple data sources, define relationships between data sources, process analysis services cubes, and develop interactive dashboards that can be shared on the web. Indeed, the new Microsoft BI tools blur the line between Excel analysis and what is traditionally IT enterprise-level data management and reporting capabilities.

With these new tools in the Excel wheelhouse, it's becoming important for business analysts to expand their skill sets to new territory, including database management, query design, data integration, multidimensional reporting, and a host of other skills. Excel analysts have to expand their skill set knowledge base from the one-dimensional spreadsheets to relational databases, data integration, and multidimensional reporting,

That's where this book comes in. Here, you're introduced to the mysterious world of Power Pivot and Power Query. You find out how to leverage the rich set of tools and reporting capabilities to save time, automate data clean-up, and substantially enhance your data analysis and reporting capabilities.

About This Book

The goal of this book is to give you a solid overview of the self-service BI functionality offered by Power Pivot and Power Query. Each chapter guides you through practical techniques that enable you to

- Extract data from databases and external files for use in Excel reporting
- Scrape and import data from the web
- Build automated processes to clean and transform data
- Easily slice data into various views on the fly, gaining visibility from different perspectives
- Analyze large amounts of data and report them in a meaningful way
- Create powerful, interactive reporting mechanisms and dashboards

Foolish Assumptions

Over the past few years, Microsoft has adopted an agile release cycle, allowing the company to release updates to Microsoft Office and the power BI tools practically monthly. This is great news for those who love seeing new features added to Power Pivot and Power Query. (It's not-so-great news if you're trying to document the features of these tools in a book.)

My assumption is that Microsoft will continue to add new bells and whistles to Power Pivot and Power Query at a rapid pace after publication of this book. So you may encounter new functionality not covered here.

The good news is that both Power Pivot and Power Query have stabilized and already have a broad feature set. So I'm also assuming that although changes will be made to these tools, they won't be so drastic as to turn this book into a doorstop. The core functionality covered in these chapters will remain relevant — even if the mechanics change a bit.

How This Book Is Organized

The chapters in this book are organized into three parts. Part I focuses on Power Pivot. Part II explores Power Query. Part III wraps up the book with the classic Part of Tens.

Part I: Supercharged Reporting with Power Pivot

Part I is all about getting you started with Power Pivot. Chapters 1 and 2 start you off with basic Power Query functionality and the fundamentals of data management. Chapter 3 provides an overview of pivot tables — the cornerstone of Microsoft BI analysis and presentation. In Chapters 4 and 5, you discover how to develop powerful reporting with external data and the Power Pivot data model. Chapter 6 focuses on creating and managing calculations and formulas in Power Pivot. Chapter 7 rounds out Part I with a look at publishing your Power Pivot reports.

Part II: Wrangling Data with Power Query

In Part II, you take an in-depth look at the functionality found in Power Query. Chapters 8 and 9 present the fundamentals of creating queries and connecting to various data sources, respectively. Chapter 10 shows you how you can leverage Power Query to automate and simply the steps for cleaning and transforming data. In Chapter 11, you see some options for making queries work together. Chapter 12 wraps up this look at Power Query with an exploration of custom functions and a description of how to leverage recorded steps to create your own amazing functions.

Part III: The Part of Tens

Part III is the classic Part of Tens section found in titles in the *For Dummies* series. The chapters in this part present ten or more pearls of wisdom, delivered in bite-size pieces. In Chapter 13, I share with you ten ways to improve the performance of your Power Pivot reports. Chapter 14 offers a rundown of ten tips for getting the most out of Power Query.

Icons Used In This Book

As you look in various places in this book, you see icons in the margins that indicate material of interest (or not, as the case may be). This section briefly describes each icon in this book.

Tips are beneficial because they help you save time or perform a task without having to do a lot of extra work. The tips in this book are time-saving techniques or pointers to resources that you should check out to get the maximum benefit from Excel.

Try to avoid doing anything marked with a Warning icon, which (as you might expect) represents a danger of one sort or another.

Whenever you see this icon, think *advanced* tip or technique. You might find these tidbits of useful information just too boring for words, or they could contain the solution you need to get a program running. Skip these bits of information whenever you like.

If you get nothing else out of a particular chapter or section, remember the material marked by this icon. This text usually contains an essential process or a bit of information you ought to remember.

Paragraphs marked with this icon reference the sample files for the book. If you want to follow along with the examples, you can download the sample files at `www.dummies.com/go/powerpivotpowerqueryfd`. The files are organized by chapter.

