Mastering Excel
Power Pivot

United Kingdom Totals

Mark Moore

Mastering Excel

PowerPivot

Mark Moore

Introduction

Welcome to another Mastering Excel lesson. If you have previous lessons, thanks for sticking around. If you are new, I hope you enjoy the lesson. The lessons are easy going, relaxed, no-nonsense and easy to understand. I try my best to explain complex topics in a simple and entertaining way. My goal is that you will finish reading each lesson and have immediately applicable skills you can use at work or home.

This lesson will focus on a new Excel feature: PowerPivot. Excel is quickly becoming a very powerful data management tool. Excel 2003 was ok in that it could store up to 65,536 rows of data and 256 columns. Then Excel 2007 came along and now it can store up to 1,048,576 rows and 16,000 columns. That's a lot of data but truthfully with technology generating so much data, that's not big enough. Many companies generate much more data than can be stored in Excel.

PowerPivot solves that issue. PowerPivot is a mini application that lives inside Excel and can manage millions or tens of millions of records. You can create relationships between different tables and build custom calculations before putting anything on a worksheet.

You can't see anything…

I want to set expectations for this lesson. If you are going to use Excel to its maximum potential and manage large (very large) amounts of data, this is the lesson for you. PowerPivot is how to set up the data so it can be used by other very cool Excel features that will elegantly display the data (Power Map and Power View). PowerPivot is the workhorse. The issue is that you really won't 'see' anything in this lesson. I mean, of course you will see PowerPivot and what it does. What I mean is that you won't see the amazing graphs because those are handled by Power Map or Power View. In this lesson you are working behind the scenes getting everything ready so the other two programs can use the data.

Even though the data sets you will be working with here are small, think about managing several millions of rows.

If you want to work along the exercises in this lesson (I strongly recommend this) please go to my website and download the follow-along workbook. My website is:

A bit of clarification on how to get the follow-along workbooks: You will input your name and email address. You will receive a confirmation email. Once you confirm, you will receive a second email with the follow-along workbook.

Why do I do this?

I can't package an Excel file with an eBook. Amazon will not allow it. Also, the *only* thing I do with your email is send you the workbook and periodically send you updates about new lessons that I am working on.

What is PowerPivot?

Microsoft is taking Excel in a new direction. Microsoft has included in Excel some very powerful tools to help users analyze large volumes of data. This analysis is usually called Business Intelligence (BI). Previously, processing and working with these large data sets was limited to experienced technical professionals that used expensive software packages. Microsoft wants to do away with that. They want to bring BI to the average user. They use the term 'Self Service Business Intelligence' to indicate that users can now perform this analysis without calling in their IT departments or external consultants.

Microsoft's toolset is called the Excel BI toolkit. It has several different tools:

- **Power Query** - Used to import data
- **PowerPivot** - Used to analyze data
- **Power View** - Used to create presentations
- **Power Map** - Used to create geographical presentations

This lesson will cover PowerPivot.

PowerPivot

There's a new 'thing' in Excel you may have heard of called a Data Model. PowerPivot is closely intertwined with this Data Model. 'Data Model'. Sounds intimidating right? Many IT folks like to impress non-IT people with their terms (and Data Model is one of them) but don't worry. It's not that complicated.

A data model is just several tables that have predefined relationships. That is, one column of one table matches another column of another table. Those columns can be set up as a relationship. When you do that, you are working with a relational data model. In pure Excel terms, if you write a VLOOKUP formula that pulls a match from a second worksheet, you just built a relational model.

In my previous lessons I talk about building your Excel models in layers. The data layer, the business logic layer and the presentation layer. The data layer is represented as a worksheet with all your data records. That's okay but now, PowerPivot really *is* a true data layer. Not only does it have the records of interest, but you will be able to build relationships between multiple tables and add calculations.

This is a true data layer.

Installing PowerPivot

Depending on which version of Excel you have, you will need to follow different steps.
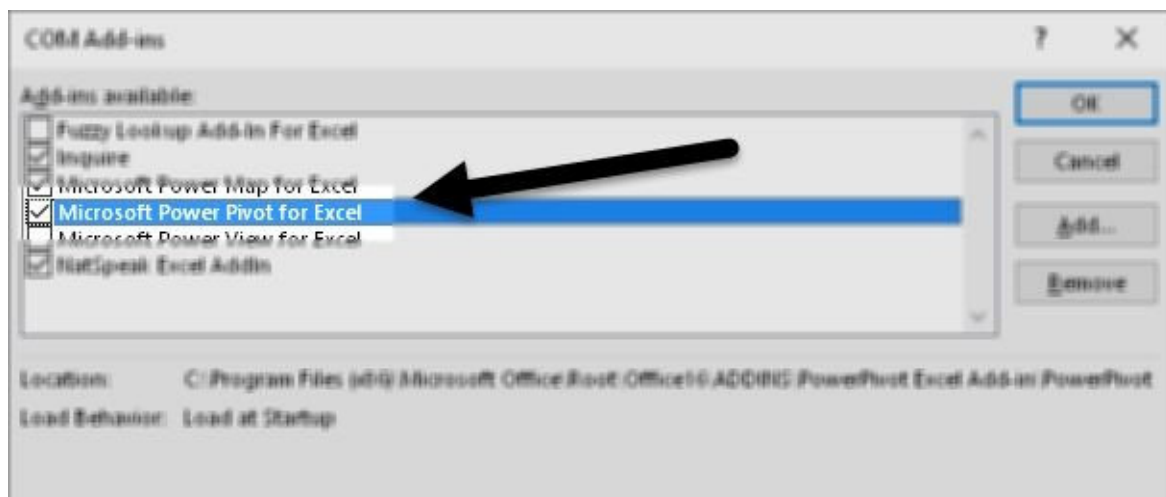
Excel 2013/Excel 2016/Office 265 Office Pro Plus

PowerPivot is installed, just not activated. To activate PowerPivot:

1. Click on File.
2. Click on Options.
3. Click on Add-Ins.
4. In the drop down list at the bottom of the window, select COM Add-Ins.
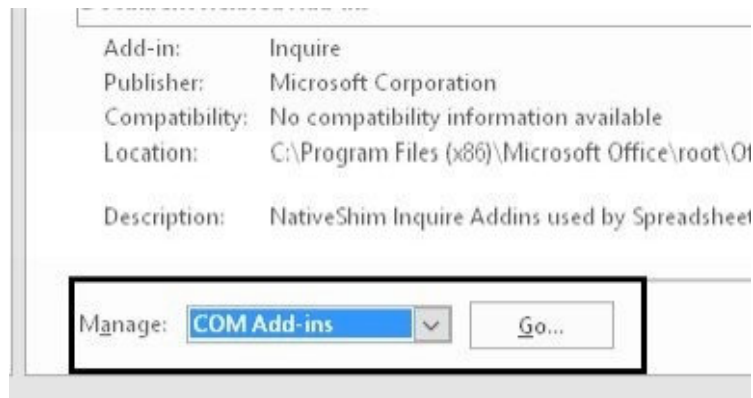
5. Click Go.
6. Select Microsoft PowerPivot for Excel.

7. Click OK.

You will now have a new PowerPivot tab in Excel.

Excel 2010

PowerPivot exists for Excel 2010. The installation is not difficult but it takes a few steps. You need to install some programs before installing PowerPivot.

1. Install .NET Framework 4.0 and Visual Studio 2010 Tools for Office Runtime.
2. If you are running Windows Vista or Windows Server 2008, install the platform update.
3. Install Microsoft® Office 2010.
4. Download PowerPivot for Microsoft Excel. The current link on the Microsoft website is: https://www.microsoft.com/en-us/download/details.aspx?id=29074

Important: If you install the 32-bit version of Excel, you must use the 32-bit version of PowerPivot. If you install the 64-bit version of Excel, you must use the 64-bit version of PowerPivot.

Note: When you start Microsoft Office 2010 Excel for the first time, you will be prompted for permission for Excel to load SQL Server 2012 PowerPivot for Microsoft Excel 2010.

5. After you install .NET Framework 4.0 and Visual Studio 2010 Tools for Office Runtime, click the Download button in the webpage.
6. Double-click the 32-bit or 64-bit msi file that was downloaded in step 4.

Note that if you are using PowerPivot for Excel 2010, the images in your version will be different than the images in this lesson.

Connecting to Data Sources

There are several data sources PowerPivot can connect to. These include Excel, MS Access, other corporate relational databases, text files and Power Query.
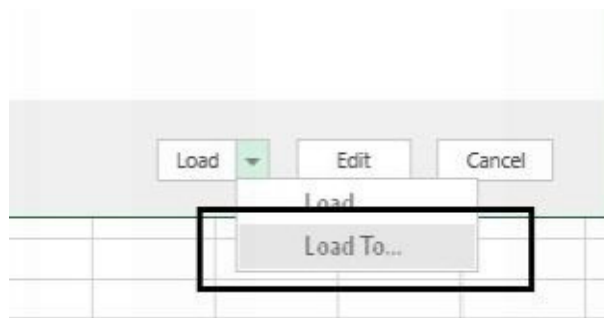
Power Query

This is the simplest data source for PowerPivot. You used Power Query to connect to the data source and prepared the data. Instead of loading the data into an Excel spreadsheet, you loaded the data into the Excel Data model. With the data in the Data Model, PowerPivot can connect directly to the data model and begin working with it.
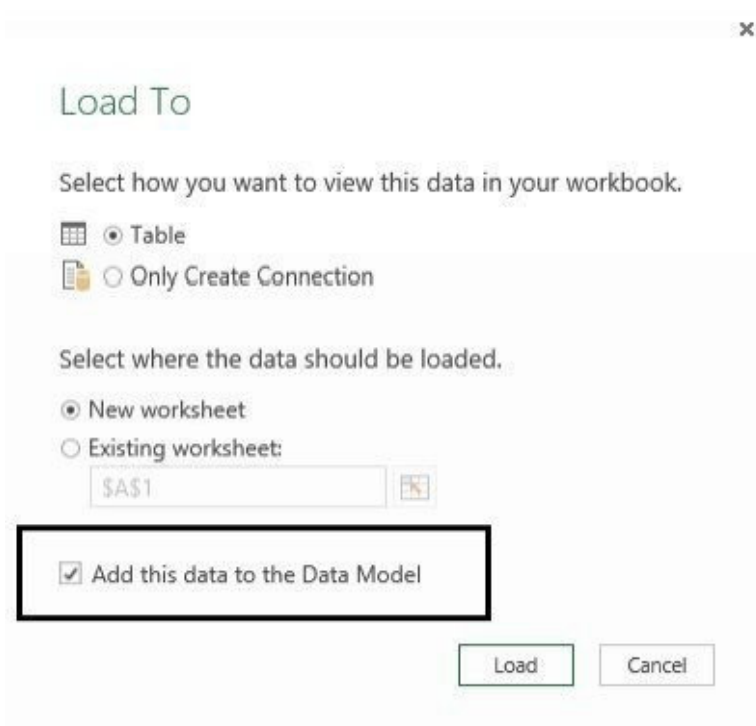
In general, the data flow would be something like this:

Data Source > Power Query > PowerPivot > Pivot Table or Power Map or Power View

When in Power Query,

Choose 'Load To…' in the Import Wizard.
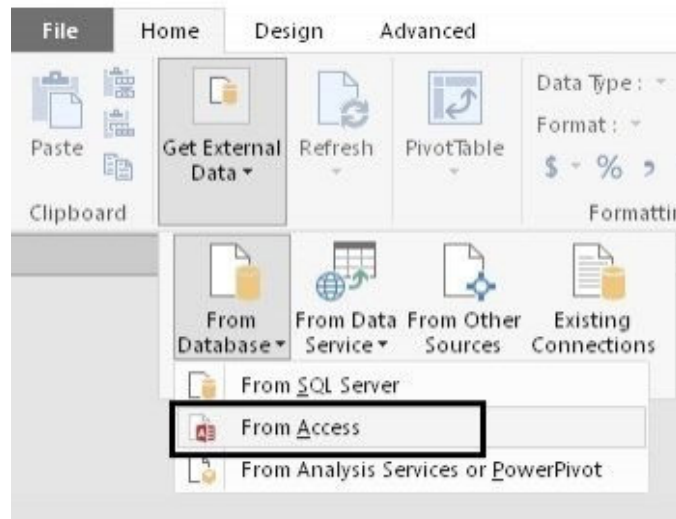

Check the box to load data into the Data Model.

Relational Databases

This is the data source you will almost always use (either directly or through Power Query). PowerPivot is excellent at handling millions of rows. Although you could load .CSV files with millions of rows, more often than not, you will be connecting to a relational database.

I'm not going to show you how to download and install a relational database engine; that's out of scope for an Excel lesson. However, if doing this interests you, search online for SQL Server Express. This is a free Microsoft database engine that you can download and install on your PC. There are several tutorials that explain step-by-step how to install and configure it.

This is how you connect to a relational database; in this case, I am connecting to MS Access.

1. Click on Manage in the PowerPivot ribbon.
2. In the new PowerPivot window, click Get External Data, From Database, From Access.

3. Fill out the requirements for the data source. I have my sample database stored in C:\Common.

4. Click Next.

In this next window, you can choose to use a wizard to select a table to import or write native SQL to retrieve data. If your IT department gave you a SQL statement, you can select the 'Write a query…' option and paste in the SQL there.

5. For this exercise, choose the 'Select from a list of tables and views to choose the data to import'.

Table Import Wizard   ?   ✕

Choose How to Import the Data
    You can either import all of the data from tables or views that you specify, or you can
    write a query using SQL that specifies the data to import.

