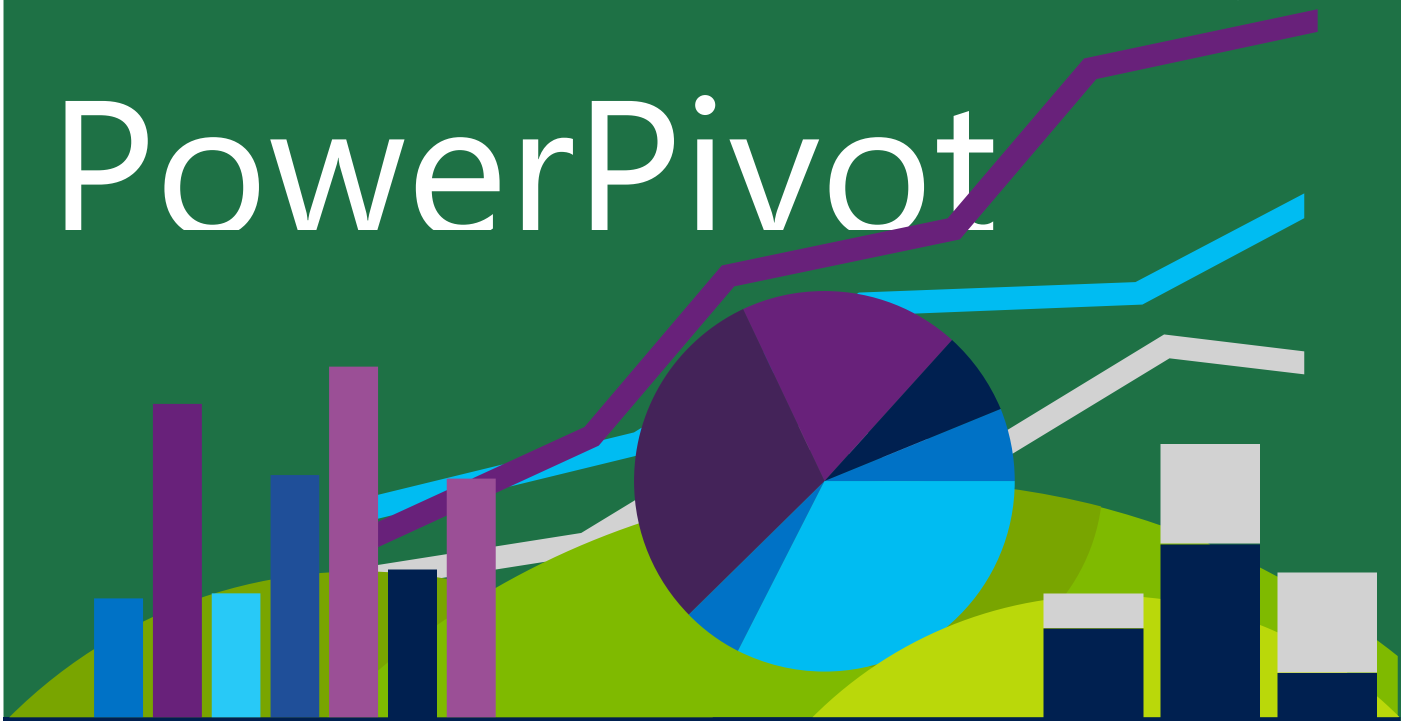
Microsoft Office Training Series

Excel

PowerPivot



- Courses never Cancelled
- 12+ Months Schedule
- 24 Months Online Support
- UK Wide Delivery

STL www.stl-training.co.uk

Learning & Performance Institute
Accredited Learning Provider

ilm Recognised Provider

Microsoft Partner
Certified Silver Partner

Ver2-2

Welcome to your Excel PowerPivot training course

1. Understand PowerPivot
2. Loading PowerPivot
3. How to create a data model to use with PowerPivot
4. Work with multiple data sources
5. Relate tables for PowerPivot analysis
6. Create measures (DAX functions)
7. Create KPI
8. Create dashboard reports use PivotTables and Pivot charts

Microsoft Office Training Series

Professional Development Series

Microsoft Technical Series

www.stl-training.co.uk/Feedback

Table of Contents

Unit 1 - When to Use PowerPivot, and How It Relates to Normal Pivot Usage

In this unit, you will learn how to:

- Understand the workflow for PowerPivot vs PivotTable

I hear this question a lot. Simply put, PowerPivot is useful in any situation where you would normally want to use a pivot. Whether you have 100 rows of data or 100 million, if you need to analyse or report on trends, patterns, and/or aggregates from that data, rather than the original rows of data themselves, chances are very good that PowerPivot has something to offer.