Beyond the Book

A lot of extra content that you won't find in this book is available at `www.dummies.com`. Go online to find the following:

- **Excel files used in the examples in this book can be found at**

  `www.dummies.com/go/excelpowerpivotpowerqueryfd`

- **Online articles covering additional topics are at**

  `www.dummies.com/extras/excelpowerpivotpowerquery`

  On this page, you can see how to integrate Power Pivot and Power Query to create a dynamic reporting duo. You can also uncover a list of resources to aid you in your Power BI journey.

- **The Cheat Sheet for this book is at**

  `www.dummies.com/cheatsheet/excelpowerpivotpowerquery`

  On this page, you find a list of useful Power Query functions that can be used to enhance the data clean-up and transformation process.

- **Updates to this book, if we have any, are also available at**

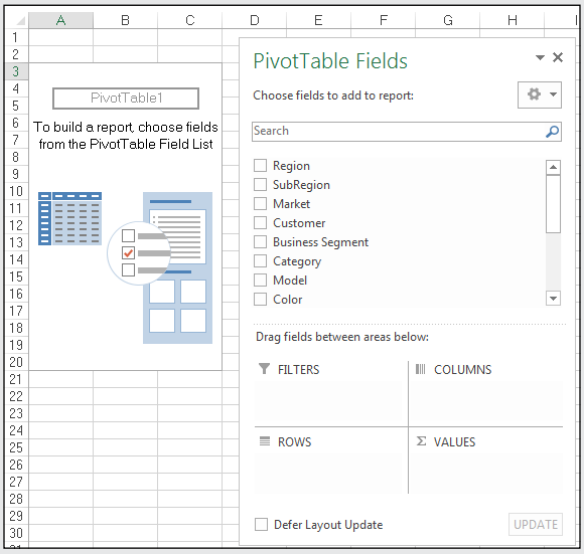
  `www.dummies.com/extras/excelpowerpivotpowerquery`

Where to Go from Here

It's time to start your self-service BI adventure! If you're primarily interested in Power Pivot, start with Chapter 1. If you want to dive right into Power Query, jump to Part II, which begins at Chapter 8.

Part I

Supercharged Reporting with Power Pivot



Go to www.dummies.com for great Dummies content online.

In this part . . .

- Discover how to think about data like a relational database.
- Get a solid understanding of the fundamentals of Power Pivot and pivot table reporting.
- Uncover the best practices for creating calculated columns and fields using Power Pivot formulas.
- Explore a few options for publishing your Power Pivot report.

Chapter 1

Thinking Like a Database

In This Chapter

- Examining traditional Excel limitations
- Keeping up with database terminology
- Looking into relationships

With the introduction of business intelligence (BI) tools such as Power Pivot and Power Query, it's becoming increasingly important for Excel analysts to understand core database principles. Unlike traditional Excel concepts, where the approach to developing solutions is relatively intuitive, you need to have a basic understanding of database terminology and architecture in order to get the most benefit from Power Pivot and Power Query. This chapter introduces you to a handful of fundamental concepts that you should know before taking on the rest of this book.

Exploring the Limits of Excel and How Databases Help

Years of consulting experience have brought this humble author face to face with managers, accountants, and analysts who all have had to accept this simple fact: Their analytical needs had outgrown Excel. They all faced fundamental challenges that stemmed from one or more of Excel's three problem areas: scalability, transparency of analytical processes, and separation of data and presentation.

Scalability

Scalability is the ability of an application to develop flexibly to meet growth and complexity requirements. In the context of this chapter, scalability

refers to Excel's ability to handle ever-increasing volumes of data. Most Excel aficionados are quick to point out that as of Excel 2007, you can place 1,048,576 rows of data into a single Excel worksheet — an overwhelming increase from the limitation of 65,536 rows imposed by previous versions of Excel. However, this increase in capacity does not solve all the scalability issues that inundate Excel.

Imagine that you're working in a small company and using Excel to analyze its daily transactions. As time goes on, you build a robust process complete with all the formulas, pivot tables, and macros you need in order to analyze the data that is stored in your neatly maintained worksheet.