    ⦿ Select from a list of tables and views to choose the data to import

    ○ Write a query that will specify the data to import
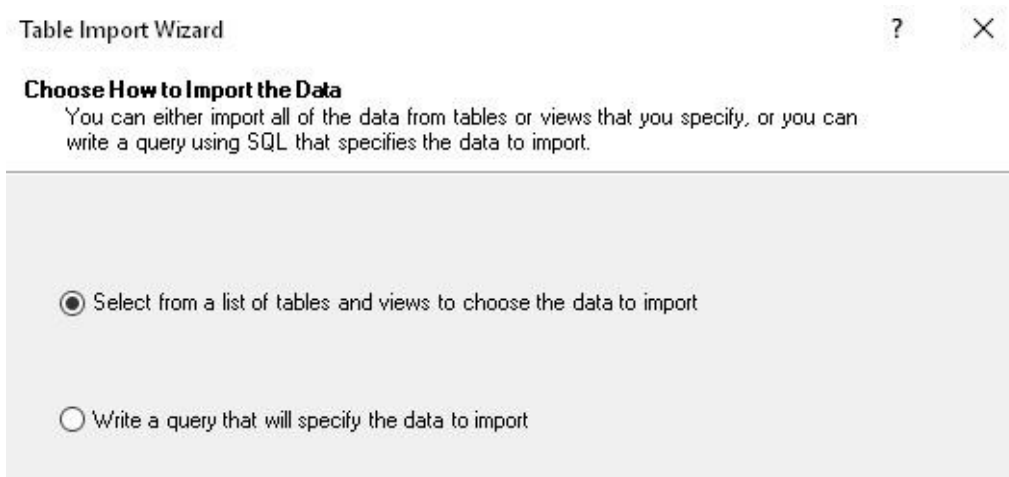
This next step is optional but worthwhile to discuss. The sample database has one table: Location. This table should be selected and you will be importing the entire table. However, realistically you are not going to import entire tables into your model. For your quarterly analysis, do you really want 20 years of historical data? Probably not. PowerPivot gives you the ability to filter out rows from being imported into the data model.

6. Click on Preview & Filter.
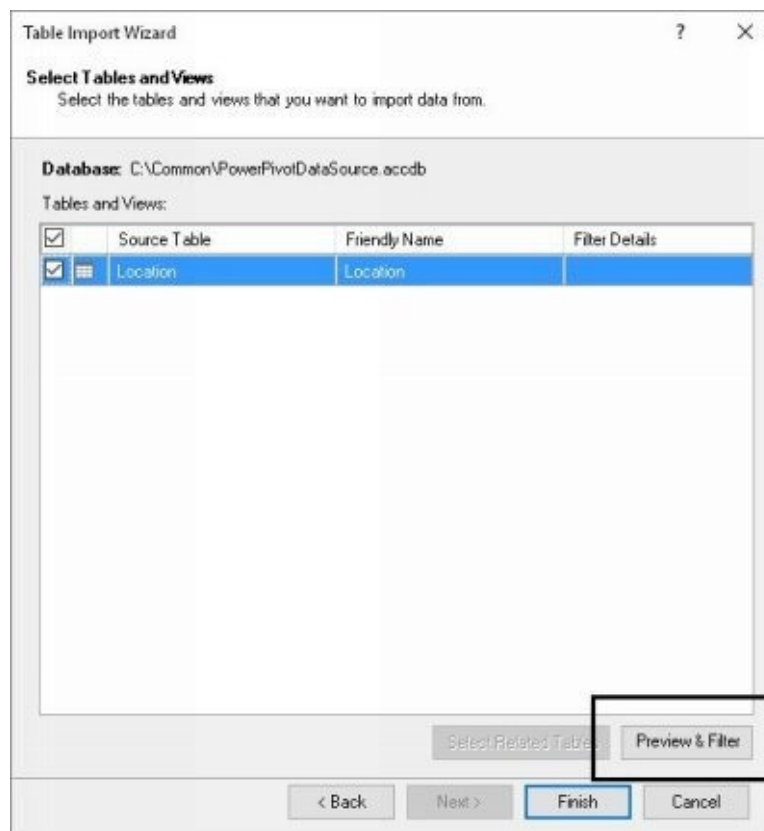
Here you get a preview of the table that is to be imported. You can uncheck a column to

skip importing it or you can click the small drop down in one column. The small drop down displays a field icon pane (almost identical to the Pivot Table pane) where you can choose criteria to skip rows from importing.
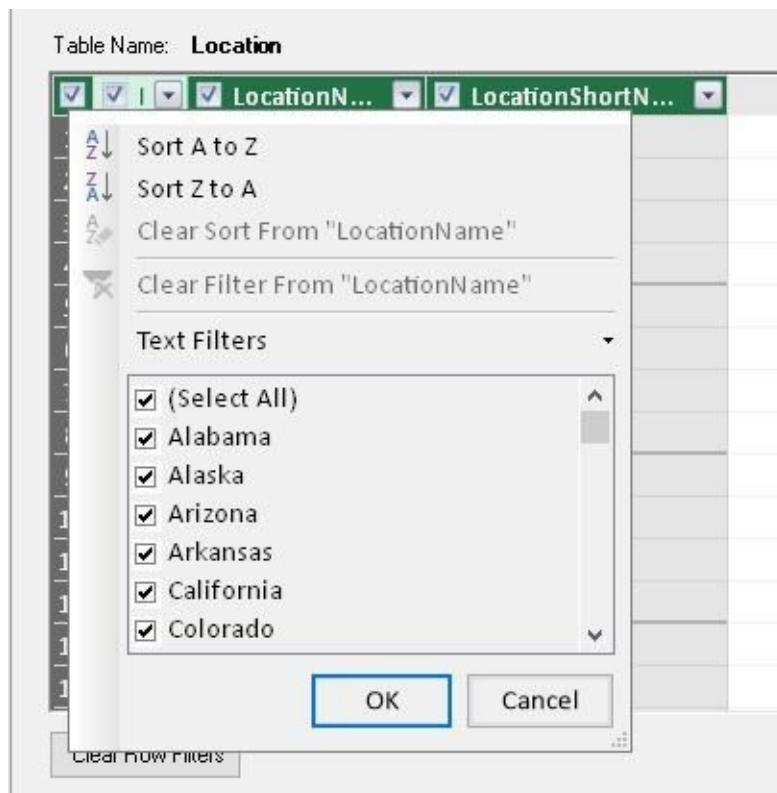
I just said a few pages before that PowerPivot can import millions of rows and is super powerful. Now I'm telling you that you don't have to import all rows. What gives?

PowerPivot is powerful but it is still a desktop solution. That means PowerPivot is limited by your PC's processing capacity and your user's processing capacity. PowerPivot is much more powerful than native Excel, but it will still slow down trying to process and calculate millions upon millions of rows. Don't import everything, just because you can. Import what you need. Later on, if you find out you need more data, alter the import connection to import more rows. When dealing with huge amounts of data, you need to pay attention to efficiency.

7. Click Cancel to exit this pane.
8. Click Finish to import the table into the data model.

The Table Import Wizard will display the records imported and if there were any errors.
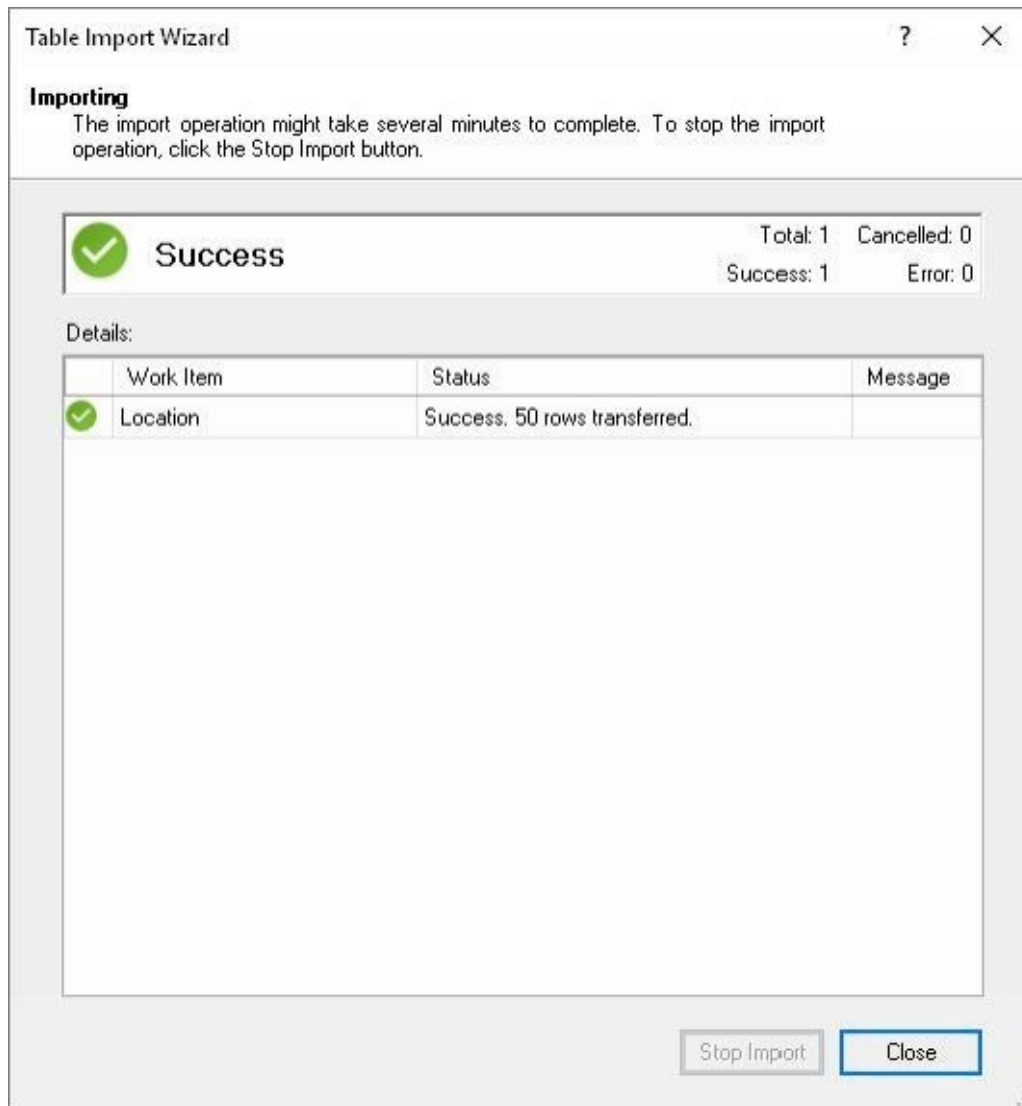
Power Query - PowerPivot doesn't have many options to fix import errors. It just gets the data. This is why Power Query is so useful. Power Query **can** apply several sophisticated treatments to the data to clean it up and then get it ready for PowerPivot.

9. Click Close.

Now it looks like the data is in Excel but it is not. Notice that you are still in the PowerPivot interface. Each table will appear in its own tab (Much like Excel. Yeah, I know it's confusing). Remember the big picture, you will be importing several tables into PowerPivot (the interface you are looking at now) and then creating relationships (and calculations) between the tables.

Importing From Excel

There are two ways to import data into Excel: as a Static Table and as a Linked Table.

Static Table

Sometimes, you just need to get the data into the data model. If the data doesn't change (suppose it is a list of sales regions), then this is the easiest way to load data. Basically you just copy the data from Excel and paste it directly into PowerPivot.

Note that when you do all the following exercises, you can leave the PowerPivot window up, click back to Excel and do normal Excel stuff.

1. Open the follow-along workbook Regions.xlsx.

2. Highlight cells A1:A5 (Don't highlight A6 for now).
3. Right-click and select Copy.
4. Click back to PowerPivot.
5. On the Home tab, click on 'Paste'.
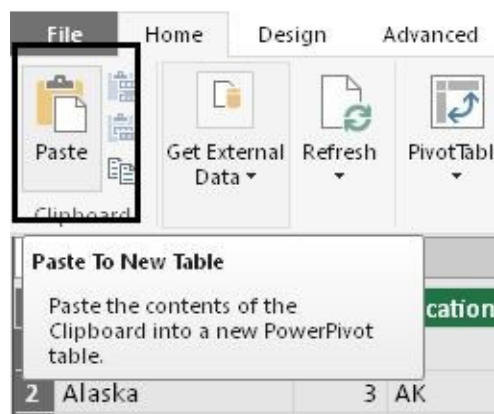
6. Type in the new table name. You can name it Regions.

Also notice that PowerPivot gives you the option to use the first column copied as a column header.

Paste Preview   ?   ×

Preview the data that you are about to paste. You can modify the table name and specify whether to use the first row as a header row in the destination table.

Table Name:

Table

Data to be pasted:

| Regions |
| North |
| South |
| East |
| West |

☑ Use first row as column headers.

OK   Cancel

7. Click OK. The Regions data has been pasted as a new table.
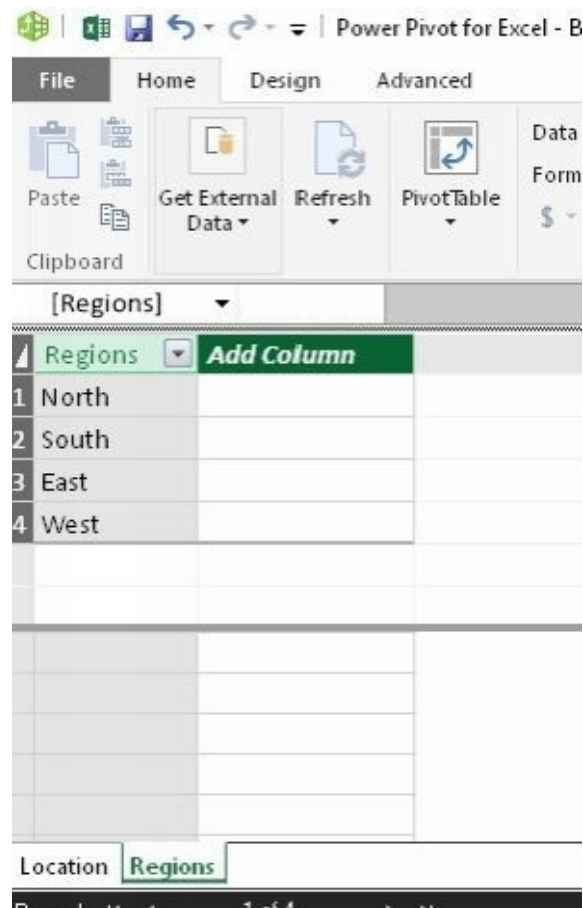
Adding New Records

Adding a new record to the Regions table is pretty straightforward except you need to use a new button.