When you use a traditional (non Power-) pivot, your workflow in Excel generally looks something like this:

1. Grab data from one or more sources, typically landing in Excel worksheets (but sometimes directly in the “pivotcache” in advanced cases).
2. If multiple tables of data are involved, use VLOOKUP() or similar to create integrated single tables
3. Add calculated columns as needed
4. Build pivots against that data
5. Either use those pivots directly as the final report/analysis, or build separate report sheets which reference into the pivots using formulas

Our guiding philosophy on PowerPivot was “make it just like Excel wherever possible, and where it’s not possible, make it ‘rhyme’ very closely with Excel.” Accordingly, the 5-step workflow from above looks like this in PowerPivot:

1. Grab data from one or more sources, landing in worksheet-tables in the PowerPivot window.
2. Use relationships to quickly link multiple tables together, entirely bypassing VLOOKUP() or similar tedious formulas.
3. Optionally supplement that data with calculated columns and measures, using Excel functions you have always known, plus some powerful new ones.
4. Build pivots against that data.

5. Either use those pivots directly as the final report/analysis, or convert pivots into formulas with a single click for flexible layout, or you can still build separate report sheets which reference into the pivots using formulas.

Unit 2 – What is PowerPivot for Excel Add-In

In this unit, you will learn how to:

- Understand the PowerPivot add-in

Microsoft SQL Server PowerPivot for Excel is a new technology aimed at providing self-service Business Intelligence (BI). PowerPivot is a real revolution inside the world of data analysis because it gives you all the power you need to perform complex analysis of data without requiring the intervention of BI technicians. This tool, an Excel add-in, implements a powerful in-memory database that can organize data, detect interesting relationships, and give you a swift way to browse information.

Standard PivotTables are good tools. Nevertheless, to let you analyse more complex data, Microsoft introduced the so-called self-service BI. The goal of this technology is to let you build complex data structures and analyse them with pivot tables, removing the current limitations of the PivotTable. PowerPivot is the first tool available from Microsoft to handle self-service BI.

PowerPivot enables you to analyse data without needing to contact IT staff to produce complex queries. Further, it removes the limitation that a PivotTable can analyse only a single table. You want to be able to query more tables at the same time, producing reports that easily integrate information coming from different sources.

These are some of the most interesting features of PowerPivot:

- The ability to organize tables for the PivotTable tool in a relational way, freeing the analyst from the need to import data as Excel worksheets before analysing the data.
- The availability of a fast, space-saving columnar database that can handle huge amounts of data without the limitations of Excel worksheets.
- DAX, a powerful programming language that defines complex expressions on top of the relational database. DAX allows you to define surprisingly rich expressions, compared to those that are standard in Excel.

- The ability to integrate different sources and almost any kind of data, such as information from databases, Excel worksheets, and sources available on the Internet.
- Analyse millions and billions of rows of data.

Unit 3 - Loading the PowerPivot Add-In

In this unit, you will learn how to:

- Download & load the PowerPivot add-in in Excel 2010

Excel 2010

To install the PowerPivot add-in, you need to install the two following files:

1. Microsoft Visual Studio 2010 tools for Office Runtime
    - MS Visual Studio 2010 Tools for Office Runtime (x86). This is often required before PowerPivot will start (3Mb)

   http://www.microsoft.com/en-us/download/details.aspx?id=20479

2. Install PowerPivot from www.PowerPivot.com

When loading PowerPivot, there are times when things don't seem to work when you load the Com Add-In.

Here are two links to forums where people have been discussing these problems that may be useful to you.

Forum MSDN
http://social.msdn.microsoft.com/Forums/en-US/sqlkjpowerpivotforexcel/thread/572168c3-7faf-41a6-88b4-790a9cd223d0/

Forum powerpivotgeek
http://powerpivotgeek.com/2010/05/12/powerpivot-for-excel-add-in-is-not-coming-up-in-excel/comment-page-1/#comment-13212

Excel 2013

PowerPivot is integrated in Excel version 2013 and nothing needs to be installed

Unit 4 - Getting started with PowerPivot

In this unit, you will learn how to:

- Launch PowerPivot
- Navigate and understand the PowerPivot window

Once you have installed PowerPivot, you can then get on with the task of analysing millions of rows of data.