As the amount of data grows, you will first notice performance issues. The spreadsheet will become slow to load and then slow to calculate. Why does this happen? It has to do with the way Excel handles memory. When an Excel file is loaded, the entire file is loaded into RAM. Excel does this to allow for quick data processing and access. The drawback to this behavior is that every time the data in your spreadsheet changes, Excel has to reload the entire document into RAM. The net result in a large spreadsheet is that it takes a great deal of RAM to process even the smallest change. Eventually, every action you take in the gigantic worksheet is preceded by an excruciating wait.

Your pivot tables will require bigger pivot caches, almost doubling the Excel workbook's file size. Eventually, the workbook will become too big to distribute easily. You may even consider breaking down the workbook into smaller workbooks (possibly one for each region). This causes you to duplicate your work.

In time, you may eventually reach the 1,048,576-row limit of the worksheet. What happens then? Do you start a new worksheet? How do you analyze two datasets on two different worksheets as one entity? Are your formulas still good? Will you have to write new macros?

These are all issues that need to be addressed.

Of course, you will also encounter the Excel power customers, who will find various clever ways to work around these limitations. In the end, though, these methods will always be simply workarounds. Eventually, even these power-customers will begin to think less about the most effective way to perform and present analysis of their data and more about how to make data "fit" into Excel without breaking their formulas and functions. Excel is flexible enough that a proficient customer can make most things fit just fine. However, when customers think only in terms of Excel, they're undoubtedly limiting themselves, albeit in an incredibly functional way.

In addition, these capacity limitations often force Excel customers to have the data prepared for them. That is, someone else extracts large chunks of data from a large database and then aggregates and shapes the data for use in Excel. Should the serious analyst always be dependent on someone else for her data needs? What if an analyst could be given the tools to access vast quantities of data without being reliant on others to provide data? Could that analyst be more valuable to the organization? Could that analyst focus on the accuracy of the analysis and the quality of the presentation instead of routing Excel data maintenance?

A relational database system (such as Access or SQL Server) is a logical next step for the analyst who faces an ever-increasing data pool. Database systems don't usually have performance implications with large amounts of stored data, and are built to address large volumes of data. An analyst can then handle larger datasets without requiring the data to be summarized or prepared to fit into Excel. Also, if a process ever becomes more crucial to the organization and needs to be tracked in a more enterprise-acceptable environment, it will be easier to upgrade and scale up if that process is already in a relational database system.

Transparency of analytical processes

One of Excel's most attractive features is its flexibility. Each individual cell can contain text, a number, a formula, or practically anything else the customer defines. Indeed, this is one of the fundamental reasons that Excel is an effective tool for data analysis. Customers can use named ranges, formulas, and macros to create an intricate system of interlocking calculations, linked cells, and formatted summaries that work together to create a final analysis.

So what is the problem? The problem is that there is no transparency of analytical processes. It is extremely difficult to determine what is actually going on in a spreadsheet. Anyone who has had to work with a spreadsheet created by someone else knows all too well the frustration that comes with deciphering the various gyrations of calculations and links being used to perform analysis. Small spreadsheets that are performing modest analysis are painful to decipher, and large, elaborate, multi-worksheet workbooks are virtually impossible to decode, often leaving you to start from scratch.

Compared to Excel, database systems might seem rigid, strict, and unwavering in their rules. However, all this rigidity comes with a benefit.

Because only certain actions are allowable, you can more easily come to understand what is being done within structured database objects such as queries or stored procedures. If a dataset is being edited, a number is being calculated, or any portion of the dataset is being affected as part of an

analytical process, you can readily see that action by reviewing the query syntax or the stored procedure code. Indeed, in a relational database system, you never encounter hidden formulas, hidden cells, or dead named ranges.

Separation of data and presentation

Data should be separate from presentation; you don't want the data to become too tied into any particular way of presenting it. For example, when you receive an invoice from a company, you don't assume that the financial data on that invoice is the true source of your data. It is *a presentation of* your data. It can be presented to you in other manners and styles on charts or on websites, but such representations are never the actual source of the data.

What exactly does this concept have to do with Excel? People who perform data analysis with Excel tend, more often than not, to fuse the data, the analysis, and the presentation. For example, you often see an Excel workbook that has 12 worksheets, each representing a month. On each worksheet, data for that month is listed along with formulas, pivot tables, and summaries. What happens when you're asked to provide a summary by quarter? Do you add more formulas and worksheets to consolidate the data on each of the month worksheets? The fundamental problem in this scenario is that the worksheets actually represent data values that are fused into the presentation of the analysis.

The point being made here is that data should not be tied to a particular presentation, no matter how apparently logical or useful it may be. However, in Excel, it happens all the time.