1. Copy cell A6 from the Regions workbook (the cell that says International).
2. Go back to PowerPivot.
3. In the Home tab, click the Paste Append button.

4. The Paste Preview will appear. Click OK to paste and append the new record.

This is how you add **static** data to PowerPivot. This means that if the source data changes, PowerPivot will **not** update itself. *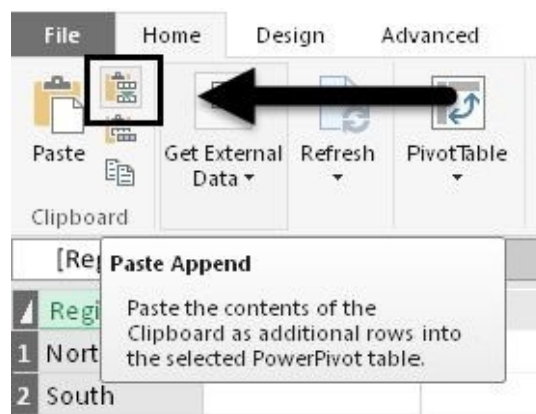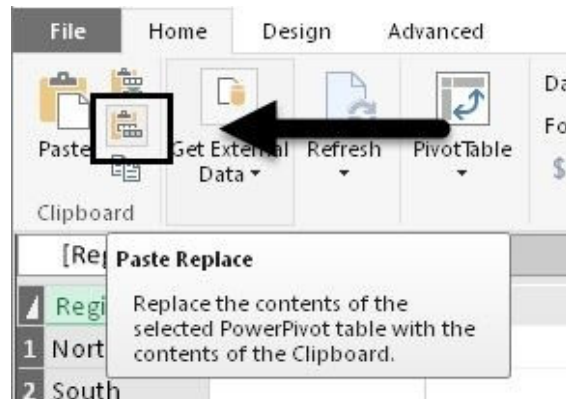*You cannot change data once it is in PowerPivot**. Go ahead, try to type over the North region. You can't. PowerPivot consumes data, it does not change it. If you want to update a static table, you need to copy the new data and select Paste Replace.

Paste Linked Table

This is a much more useful feature. Now you will import a table into PowerPivot but keep the link to the table active. This means once the data in the Excel table is updated, you can refresh the data in PowerPivot and the newly-updated data will appear in PowerPivot. This is very similar to how Pivot Tables work.

5. Right-click on the PowerPivot Regions tab.
6. Select Delete.
7. Click Yes when prompted to delete.
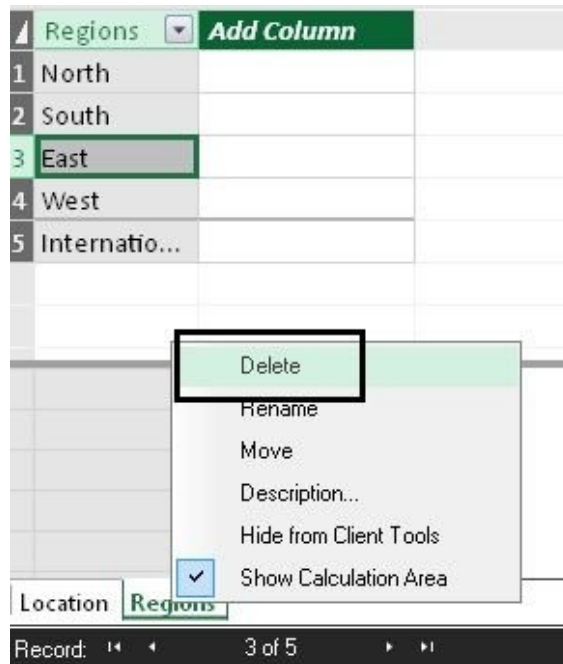
8. Go back to Excel in the Regions.xlsx workbook.
9. In the PowerPivot table, click 'Add to Data Model'.

Excel will first create a data table.

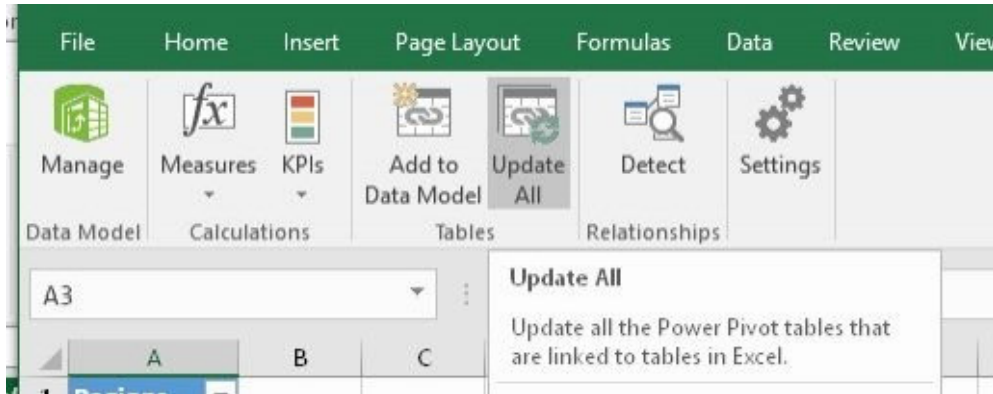10. Check that the data range is correct and that the 'My table has headers' box is checked.
11. Click OK.

That's it. The new table in PowerPivot is linked to the Excel table. You can see in PowerPivot that the linked table has a link icon in the Table tab.

Test out the link.

12.   Change a region name. I changed South to South2 in Excel.
13.   In the PowerPivot ribbon click 'Update All'.

The data in PowerPivot will be updated. Note that if the source file is open, PowerPivot will refresh automatically.

Data Types

PowerPivot is not as forgiving as Excel. When importing data, PowerPivot will try to convert each column into one of its data types. If any cell does not conform to that data type, PowerPivot will default to text. This is another reason why data prep is so important when working with PowerPivot.

PowerPivot understands the following six data types:

Whole numbers: Numbers that have no decimal places between
-9,223,372,036,854,775,808 and 9,223,372,036,854,775,807.

Decimal Numbers: Negative values range from -1.79 e+308 through -2.23E -308.
Positive values range from 2.23 e-308 through 1.79 e+308.

TRUE/FALSE: Either TRUE or FALSE.

Text: Maximum string length is 268,435,456 Unicode characters. Unicode means you can

have characters from non-English alphabets.

Date: Any dates after January 1, 1900.

Currency: Numbers with up to 4 decimal places that range from -922,337,203,685,477.5808 to 922,337,203,685,477.5807.

N/A: This is used to represent SQL NULL values.

You can find the exact specifications in the Microsoft website here:

https://msdn.microsoft.com/en-us/library/gg413463%28v=sql.110%29.aspx

Managing Data

Now that you know how to import data into PowerPivot, you are ready to create a data model. Granted, you *could* just use one table, build a pivot table and call it a day but then why bother with PowerPivot in the first place? The big deal (well, one of the big deals, there are others I will cover later) with PowerPivot is that you can create relationships between tables. If you have worked with Microsoft Access, it is extremely similar.

Let's start by importing a few external data sets and work with them in PowerPivot.

1. Open the SalesData.xlsx workbook.
2. Click in any data cell in the Clients worksheet.
3. In the PowerPivot tab, click 'Add to Data Model'.
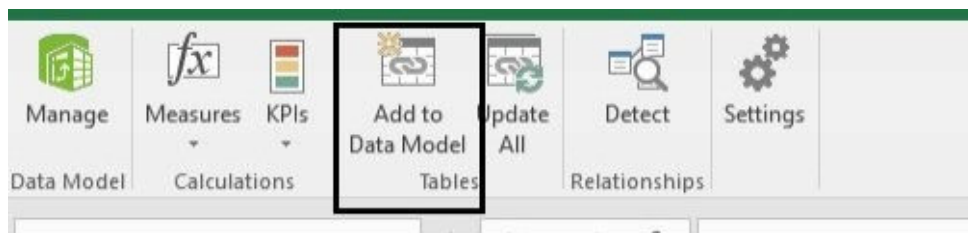
You have now added a linked table to the Data Model.

4. Repeat steps 2 and 3 to add linked tables for the States, Products and SalesData worksheets.

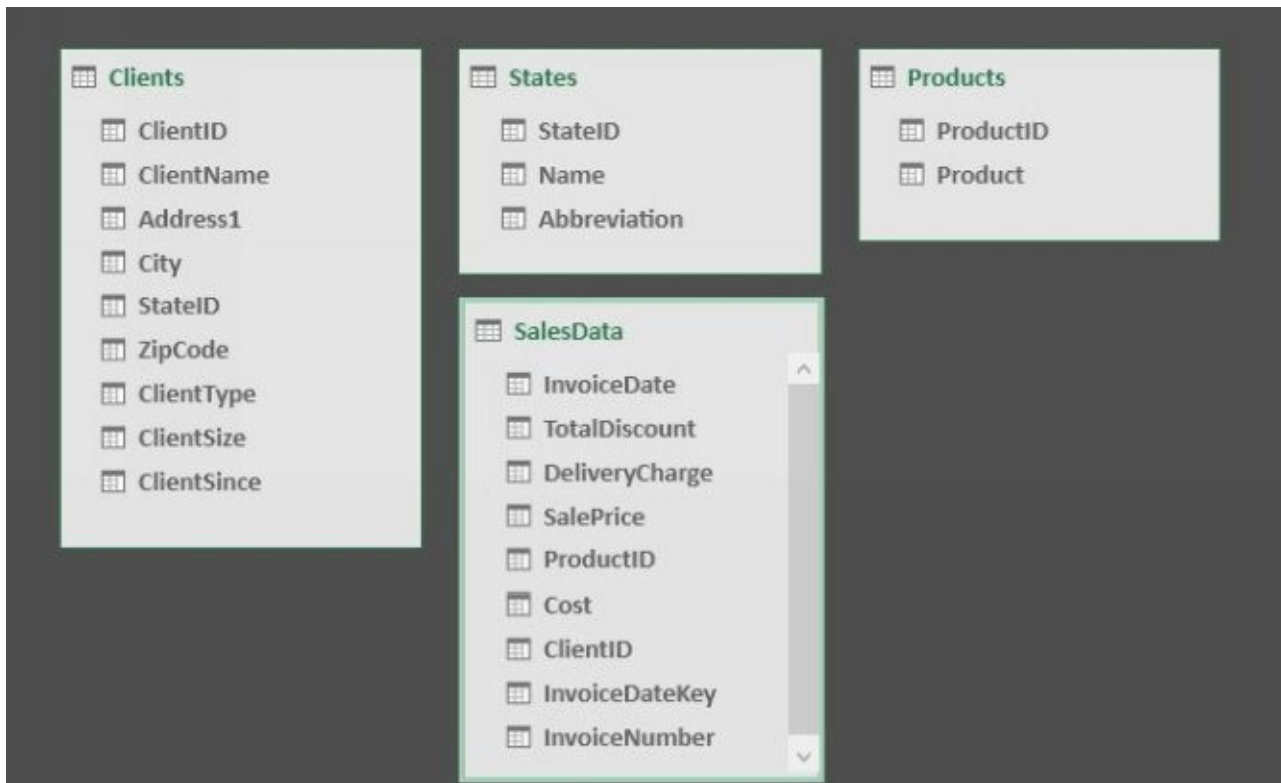You should now have four separate PowerPivot tables.

Right now, you are in the PowerPivot data view. You can see the individual data values and get a feel for the data in each table. To create relationships, it is better to use the diagram view.

5. Click the diagram view button in PowerPivot.

You will see each table and the fields in each.

The data view is great to explore and get familiar with your data. Once you know the data you can switch to this view. In this exercise, I named the key fields explicitly and logically (I wish database people would do the same. You'll see when you connect to the data. Every table will have a 'Key' field instead of 'Client ID').

You can see that the Clients table has a ClientID field. The SalesData table also has a ClientID field. This is the common field and we can use this to create a relationship between Clients and SalesData.

A Quick Explanation

If having common fields between tables is a new concept for you, look at this image:

ClientID	ClientName
1	Barker Corp
2	Carlson Medical Aeon Technology
3	Chemical Biological
4	Davis Foreign Horizon Service

Client ID	Price
1	2622
2	3148
3	4043
4	2837
1	2750
2	4200
3	4790
4	2047

Client ID	Price
Barker Corp	2622
Carlson Medical Aeon Technology	3148
Chemical Biological	4043
Davis Foreign Horizon Service	2837
Barker Corp	2750
Carlson Medical Aeon Technology	4200
Chemical Biological	4790
Davis Foreign Horizon Service	2047

Efficient data storage.

It is easier to store the client name once, in the blue table and store the client number in the green table.

Added benefit: Customer changes name? Change it once.

Inefficient data storage.

A system would have to store repeating customer names for each record.

Problem: Customer changes name? You need to change it in every record.

Using common fields means efficient data storage. It also means faster processing. Although replacing new customer names in the gray table would take a few seconds, imagine if you had several million tables. A Find/Replace operation will take quite some time to finish.

One last point and not to get too geeky on you but you will probably hear these terms when dealing with data; so let me explain them using the blue and green tables in the above image.

Each record in a table should be identified by a unique number. That number is a key. Since it uniquely identifies a record, it is called a **primary key (PK)**. ClientID is the primary key of the client table.

The green table also has the ClientID but you see that it is repeated. That means it is not unique; it can't be a primary key. The ClientID field in the green table is called a **foreign key** (FK). A foreign key is always a primary key of a different table.

This is waaay important for managing data. Keys enforce something called **referential integrity**. It sounds freaky huh? "Hey doc, I woke up with a huge pain in my referential integrity. Can you give me some meds so I can go to work?" :-)

What it means is that every row in the green table **must** have a related record in the Client table identified by a ClientID. The keys maintain the integrity of the references between tables. Why is this important? If we did not enforce this rule, a person could input any number of sales in the green table with no clients! That's fraud! You would be inflating

your sales numbers by entering false sales. No client would ever get billed.