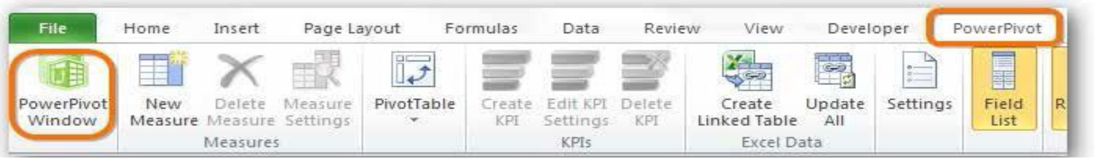
Start Excel & launch the PowerPivot

You now are ready to add the PowerPivot Tab and Ribbon. To achieve this, follow the instructions below:

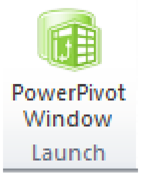
1. Go to File and then Options.
2. On the left-hand panel, select the Add-Ins option.
3. At the bottom of the screen, choose the Com Add-Ins from the Manage box and click OK.
4. Tick the checkbox for the PowerPivot for Excel.
5. The Ribbon will appear after the last Tab on the Ribbon.

Select the PowerPivot tab and you will see the following screen:

Excel 2010:

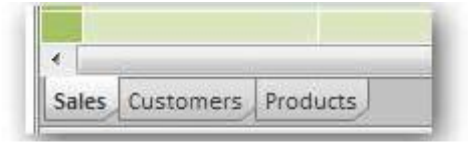


The PowerPivot Window

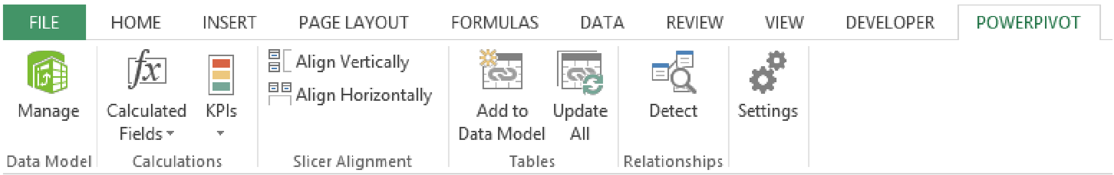


Click on the PowerPivot Button in the Launch group to open the window to connect to data sources.

Every table of data you load into PowerPivot gets its own sheet tab. So if you import three different tables of data, you will end up with something like this:

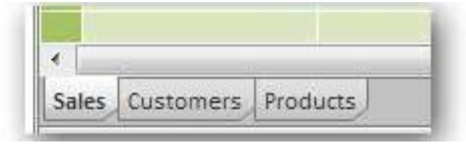


Excel 2013:



Click on the Manage Button in the Data Model group to open the window to connect to data sources.

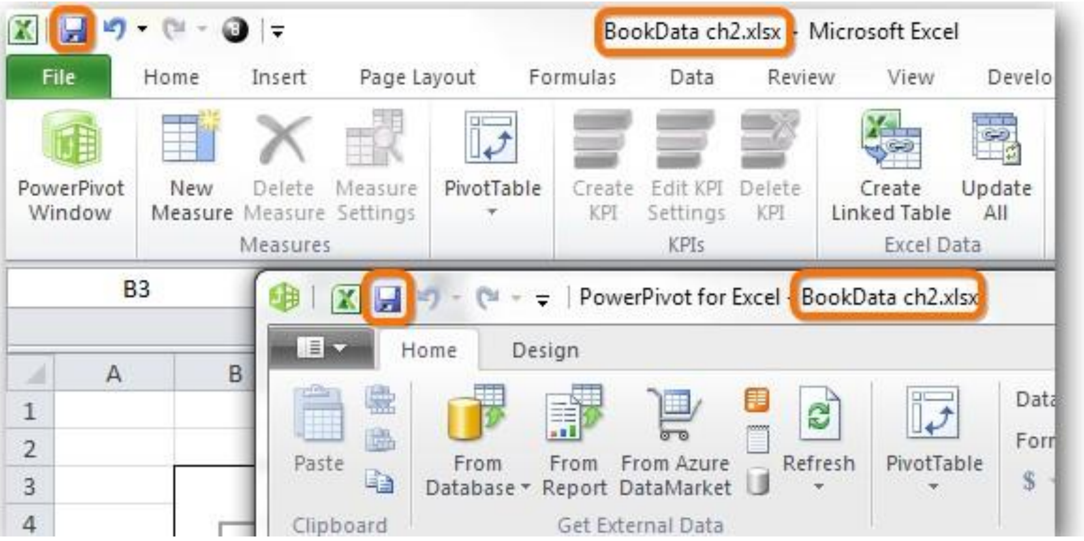
Every table of data you load into PowerPivot gets its own sheet tab. So if you import three different tables of data, you will end up with something like this:



Editing Cells in the PowerPivot window?

The PowerPivot sheets are read-only. You can't just select a cell and start typing. You can delete or rename entire sheet tabs and columns, and you can add calculated columns, but you cannot modify cells of data, ever.

Does that sound bad? Actually, it's a good thing. It makes the data more trustworthy, but even more im- portantly, it forces you to do things in a way that saves you a lot of time later. More on this in subsequent chapters.



The PowerPivot window is linked to the workbook so no need to carry two workbooks around with you.

Unit 5 – Adding Data to PowerPivot

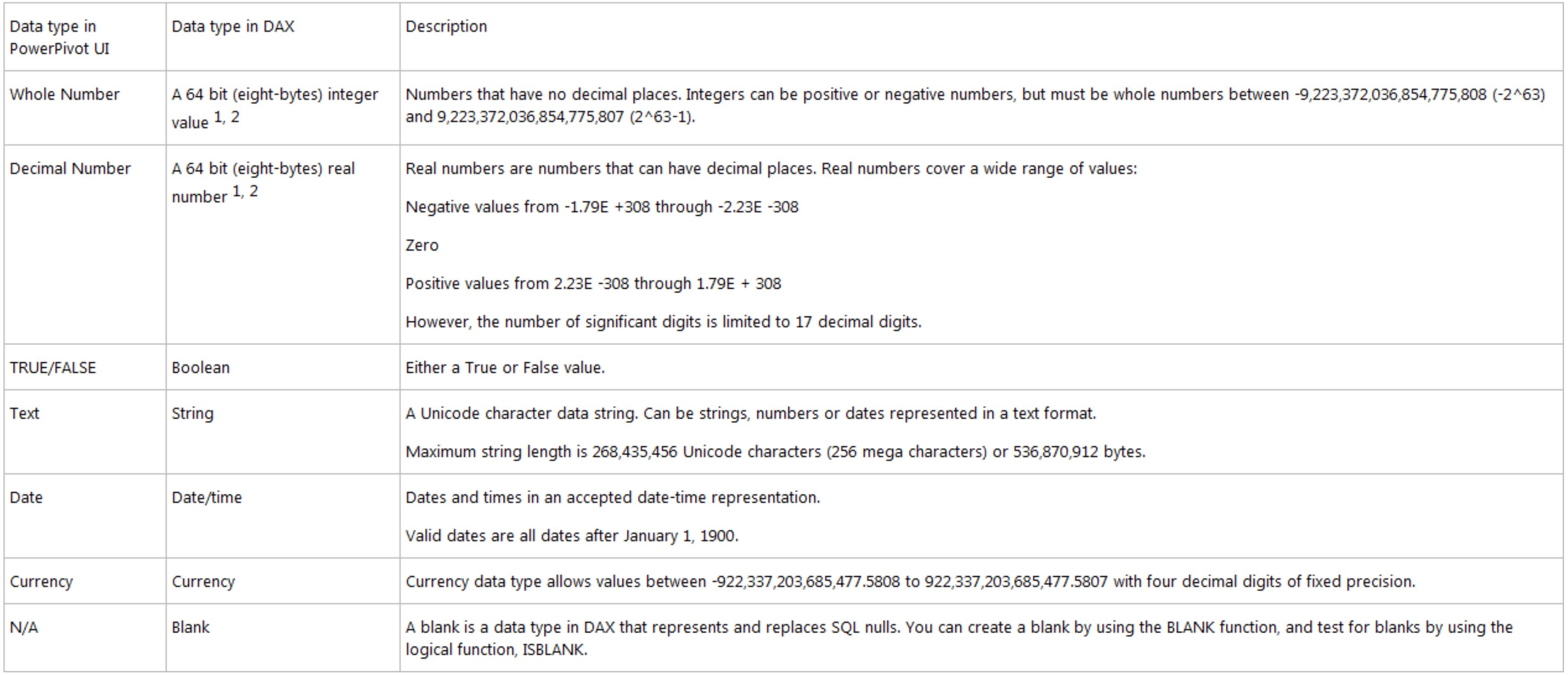
In this unit, you will learn how to:

- Add data to PowerPivot
- Link data to PowerPivot
- Paste data to PowerPivot

Data Sources and types supported in PowerPivot workbooks

Data Types

(Please use this link to see additional information - http://msdn.microsoft.com/en-us/library/gg413463.aspx



Data type in PowerPivot UI	Data type in DAX	Description
Whole Number	A 64 bit (eight-bytes) integer value [1, 2]	Numbers that have no decimal places. Integers can be positive or negative numbers, but must be whole numbers between -9,223,372,036,854,775,808 (-2^63) and 9,223,372,036,854,775,807 (2^63-1).
Decimal Number	A 64 bit (eight-bytes) real number [1, 2]	Real numbers are numbers that can have decimal places. Real numbers cover a wide range of values: Negative values from -1.79E +308 through -2.23E -308 Zero Positive values from 2.23E -308 through 1.79E + 308 However, the number of significant digits is limited to 17 decimal digits.
TRUE/FALSE	Boolean	Either a True or False value.
Text	String	A Unicode character data string. Can be strings, numbers or dates represented in a text format. Maximum string length is 268,435,456 Unicode characters (256 mega characters) or 536,870,912 bytes.
Date	Date/time	Dates and times in an accepted date-time representation. Valid dates are all dates after January 1, 1900.
Currency	Currency	Currency data type allows values between -922,337,203,685,477.5808 to 922,337,203,685,477.5807 with four decimal digits of fixed precision.
N/A	Blank	A blank is a data type in DAX that represents and replaces SQL nulls. You can create a blank by using the BLANK function, and test for blanks by using the logical function, ISBLANK.

Data Sources

Here is a sample list of Data sources to connect to:

- Access databases
- SQL Server relational databases
- SQL Server Parallel Data Warehouse (PDW)
- Oracle relational databases
- Teradata relational databases
- IBM DB2 relational databases
- Sybase relational databases
- Text files
- Excel files
- Data Feeds
- Office Database Connection files

Importing Data (Pre-prepared file)

To import data from Access or SQL database follow these steps:

1. Click on the Home tab.
2. Choose Get External Data from Database.
3. Select from Access.
4. Click on 'From Access'.

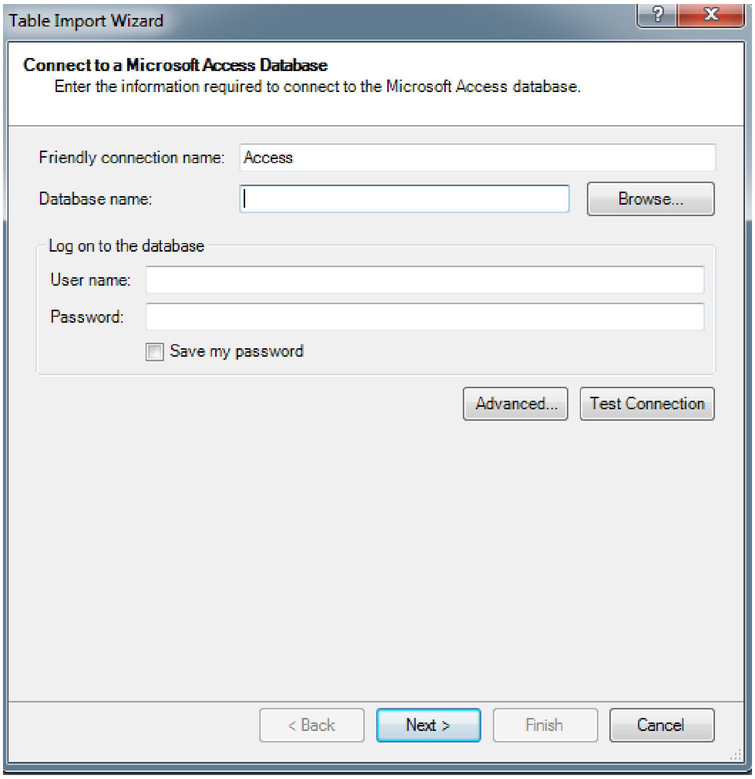


Table Import Wizard

Connect to a Microsoft Access Database

Enter the information required to connect to the Microsoft Access database.