In addition, as discussed earlier in this chapter, because all manners and phases of analysis can be done directly within a spreadsheet, Excel cannot effectively provide adequate transparency to the analysis. Each cell has the potential to hold formulas, be hidden, and contain links to other cells. In Excel, this blurs the line between analysis and data, which makes it difficult to determine exactly what is going on in a spreadsheet. Moreover, it takes a great deal of effort in the way of manual maintenance to ensure that edits and unforeseen changes don't affect previous analyses.

Relational database systems inherently separate analytical components into tables, queries, and reports. By separating these elements, databases make data less sensitive to changes and create a data analysis environment in which you can easily respond to new requests for analysis without destroying previous analyses.

You may find that you manipulate Excel's functionalities to approximate this database behavior. If so, you must consider that if you're using Excel's functionality to make it behave like a database application, perhaps the real thing just might have something to offer. Utilizing databases for data storage and analytical needs would enhance overall data analysis and would allow Excel power-customers to focus on the presentation in their spreadsheets.

In these days of big data, customers demand more, not less, complex data analysis. Excel analysts will need to add tools to their repertoires to avoid being simply "spreadsheet mechanics." Excel can be stretched to do just about anything, but maintaining such creative solutions can be a tedious manual task. You can be sure that the sexy aspect of data analysis does not lie in the routine data management within Excel; rather, it lies in leveraging BI Tools such as providing clients with the best solution for any situation.

Getting to Know Database Terminology

The terms *database, table, record, field,* and *value* indicate a hierarchy from largest to smallest. These same terms are used with virtually all database systems, so you should learn them well.

Databases

Generally, the word *database* is a computer term for a collection of information concerning a certain topic or business application. A database helps you organize this related information in a logical fashion for easy access and retrieval. Certain older database systems used the term *database* to describe individual tables. The current use of *database* applies to all elements of a database system.

Databases aren't only for computers. Manual databases are sometimes referred to as manual filing systems or manual database systems. These filing systems usually consist of people, papers, folders, and filing cabinets — paper is the key to a manual database system. In a real-life manual database system, you probably have in-baskets and out-baskets and some type of formal filing method. You access information manually by opening a file cabinet, removing a file folder, and finding the correct piece of paper. Customers fill out paper forms for input, perhaps by using a keyboard to input information that is printed on forms. You find information by manually sorting the papers or by copying information from many papers to another piece of paper (or even into an Excel spreadsheet). You may use a spreadsheet or calculator to analyze the data or display it in new and interesting ways.

Tables

A database stores information in a carefully defined structure known as a table. A *table* is just a container for raw information (called *data*), similar to a folder in a manual filing system. Each table in a database contains information about a single entity, such as a person or product, and the data in the table is organized into rows and columns. A relational database system stores data in related tables. For example, a table containing employee data (names and addresses) may be related to a table containing payroll information (pay date, pay amount, and check number).

To use database wording, a table is an object. As you design and work with databases, it's important to see each table as a unique entity and to see how each table relates to the other objects in the database.

In most database systems, you can view the contents of a table in a spreadsheet-like form called a *datasheet,* composed of rows and columns (known as *records* and *fields,* respectively — see the following section). Although a datasheet and a spreadsheet are superficially similar, a datasheet is quite a different type of object. You typically cannot make changes or add calculations directly within a table. Your interaction with tables will primarily come in the form of queries or views — see the later section "Queries").

Records, fields, and values

A database table is divided into rows (called *records*) and columns (called *fields*), with the first row (the heading on top of each column) containing the names of the fields in the database.

Each row is a single record containing fields that are related to that record. In a manual system, the rows are individual forms (sheets of paper), and the fields are equivalent to the blank areas on a printed form that you fill in.

Each column is a field that includes many properties specifying the type of data contained within the field and how the database should handle the field's data. These properties include the name of the field (Company) and the type of data in the field (Text). A field may include other properties as well. For example, the Address field's Size property tells the database the maximum number of characters allowed for the address.

At the intersection of a record and a field is a *value* — the actual data element. For example, in a field named Company, a company name entered into that field would represent one data value.

When working with Microsoft Access, the term *field* is used to refer to an attribute stored in a record. In many other database systems, including SQL Server, *column* is the expression you hear most often in place of *field* — field and column mean the same thing. The exact terminology that's used relies somewhat on the context of the database system underlying the table containing the record.