All right, that's as far as I'm going into that here. I left out A LOT of details here, but my goal was just to give you an idea of the why and how these related fields are so important.
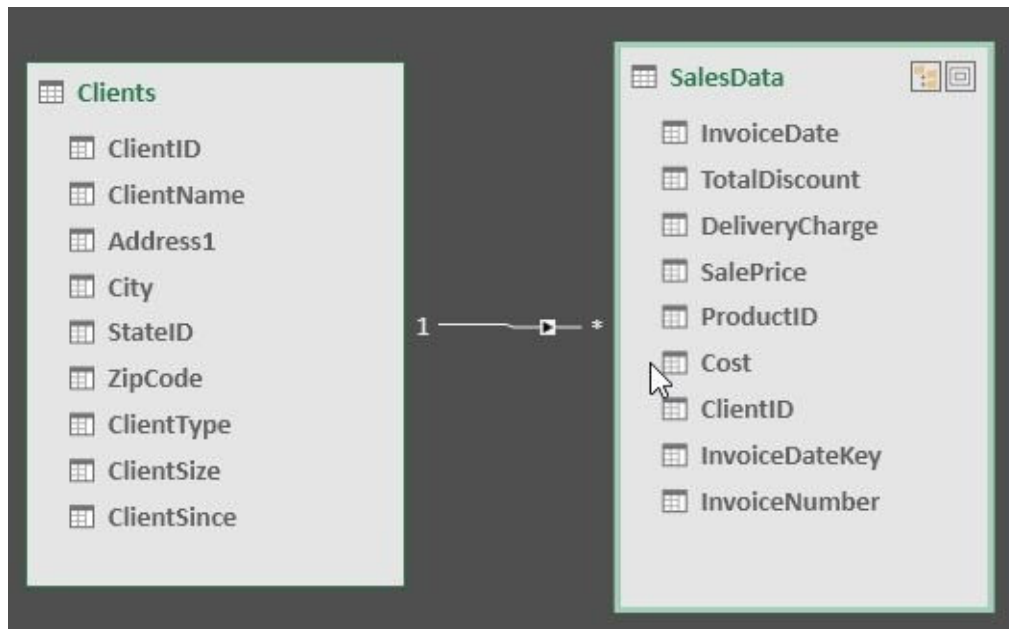
End of Quick Explanation

Back to PowerPivot…

In the database view, creating relationships is straightforward. You click and drag a field from one table to another table.

1. Click the ClientID field in the Clients table and drag it to the ClientID field in the SalesData table.
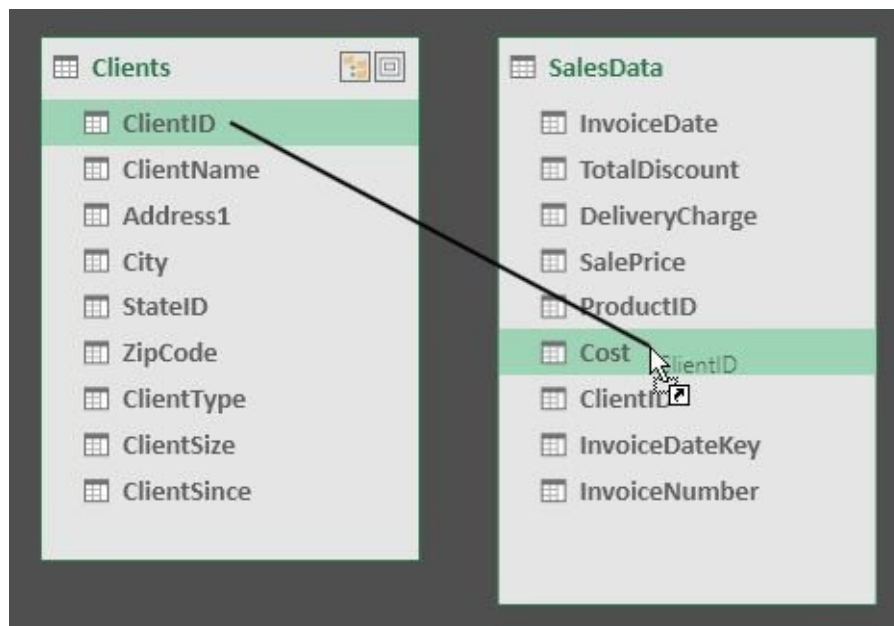
You will see the relationship arrow between the tables.

What's up with that weird arrow, the numbers and the asterisk?

More databasey stuff. The 1 means, well, 1. The * means many. That arrow is shorthand for saying "there is a one to many relationship between Clients and SalesData". In other words, one client can buy many items. Yeah, that makes sense.

Changing Relationships

If you mess up with the click and drag to create a relationship, don't worry. You can right-click the relationship line and select Delete to delete it.

Alternatively, you can double-click on the relationship line to edit the relationship.

1. Double-click on the relationship line.

This pane appears. You can use the drop downs to change the tables and then click the column header to change the related fields.

Sigh, I wish it was that easy to change relationships in real life…

2. Click Cancel to exit this relationship (I'm not touching that one…)
3. Click and drag StateID from the States table to the StateID in the Clients table.

4. Click and drag the ProductID from the Products table to the ProductID in the SalesData table

Your data model will now look like this:
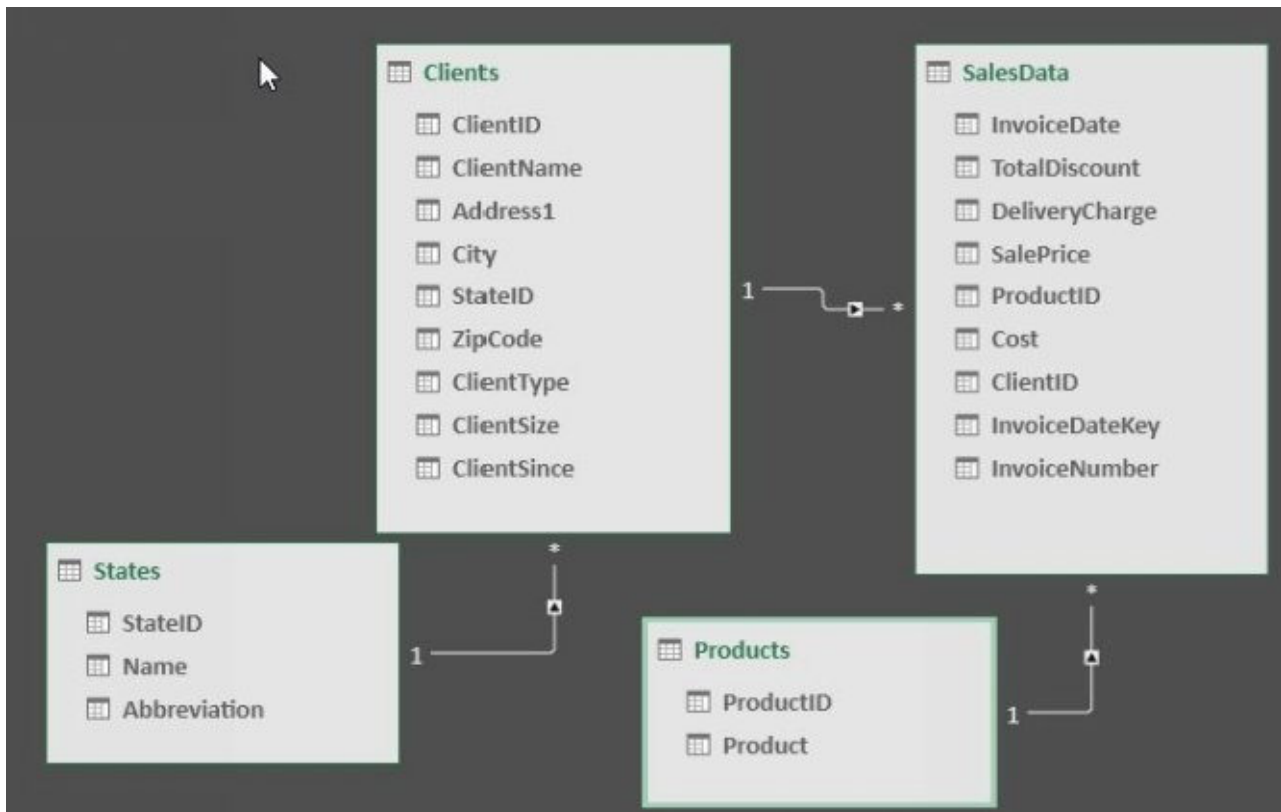
Now you are going to do something pretty cool. You are going to use the data model in a Pivot Table.

5. Click in Excel to start working in Excel. You can leave PowerPivot open or close it; it's your choice.
6. Insert a new worksheet in the SalesData workbook.
7. In the Insert tab, select Pivot Table to insert a new Pivot Table.
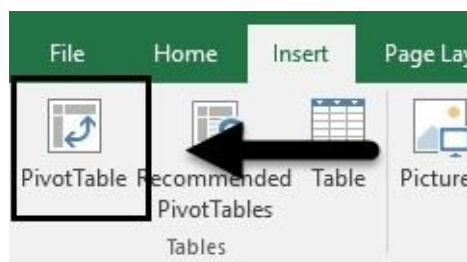
You will now see the standard Pivot Table window. But wait, what data source are you going to use? Look at the pane, there's a new button available, 'Use this workbook's Data Model'
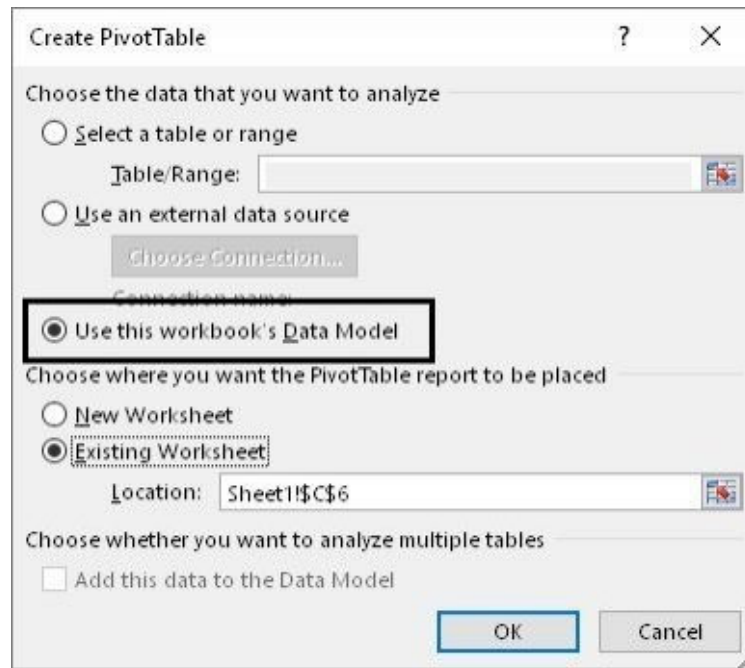
8. Select Existing Worksheet.
9. Select a cell for the Pivot Table location.
   10.   Click OK.

You'll see that the Pivot Table interface looks a little different. Instead of just columns, you see the Data Model tables and their corresponding fields.

Excel 2013 Users

You will not have the 'Use this workbook's Data Model'. Instead follow these steps to source the Pivot Table from the Data Model:

- Click on 'Use External Data Source'

- Click on Choose Connection'

- Click on the Tables tab

- Select 'Tables in Workbook Data Model'

- Click 'Open'

- Build the Pivot Table as usual
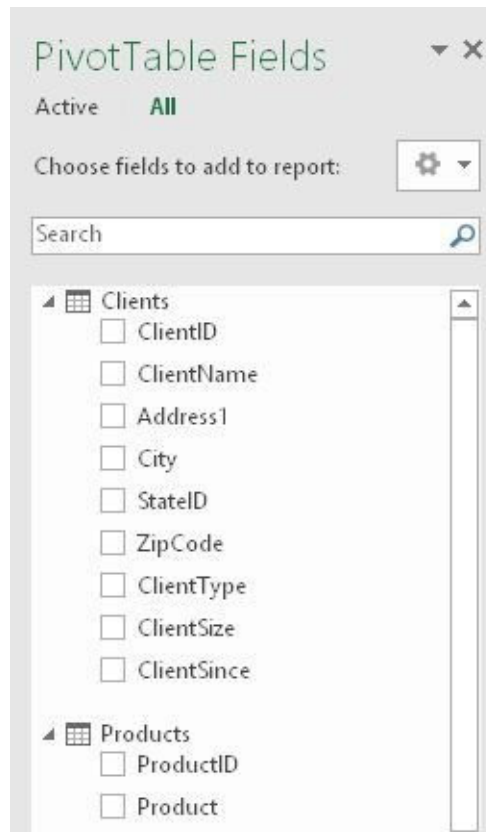
11.   Click on these fields to add them to the Pivot Table: ClientName, Product, SalePrice, Cost.

The Pivot Table should look like this:

Row Labels	Sum of Cost	Sum of SalePrice
⊟ Barker Corp	780915	1125550
Dingeco	116500	181000
Indigofresh	194700	315500
Stim Flex	104700	244800
Supertonstock	278015	336500
Ventoity	87000	47750
⊟ Carlson Medical Aeon Technology	925890	1173250
Dingeco	189390	373750
Indigofresh	117000	198000
Lexilam	62000	178500
Stim Flex	124500	209000
Supertonstock	270500	166250

From now on, you can do anything you usually do with a Pivot Table. You can format the

numbers, add calculated fields, etc.

Take a step back now and think about what PowerPivot let you do. You took three separate worksheets, loaded them into the Excel Data Model and built a Pivot Table using them **as if they were a single table**.

This feature alone will make your dashboards and reports exponentially more powerful. Think about a Data Model that has data in Excel, data from your accounting system and data from the Web. You can combine all those data sources directly in Excel and use a Pivot Table with slicers, conditional formatting, dynamic charts and all the other Excel features to present your information.

Hierarchies

You can create hierarchies in PowerPivot that will make navigating fields in a Pivot Table very easy for you and your users. Let's see how this works.

You are going to add a new table to the SalesData workbook with Regions and then map each State to a Region. Then you will set up a hierarchy where each State is a child of its respective Region.

1. Insert a new worksheet in SalesData.xlsx.
2. Rename the sheet to Regions.
3. Input the Region data per the image below:

	A	B
1	RegionID	RegionName
2	1	Central
3	2	Southern
4	3	Northern
5	4	Western
6	5	Pacific
7		

Now you have to map each State to its Region in the States worksheet. No, I'm not going to make you do each State by hand. That's too tedious! I have already done the mapping for you.

4. Open the StatebyRegion.xlsx workbook.

	A	B
1	Name	RegionID
2	Alabama	2
3	Alaska	5
4	Arizona	5
5	Arkansas	2
6	California	5
7	Colorado	4
8	Connecticut	3

The States in the States worksheet and in the StatebyRegion workbook are in the same order, you just have to copy paste from one into the other.

> 5. Copy column B from StatebyRegion into column D in States (in the SalesData workbook).

The States table should have expanded to include the new column. If it didn't you can manually expand the table.

> 6. Click in any cell inside the State table (you are now in the SalesData workbook).
> 7. Click on the Design ribbon to design the table.
> 8. Click Resize Table.

Table Name:
States
Resize Table
Properties
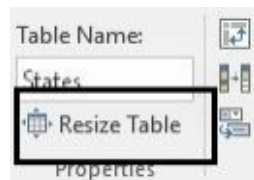
> 9. Change the range to include column D.

Resize Table   ?   ✕

Select the new data range for your table:

=$A$1:$D$51

Note: The headers must remain in the same row, and the resulting table range must overlap the original table range.

OK   Cancel
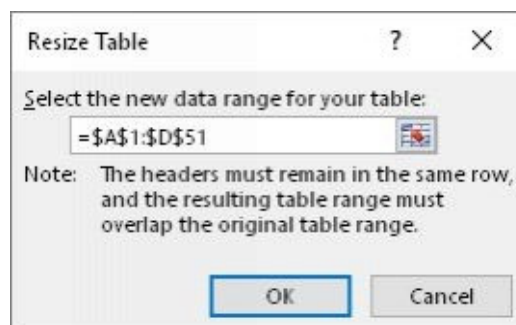
> 10.   Click OK.

You will see new Region column with banded row formatting applied.

All right, you added a new Region table and you added the related records in the State table. Now you have to update the Data Model.

11.   Go back to the PowerPivot window. If you closed it, you can click on Manage in the PowerPivot ribbon.
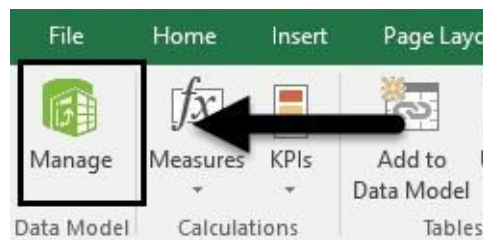
Check out the States Table. The RegionID field is already there. What? How? Remember you built this as a linked table. Any changes to the Excel table will get reflected in the Data Model. Pretty sweet.
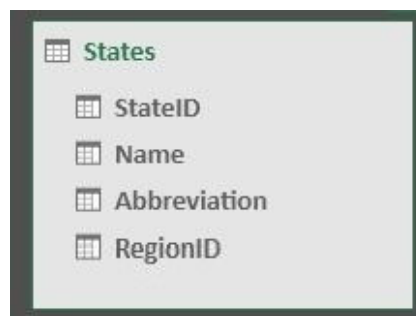
If the RegionID field does not show up, click on the Refresh button in the PowerPivot ribbon.

Now you have to add the Region table to the Data Model.

12.   Go to the new Region worksheet you created.
13.   Click on any cell inside the table.
14.   Click on Add to Data Model in the PowerPivot ribbon.
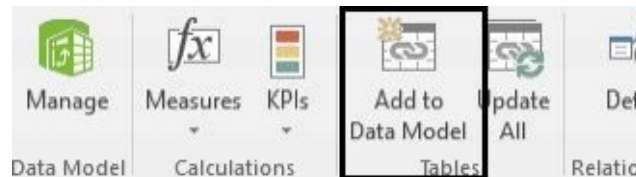
15.   Check to make sure the range is correct. Check the 'My table has headers' checkbox.

16.   Click OK.

The table has now been added to the Data Model. Let's follow best practices and rename the table. Then create a new relationship from this table to the States table.

17.   Right-click on the table and select Rename.
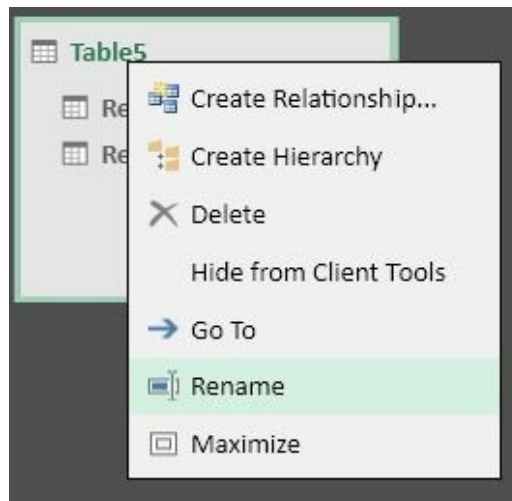
18.   Rename the table to Regions.

19.   Click on RegionID in Regions and drag it over to RegionID in States to create a new relationship between the tables.

Let's recap. All of this has just been prep work! The ultimate goal is to create a hierarchy of Region and State. The problem was that the Data Model did not have the required data. You had to update the source data and then update the Data Model. Along the way, you go some more practice with PowerPivot.

A hierarchy lets users drill down from a parent to a child. In this case, the user will be able to expand the Regions to see the States under it.

There a few conditions that you have to follow when creating hierarchies. The first condition (and it is a bit annoying) is that the fields of the hierarchy **cannot be in different tables**.
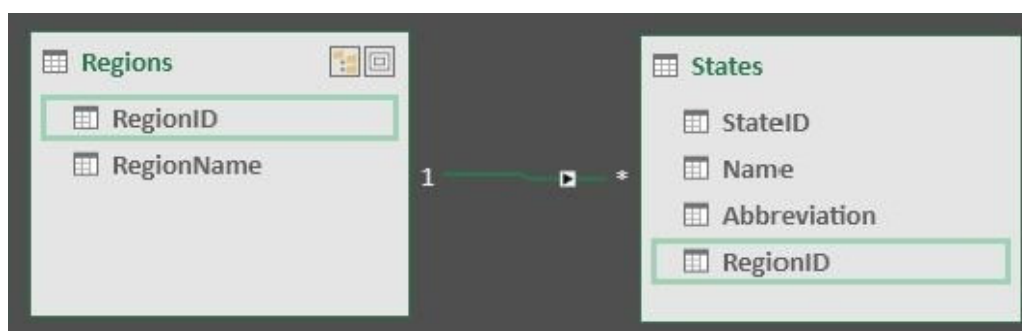
See how Region Name and Name are in different tables? That's how a good data model is

built but hierarchies don't like that.

20. Click on the Grid button to see the data grid view. (It's at the bottom right of the PowerPivot window).

21. Click on the States table.

In Excel you probably have input a helper column with VLOOKUP or other formulas to retrieve extra information from other worksheets. You can do the same thing in PowerPivot. In this case, you are going to add a new calculated column that will retrieve the Regions from the Region table. Conceptually, it is exactly like writing a VLOOKUP formula. However, you are doing it in PowerPivot and it is much simpler since you have already set up the table relationships. Once again, we are doing this just so we can have States and Regions in the same table.

Nooooo, you do not want to redesign your tables to put the Regions directly in the States tables. If you have multi-million rows, it will slow everything down to crawl.

Let's bring in the related records from Region into States. Wait until you see how easy this is…

22. Click on the first row in the Blank column, then click in the formula bar.
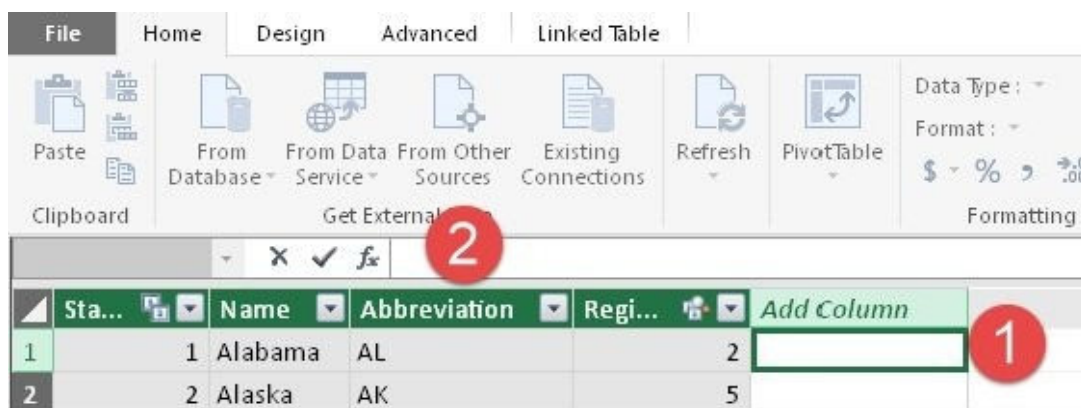
23.   Type in =Related(

Since there is only one relationship in this table, PowerPivot lists the potential choices for you.
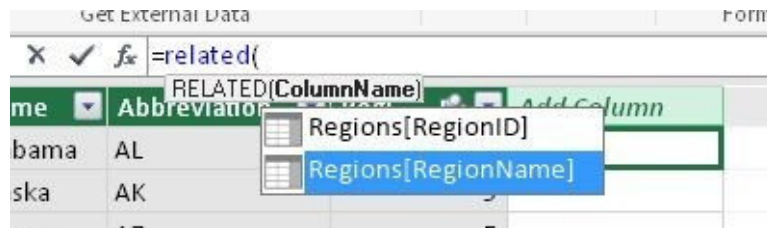
24.   Choose RegionName.
25.   Type in) to close the parenthesis.
26.   Press Enter.

The related Region names will be displayed in the States table.

27.   Right-click on the new Calculated Column 1 and select Rename.
28.   Rename the Column RegionName.

Now that you have both the parent (RegionName) and the child (Name) in the same table, you can build a hierarchy.

29.   Click the diagram view button (at the bottom right of the PowerPivot window).
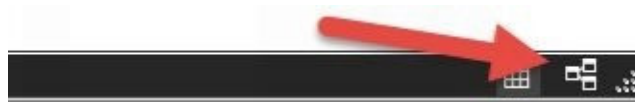
30.   Right-click on the RegionName field in the States table.
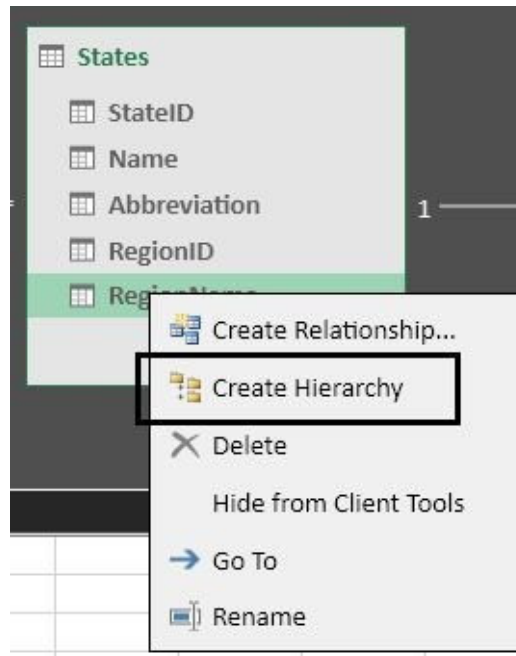
31.   PowerPivot starts by giving you the ability to rename the hierarchy. Rename it to Geography.
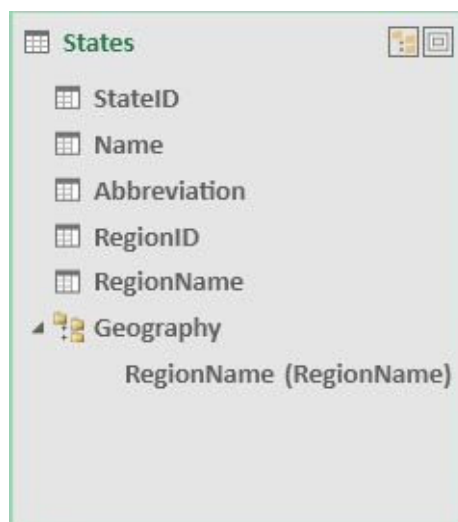
32.   Click and drag the Name field under RegionName (RegionName).
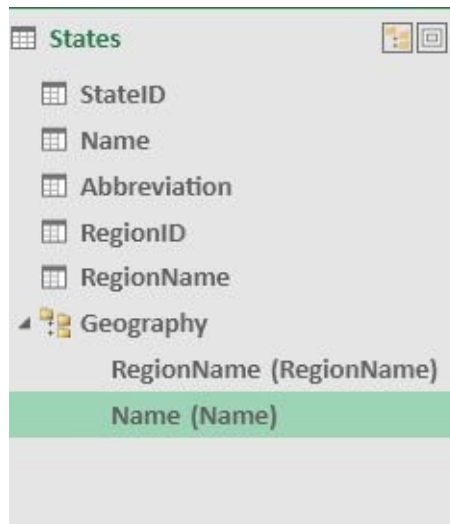
Now let's see how this new hierarchy can be used.