Queries

Most relational database systems allow the creation of queries (sometimes called views). A query extracts information from the tables in the database; a query selects and defines a group of records that fulfill a certain condition. Most database outputs are based on queries that combine, filter, or sort data before it's displayed. Queries are often called from other database objects, such as stored procedures, macros, or code modules. In addition to extracting data from tables, queries can be used to change, add, or delete database records.

An example of a query is when a person at the sales office tells the database, "Show me all customers, in alphabetical order by name, who are located in Massachusetts and who made a purchase over the past six months." Or "Show me all customers who bought Chevrolet car models within the past six months, and display them sorted by customer name and then by sale date."

Rather than ask the question using English words, a person uses a special syntax, such as Structured Query Language (or SQL), to communicate to the database what the query will need to do.

Understanding Relationships

After you understand the basic terminology of databases, it's time to focus on one of their more useful features: A *relationship* is the mechanism by which separate tables are related to each other. You can think of a relationship as a VLOOKUP, in which you relate the data in one data range to the data in another data range using an index or a unique identifier. In databases, relationships do the same thing, but without the hassle of writing formulas.

Relationships are important because most of the data you work with fits into a multidimensional hierarchy of sorts. For example, you may have a table showing customers who buy products. These customers require invoices that have invoice numbers. Those invoices have multiple lines of transactions listing what they bought. A hierarchy exists there.

Now, in the one-dimensional spreadsheet world, this data typically would be stored in a flat table, like the one shown in Figure 1-1.

	A	B	C	D	E	F
1	CustomerID	CustomerName	InvoiceNumber	InvoiceDate	Quantity	UnitPrice
2	BAKERSEM0001	Baker's Emporium Inc.	ORDST1025	5/8/2005	1	19.95
3	BAKERSEM0001	Baker's Emporium Inc.	ORDST1025	5/8/2005	5	1759.95
4	BAKERSEM0001	Baker's Emporium Inc.	ORDST1025	5/8/2005	4	9.95
5	BAKERSEM0001	Baker's Emporium Inc.	STDINV2251	4/12/2007	4	9.95
6	AARONFIT0001	Aaron Fitz Electrical	ORDST1026	5/8/2005	5	9.95
7	AARONFIT0001	Aaron Fitz Electrical	ORDST1026	5/8/2005	3	1759.95
8	AARONFIT0001	Aaron Fitz Electrical	ORDST1026	5/8/2005	2	79.95
9	AARONFIT0001	Aaron Fitz Electrical	STDINV2252	4/12/2007	3	1759.95
10	AARONFIT0001	Aaron Fitz Electrical	STDINV2252	4/12/2007	5	9.95
11	METROPOL0001	Metropolitan Fiber Systems	ORD1002	5/7/2004	1	9.95
12	AARONFIT0001	Aaron Fitz Electrical	INV1024	2/10/2004	1	119.95
13	AARONFIT0001	Aaron Fitz Electrical	INV1025	2/15/2004	1	109.95
14	LECLERC0001	LeClerc & Associates	ORDPH1005	5/10/2004	2	189.95
15	MAGNIFIC0001	Magnificent Office Images	ORD1000	5/8/2004	1	359.95
16	HOLLINGC0001	Holling Communications Inc.	ORD1001	5/10/2004	2	59.95
17	MAHLERST0001	Mahler State University	ORDST1008	5/10/2004	1	5999.95

Figure 1-1: Data is stored in an Excel spreadsheet using a flat-table format.

Because customers have more than one invoice, the customer information (in this example, CustomerID and CustomerName) has to be repeated. This causes a problem when that data needs to be updated.

For example, imagine that the name of the company Aaron Fitz Electrical changes to Fitz and Sons Electrical. Looking at Figure 1-1, you see that multiple rows contain the old name. You would have to ensure that every row containing the old company name is updated to reflect the change. Any rows you miss will not correctly map back to the right customer.

Wouldn't it be more logical and efficient to record the name and information of the customer only one time? Then, rather than have to write the same customer information repeatedly, you could simply have some form of customer reference number.

This is the idea behind relationships. You can separate customers from invoices, placing each in their own tables. Then you can use a unique identifier (such as CustomerID) to relate them together.

Figure 1-2 illustrates how this data would look in a relational database. The data would be split into three separate tables: Customers, InvoiceHeader, and InvoiceDetails. Each table would then be related using unique identifiers (CustomerID and InvoiceNumber, in this case).