33.   Click back in the Excel Pivot Table.

34.   Click Refresh to refresh the Pivot Table.

35.   Expand the States tab in the Pivot Table fields pane. You will see the new Geography hierarchy.
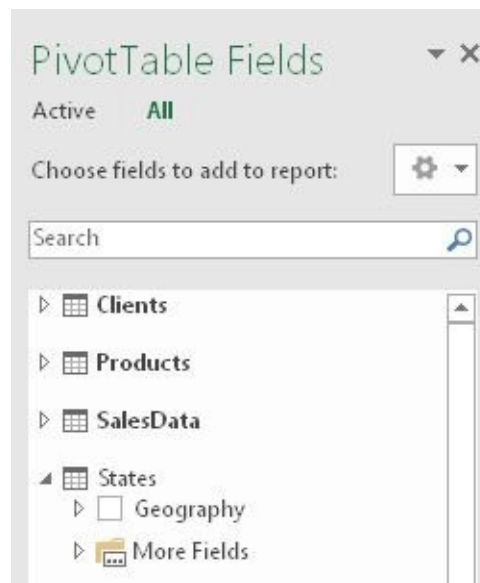
36.   Drag the Geography item to the columns and remove the SalePrice field (you are removing it so it's easier to see the hierarchy).
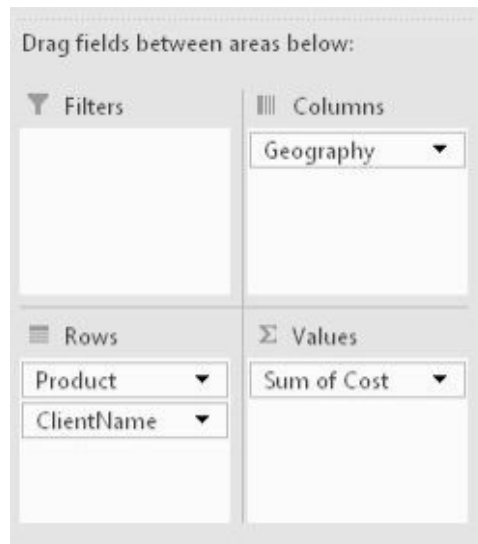
Take a peek at your Pivot Table. You now see the parents Central, Northern, Pacific, Southern and Western. However, you can click on them to drill down to their children.

This hierarchy had just one level but you can add as many levels as you need to. To do add more levels, drag additional fields in PowerPivot into the hierarchy. The order of the fields under the Geography hierarchy determines which is parent and child. You can move the fields around until you get the right order.

If you drag Geography to the Filters section of the Pivot Table, Excel will filter the entire Pivot Table by the parent or child.
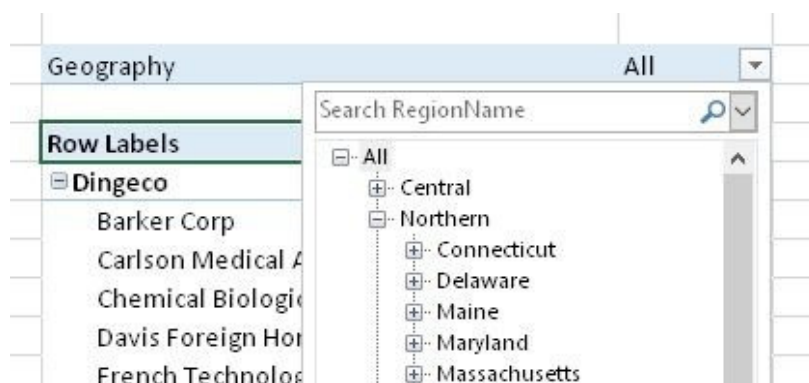
Bonus Exercise

I think this is so cool. I had to share it with you.

1. Click on any cell in the Pivot Table.
2. Go to the Insert tab.
3. Click on Slicer.
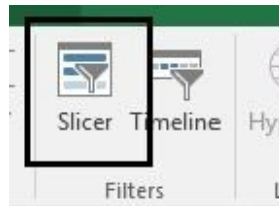
4. Select Geography.
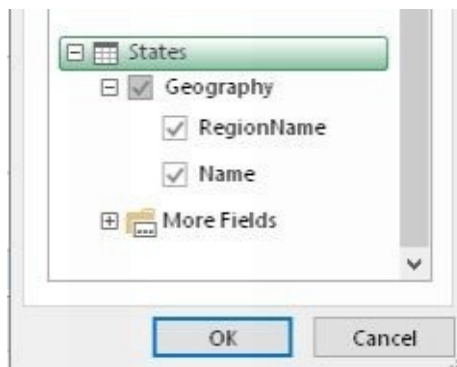
5. Click OK.

Two slicers are created. One for RegionName and another for Name. The great thing is that if you select a Region, all the related States will also be selected. Users can select a Region (and all the States in the region) or an individual State.

RegionName				
Central				
Northern				
Pacific				
Southern				
Western				

Name				
Illinois	Indiana	Iowa	Kentucky	Michigan
Minnes...	Missouri	Ohio	Wiscon...	Connec...
Delaware	Maine	Maryland	Massac...	New Ha...
New Je...	New York	Pennsyl...	Rhode I...	Vermont
Alaska	Arizona	California	Hawaii	Nevada
Oregon	Utah	Washin...	Alabama	Arkansas
Florida	Georgia	Louisiana	Mississi...	North C...
South C...	Tennes...	Virginia	West Vi...	Colorado
Idaho	Kansas	Montana	Nebraska	New M...
North D...	Oklaho...	South D...	Texas	Wyoming

Calculated Columns

Calculated columns are almost identical to the calculated columns you use in Excel. In Excel, you input a formula in a cell, then you drag it down to the last row. In PowerPivot, you just click in 'Add Column', input the formula in the formula bar and press Enter. PowerPivot will apply the formula to every row.

PowerPivot uses a new formula language called Data Analysis Expressions (DAX). The basic functions in DAX are similar to regular Excel formulas. You'll be comfortable with them soon. The more complex applications of DAX will take some practice but they really extend the analytical capabilities of PowerPivot.

Let's create a simple calculated column that derives the Net Profit in the Sales Data table.

1. In PowerPivot, go to the SalesData table.
2. Click in any cell in the 'Add column' column.
3. Type =[

You'll see that PowerPivot displays a list of all the available fields

4. Finish the formula for Net Profit.

    =[SalePrice]-([Cost]+[DeliveryCharge])

5. Press Enter.
6. Right-click on the new column and select 'Rename Column'.
7. Rename the column to Net Profit.

Now this field will be available for you to use in the Pivot Table.

Calculated Fields

Calculated fields (also known as Measures) are calculations that are not evaluated on a row by row basis; instead they are evaluated on the entire table. For example, calculating how many units were sold in a region or by quarter are not row-specific.

Calculated fields are input in the calculation area under the data table.

Once again, I know I'm sounding like a broken record but it important to keep the big picture in mind. You could mimic calculated columns and calculated fields directly in Excel…but not for 3 million rows. PowerPivot is for dealing with extremely large data sets. You can certainly use it for smaller data sets because it manages data more efficiently than pure Excel. The other reason that these calculations are important is because the other Excel reporting tools (Power View and Power Map) need to have the calculations in the Data Model in order to use them.

Ok, enough about that. Let's build a calculated field.

1. You should be on the SalesData tab, if not click on it.
2. Click in a blank cell in the calculation area.

Calculated Fields also use DAX formulas. As I mentioned, DAX is new but similar to Excel formulas. An easy way to see what formulas are available is to use the PowerPivot DAX formula wizard.

3. Click the Formula Wizard button.

4. Scroll down until you see the COUNTROWS function. Select it.
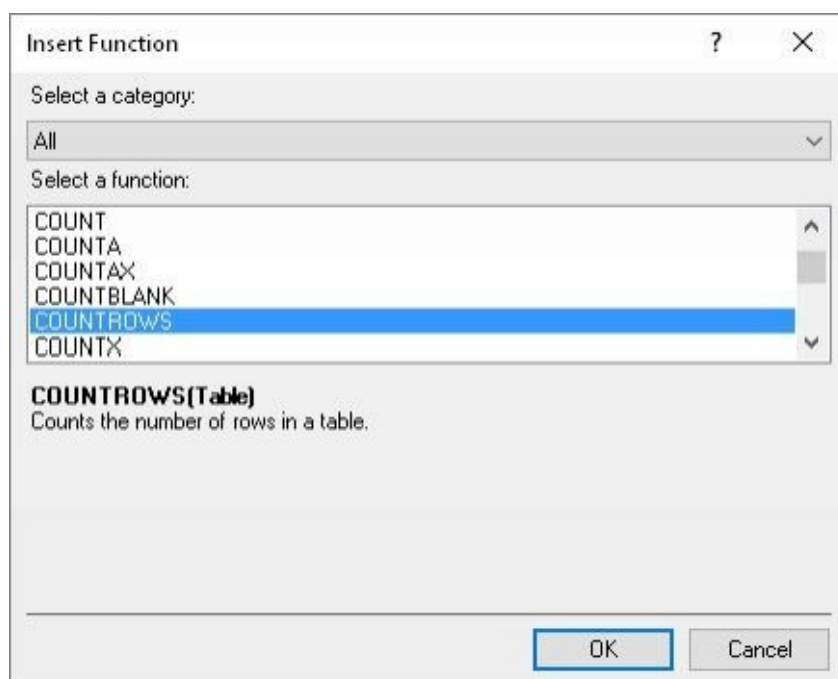
5. Click OK.
6. In the formula bar, finish the formula so it reads =COUNTROWS(SalesData).
7. Change the text before the colon to read SalesRecords.

The final formula should look like this:

SalesRecords:=COUNTROWS(SalesData)

Note that the formula syntax is a bit different. There is a := in between the name and the formula

8. Press Enter.

Your new calculated field will appear in the calculation area.

Now let's see how this works in the Pivot Table.

9. Go into Excel and choose a cell inside the Pivot Table you built.
10.   Expand the Sales Data table and you will see the calculated field.
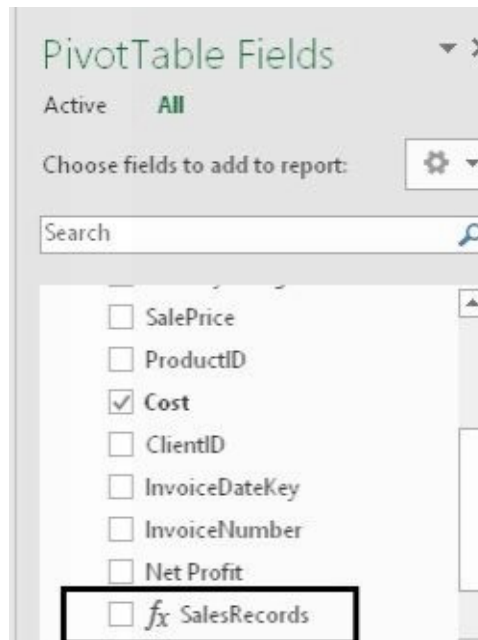
11.   Drag the SalesRecords field to the Values area of the Pivot Table.
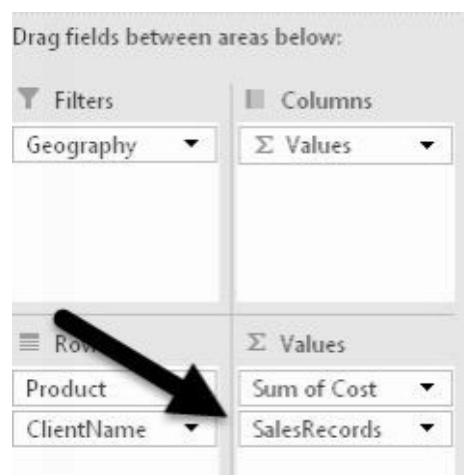
The Pivot table will now calculate the number of records **based on the group and slicer settings**. The SalesRecord calculation in PowerPivot, just gives you one number, the total number of rows. However, in a Pivot Table the SalesRecord calculation 'listens' to the Pivot Table and recalculates based on your selections.

You can certainly test it out (and you always should test your calculations). The Pivot Table says there are two records for Gold Game Confectioners. Double-click on the 2 or the 43200 to drill down. Excel will pull up the individual records. You must see two records in the new worksheet to validate the formula is working.

	A	B	C	D	E
1					
2					
3					
4			Geography	Northern	
5					
6			Row Labels	Sum of Cost	SalesRecords
7			⊟Dingeco	60200	3
8			Gold Game Confectioners	43200	2
9			Network Sciences	17000	1
10			⊟Indigofresh	96200	3
11			Paragon Dynomantics	96200	3
12			⊟Supertonstock	31125	1
13			Gold Game Confectioners	31125	1
14			Grand Total	187525	7
15					

Pretty cool huh?

You can also do aggregate calculations in calculated fields for example:

TotalSales:=sum(SalesData[Cost])

Would give you the total sales for the entire SalesData table.

KPI

KPI's are Key Performance Indicators. They are metrics that you design and they create visual alerts in PowerPivot to call attention to values that are out of line. In other words, KPI's work like the Icon Sets in the Excel Data Validation ribbon.

Let's create a KPI so you can see how it works.

A KPI needs a calculated field as a base. Let's create a new one.

1. Click the calculation area in the SalesData tab.

2. Type in this new calculated field:

$$AvgNetProfit:=AVERAGE([Net\ Profit])$$

3. Right-click on AvgNetProfit and select 'Create KPI'.
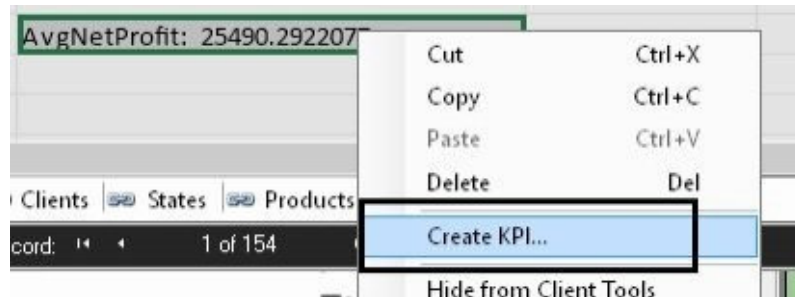
Note that KPI's are only available for measures created in PowerPivot. If you create a calculation in Excel and then include it in PowerPivot, the calculation will not be available for a KPI to use.

4. Change the KPI window so it matches these values (Absolute value 15000, select the first icon style and move the slider to around where I have them, it doesn't have to be exact).

5. Click OK.

The KPI in the calculation area now has a small icon added to it as an indicator that there is a KPI associated with it.

SalesRecords: 154
AvgNetProfit: 25490.2922077922

The KPI is also added to the Pivot Table.

6. Expand the SalesData fields to see the new KPI.
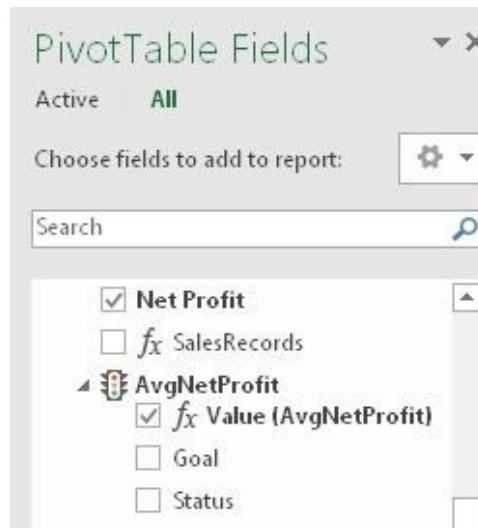
Go ahead and change the Pivot Table to see how it can now include the calculated field and the Status. I changed my Pivot Table around and this is how it looks:

There's a lot going on in this small Pivot Table.

- It has limited the data to only the customers that are in the Northern Region. Recall that we set up Region as a hierarchy.
- It is displaying the Net Profit which was never calculated in Excel; it is a Calculated Column in PowerPivot.
- It has a Calculated Field (AvgNetProfit). AvgNetProfit is calculated for each customer. (This is very nice. I only created one formula in the calculation area but I can use it for every customer).
- The Avg Net Profit Status KPI appears in the Pivot Table.

Optional Mini Exercise

Here's a small test to make sure that the data is flowing through from Excel to PowerPivot, through the calculated Fields/Calculated Columns/KPI and into the Pivot

table.

1. In Excel, go to the SalesData worksheet.
2. Find a data point that is in the Pivot Table.

For example, I want to change Gold Game Confectioners. The Clients worksheet tells me that it is ClientID 7. I use Autofilter to find the rows that have ClientID =7 in Sales Data.

	A	B	C	D	E	F	G	H	I
1	InvoiceDate	TotalDiscount	DeliveryCharge	SalePrice	ProductID	Cost	ClientID	InvoiceDateKey	InvoiceNumber
57	9/23/2013	500	775	37690	3	25700	7	20130923	1055
60	10/4/2012	775	1750	22500	3	17500	7	20121004	1058
81	2/2/2013	500	775	41250	4	31125	7	20130202	1079
156									

3. I'm changing D81 to another number. I'm adding a few zeros to make it 4125000.
4. Click back on the Pivot Table.
5. Right-click on a Pivot Table cell and Select Refresh.

The Pivot Table values and the KPI should update. If they do, all your data linkages are working.

Row Labels	Sum of Net Profit	AvgNetProfit	AvgNetProfit Status
⊞Northern	4,417,540	631,077	●
Gold Game Confectioners	4,107,565	1,369,188	●
Network Sciences	20,750	20,750	●
Paragon Dynomantics	289,225	96,408	●
Grand Total	4,417,540	631,077	●

Calendar Table

	A	B	C	D	E	F	G	H	I	J
1	InvoiceDate	TotalDiscount	DeliveryCharge	SalePrice	ProductID	Cost	ClientID	InvoiceDateKey	InvoiceNumber	
2	2/1/2012	500	0	45800	1	25700	1	20120201	1000	
3	7/2/2013	50	0	110000	2	37500	2	20130702	1001	
4	2/2/2013	450	1750	110000	1	22500	3	20130202	1002	
5	1/9/2013	50	50	39500	1	37500	2	20130109	1003	
6	4/15/2013	50	500	37000	1	75890	5	20130415	1004	
7	4/2/2013	500	500	44000	1	37500	4	20130402	1005	
8	11/2/2013	500	50	39500	2	99000	3	20131102	1006	
9	5/2/2013	975	500	102500	3	17500	6	20130502	1007	

Take a look at the raw data in the SalesData table you've been working with. So far, it has served you well for learning PowerPivot but truthfully, it's not enough for meaningful analysis. For example, what if you want to analyze TotalDiscount by month, or Net Profit by quarter? There is no way to do that in PowerPivot. PowerPivot doesn't know what a month is, or what a quarter is. Remember, you can't change data in PowerPivot, if it isn't loaded or calculated in PowerPivot you are out of luck.

The solution is to create a Calendar Table that stores all the dates in the raw data and assigns date measures to each record. For example, a Calendar Table would have **every single day (No gaps allowed! Gaps will mess things up!)** in one column and the quarters, months, etc. that the day corresponds to. You then load the Calendar Table into PowerPivot, set it as a date table and create a relationship to the SalesData table.

Basically the Calendar Table will have every possible date representation you will need in your visualizations, year, month, quarter, etc.

By the way, when building a Pivot Table, PowerPivot tries to help you in this respect. I'm not sure if you noticed but PowerPivot already parsed out the Invoice Date into Date measures.
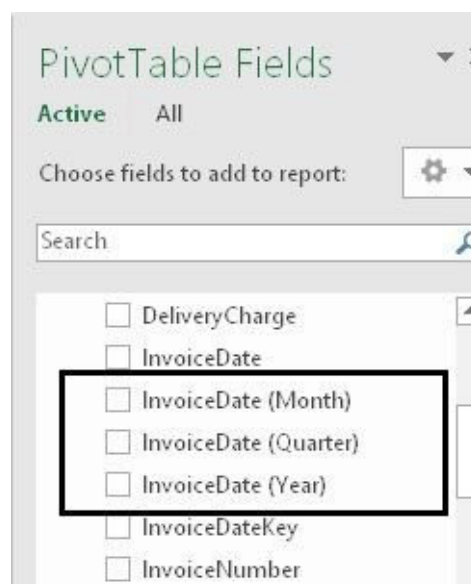
If all you are doing is building Pivot Tables and you just need Month, Quarter and Year, you are all set. However, if you want to use Power Map, Power View or more date measures, you will need to build a Calendar Table.

The easiest way to build a Calendar Table is to use Excel. List the days in one column and

use the built-in Excel date functions for the rest of the columns. I'm not going to make you do all that work! I have a basic Calendar Table that you can use to get you started. Once you have it set up to have what you need, you just have to import it into each PowerPivot data model you build.

Let's do that now.

1. Go to PowerPivot (if you were in Excel).
2. In the Home tab, click 'From Other Sources'.
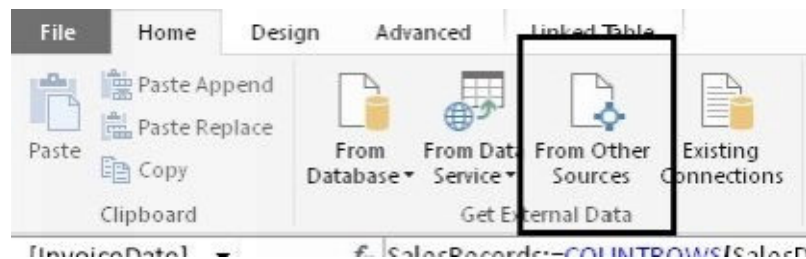
3. Towards the bottom of the list, click 'Excel File'.

Table Import Wizard

Connect to a Data Source
You can either create a connection to a data source, or you can use one that already exists.

Data Feeds

Report
Create a connection to a Microsoft Reporting Services Report. Import data from the feed.
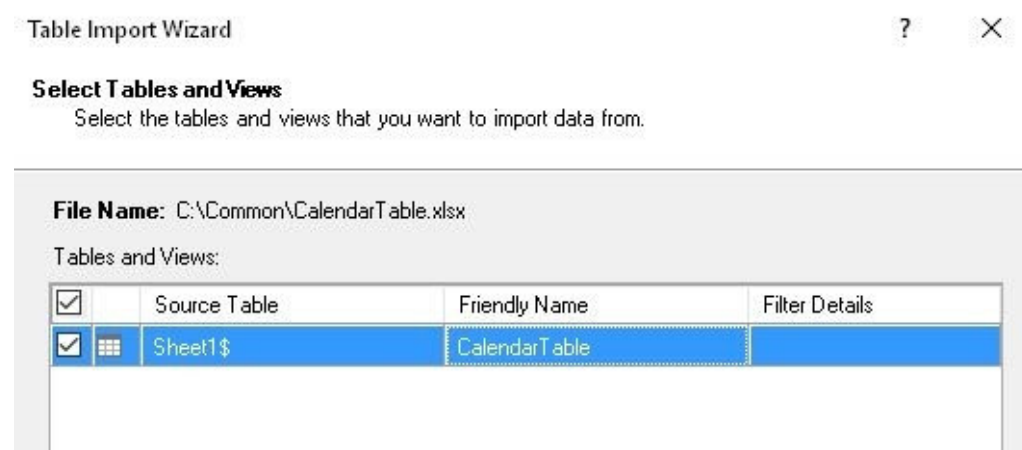
From Microsoft Azure Marketplace
Get external from Microsoft Azure Marketplace.

Suggest Related Data
Get suggestions of external data.

Other Feeds
Create a connection to a data feed. Import data from the feed.

Text Files

Excel File
Import data from an Excel file.

Text File
Import data from a text file.

< Back | Next > | Finish | Cancel

4. Click Next.
5. Browse to the location of the follow-along workbook CalendarTable.xlsx.
6. Check 'Use first row as column headers'.
7. Click Next.
8. Click on Sheet1 in Friendly Name and rename the tab to CalendarTable.

Table Import Wizard

Select Tables and Views
Select the tables and views that you want to import data from.

File Name: C:\Common\CalendarTable.xlsx

Tables and Views:

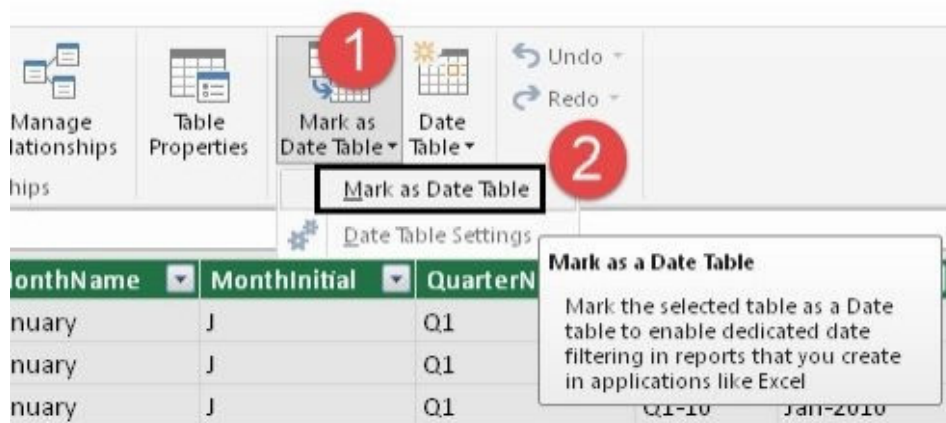| ☑	Source Table	Friendly Name	Filter Details
☑ ⊞	Sheet1$	CalendarTable	

9. Click Finish.
10.   Click Close.

Now you have to tell PowerPivot to treat this table as a Date Table.
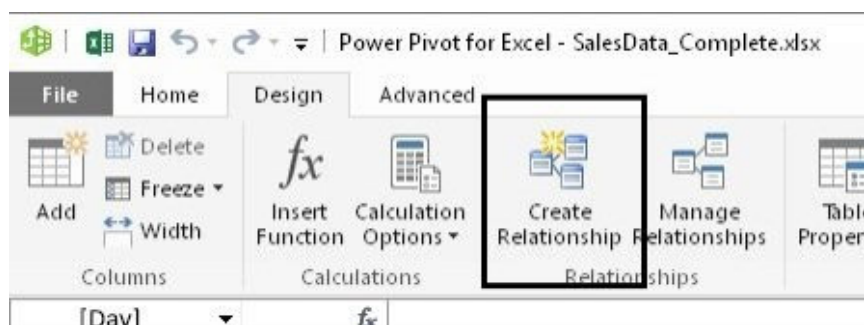
11.   In the Design tab, click 'Mark as Date Table'.


12.   Select Day as the Date field. (Note that the field must have unique values).
13.   Click OK.

Now you just have to create a relationship between the Calendar Table and the table that has the data you want to analyze over time (SalesData).

14.   In the Design tab, click 'Create Relationship'.

15.   In the Create Relationship window, select SalesData as Table2.
16.   Click on the Day field then on the InvoiceDate field to set up a relationship between the two.
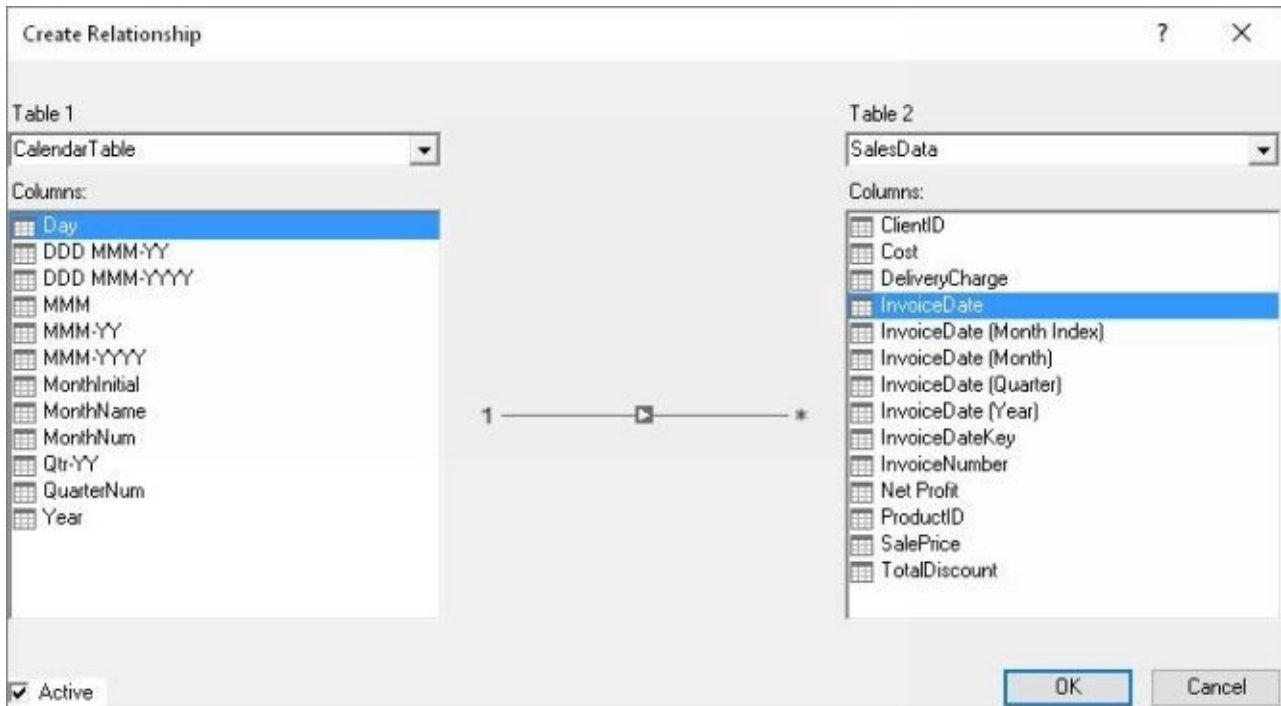
17.   Click OK.

Note: In my case, every once in a while, I get a 'Create Relationship Failed' error (No jokes about this! :-))

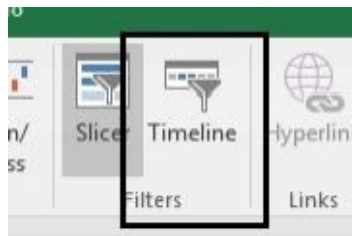When I click OK, again the relationship is created successfully. If this happens to you, try recreating the relationship until it works.

Pivot Table Tip

Let me throw this in here.

When you have Date fields you can use a new type of Filter, the Timeline to control Pivot Tables.

1. Go back to your Pivot Table.
2. Click on the Insert tab.
3. Click on Timeline.

4. Click on InvoiceDate.
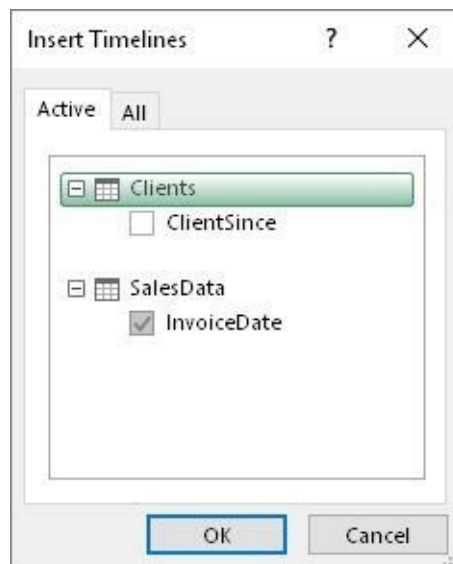
5. Click on OK.

Now your users have a much more intuitive way to select a range of dates and see the related data in a Pivot Table.

Dashboards

Now you can use PowerPivot to manage million and millions of data points. You create a few calculated columns and a few calculated fields. Then you have to use a Pivot Table to

present your data. Really? Granted, Pivot Tables are very powerful, they can be augmented with slicers and timelines but you can't really make a dashboard with them. You can't insert rows in the middle of a pivot table. Formatting is pretty tedious. There must be a better way.

There is a better way. Go get some coffee, take a breather or something because what I am going to show you is way cool and almost no one knows about it.

I'm going to show you how to create a dashboard with formulas that can read data directly from the Data Model…no Pivot Table needed. Think of it as a VLOOKUP that retrieves data from the Data Model.

Let's start by creating a simple formula that pulls in SalePrice then you can get creative and make it more flexible.

New Formula: CUBEVALUE

You are going to use the CUBEVALUE formula to retrieve data from the Data Model. This formula lets Excel connect to a data cube.

You are familiar with tables and table relationships. Those systems are very good for data storage. For analysis, those systems slow down considerably. There's a technology where you can store the data in a multidimensional cube. Users can pivot the data in the cube to drill down/up to perform analysis. This is called OLAP (Online Analytical Processing) and the cubes are called OLAP cubes. A PowerPivot data model is one of these types of cubes; that's why you can use the new cube formula to retrieve data from the Data Model.

The syntax for CUEBVALUE is:

$$=CUBEVALUE(connection,member\_expression1,…)$$

CUBEVALUE retrieves data from the calculation area of PowerPivot. You only created two calculations so far, SalesRecords and AvgNetProfit. However, PowerPivot created some **Implicit Measures** for you. In other words, PowerPivot did some calculations but didn't tell you about them.

1. Insert a new worksheet in the SalesData follow-along workbook. Name it

Dashboard.
2. In PowerPivot, click on the Advanced tab.
3. Click on 'Show Implicit Calculated Fields.

Now you can see all the measures in the SalesData tab.

9	12/2/2013 12:00:00 AM	50	50	39500
10	12/2/2013 12:00:00 AM	50	500	110000
11	5/18/2013 12:00:00 AM	500	500	95000
12	6/2/2013 12:00:00 AM	1000	50	110000

SalesRecords: 154   AvgNetProfit: 2..   Sum of SalePrice: 1034..

You are going to start by retrieving the NetProfit from the Data Model. This formula will be the basis of the dashboard.

4. Choose a cell around F12 or so. Type in this formula:

=CUBEVALUE("ThisWorkbookDataModel","[Measures].[Sum of SalePrice]")

This formula connects to ThisWorkbookDataModel and retrieves Sum of Sale Price from Measures (remember that Calculated Fields are also called Measures).

This is ok, but you need some more granularity of the data. Let's change it so we can see Sum of SalePrice for the Northern Region (remember that the Northern region is part of the hierarchy you created. Yes, those are still available to this formula.)

You just need to start specifying additional criteria in the CUBEVALUE formula. The criteria follows the form of [Table].[Field].[Value].

To get sales for the Northern Region, change the formula so it read like this (You'll also notice that Excel helps you by displaying the available fields….very handy):

=CUBEVALUE("ThisWorkbookDataModel","[Measures].[Sum of SalePrice]","[States].[RegionName].[Northern]")

You should see a cell with $527,690.

The rest of this exercise is going to be basically splitting up the above formula to point to cells with the values instead of hardcoding them inside the formula.

5. Set up your dashboard like this:

	A	B	C	D	E	F
1						
2						
3						
4						
5						
6						
7						
8			Sum of Sale Price			
9						
10						
11				2012	2013	
12			Central	527,690		
13			Southern			
14			Northern			
15			Western			
16			Pacific			
17						

Instead of having "Northern" hardcoded in the formula, you are going to build the string using the & operator.

6. Change the formula to this:

=CUBEVALUE("ThisWorkbookDataModel","[Measures].[Sum of SalePrice]","[States].[RegionName].["&$C12&"]")

C12 has the region name. Now you have made the formula more flexible.

7. Add a new member expression to retrieve data by year. The updated formula is:

=CUBEVALUE("ThisWorkbookDataModel","[Measures].[Sum of SalePrice]","[States].[RegionName].["&$C12&"]","[CalendarTable].[Year].["&D$11&"]")

8. Drag the formula down and across to fill out the dashboard.

I added some formatting to make my dashboard look pretty for you.

| Sum of Sale Price	2012	2013
Central	244,000	1,520,750
Southern	537,800	2,317,190
Northern	201,000	326,690
Western	518,690	1,368,750
Pacific	1,172,550	2,136,790

Since these are formulas, you can insert rows and apply whatever formatting you like, this is much more flexible than a Pivot Table. Of course, the drawback is that this is not suited for any type of ad-hoc analysis, you can't pivot anything.

Do you realize how cool this is? Theoretically, you can have multi-million data rows in PowerPivot and build a dashboard off all that data without ever using a Pivot Table.
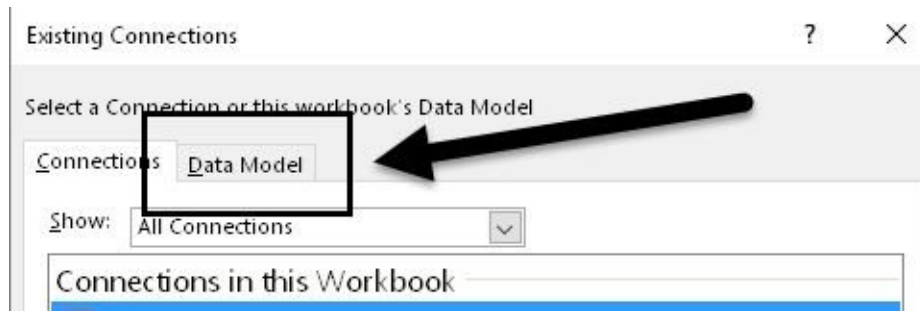
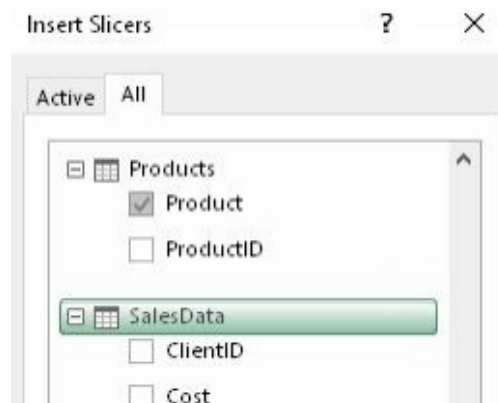But wait, there's more. It gets even better…

9. Click on the Insert tab.
10. Click on the Slicer button.

Slicer Timeline F

Filters
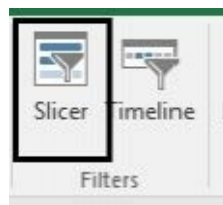
11. Click on the Data Model tab.

12.   Click Open to open the data model.

13.   Scroll down the pane and select Product.
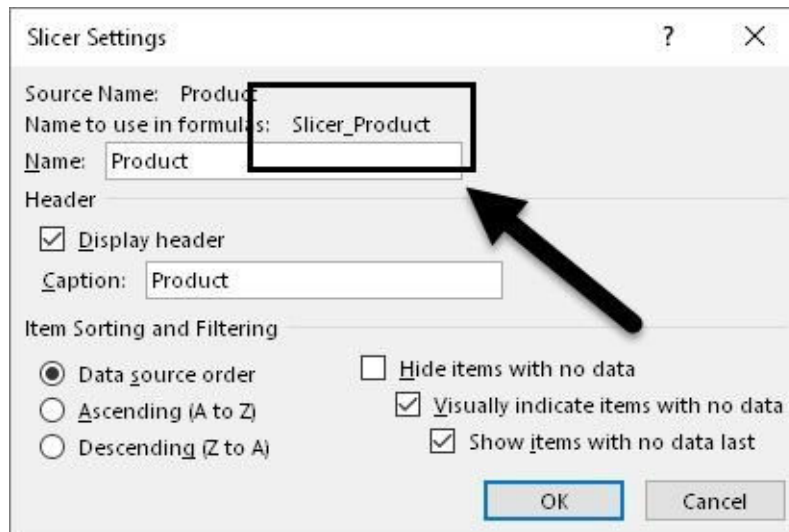

14.   Click OK.

This slicer is now directly connected to the Data Model and will retrieve all the Products.

Now you are going to connect the Slicer to a cell.

15.   Right-click on the slicer and select Slicer Settings.

16.   Don't change anything here but make a note of the name of the slicer. It is Slicer_Product. You will need that shortly.

17.   Click on the cell right under the label "Sum of Sale Price". In my dashboard it is cell C9.

18.   Type in this formula:

=CUBERANKEDMEMBER("ThisWorkbookDataModel",Slicer_Product,1)

CUBERANKEDMEMBER is the cube formula that retrieves a specific member from a data set. In this case, the data set is the Slicer_Product data set and it is retrieving the first value.

19.   Change the selected product slicer to see the new value reflected in cell C9.

Note: This trick will not work with multiple slicer selections.

Guess what you are going to do next? Yup, add another member expression to include Product in the CUBEVALUE function.

20.   Update the formula to include a reference to cell C9 (or your particular cell that is linked to the slicer).

=CUBEVALUE("ThisWorkbookDataModel","[Measures].[Sum of SalePrice]","[States].[RegionName].["&$C12&"]","[CalendarTable].[Year].["&D$11&"]","[Products].["&$C$9&"]")

21.   Drag the formula down and across to fill in the dashboard.

	2012	2013	Product	
Sum of Sale Price			Dingeco	
Lexilam			Indigofresh	
			Lexilam	
Central		178,500	Softtrax	
Southern	39,500	141,940	Stim Flex	
Northern				
Western		171,250	Supertonstock	
Pacific	179,250	174,000	Ventoity	

Now as you select a product using the slicer, the CUBEVALUE formula will read the linked cell and return data from the Data Model.

DAX Formulas

In this lesson, I only briefly touched on DAX formulas. This was by design. DAX is an enormous topic that really deserves its own lesson.

This was quite the lesson wasn't it? You are further into the world of Business Intelligence. This is where Excel is going. Get up to speed with this and you will be way ahead of everyone else.

I hope I have given you enough of a PowerPivot foundation so that you can continue learning and developing your skills.

Other Lessons

I have many other lessons covering various Excel topics.

You can find all of them on my website at: http://markmoorebooks.com/excel-lessons/

If this lesson has helped you, please take a few minutes and leave a review on Amazon. The more reviews the lesson gets, the easier other students will be able to find it.

Thank you!