Friendly connection name: Access

Database name:

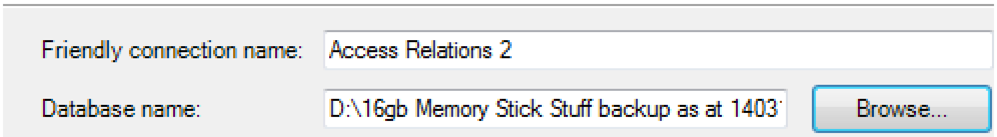
Log on to the database

User name:

Password:

☐ Save my password

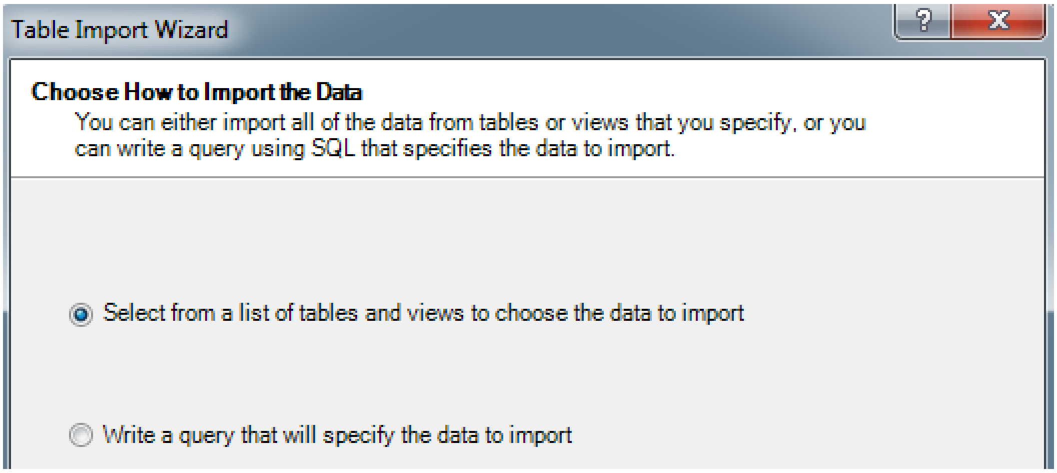
5. Click the Browse button to navigate and select the database you wish to connect to. Select the database and click the 'Open' button.



Friendly connection name: Access Relations 2

Database name: D:\16gb Memory Stick Stuff backup as at 1403

6. Click the ‘Next’ button.



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

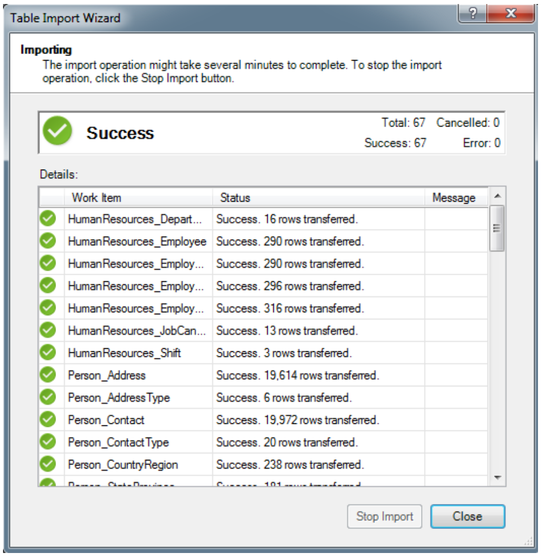
You can now choose which tables you want to import or create a query that combines information from all tables to give you specific data to analyse.

7. Click the ‘Next’ button to select from the list of tables

8. Select the Tables you want to use by clicking on the appropriate checkboxes.

9. Click Finish

10. You will then be presented with the following confirmation screen.



Importing

The import operation might take several minutes to complete. To stop the import operation, click the Stop Import button.

Success

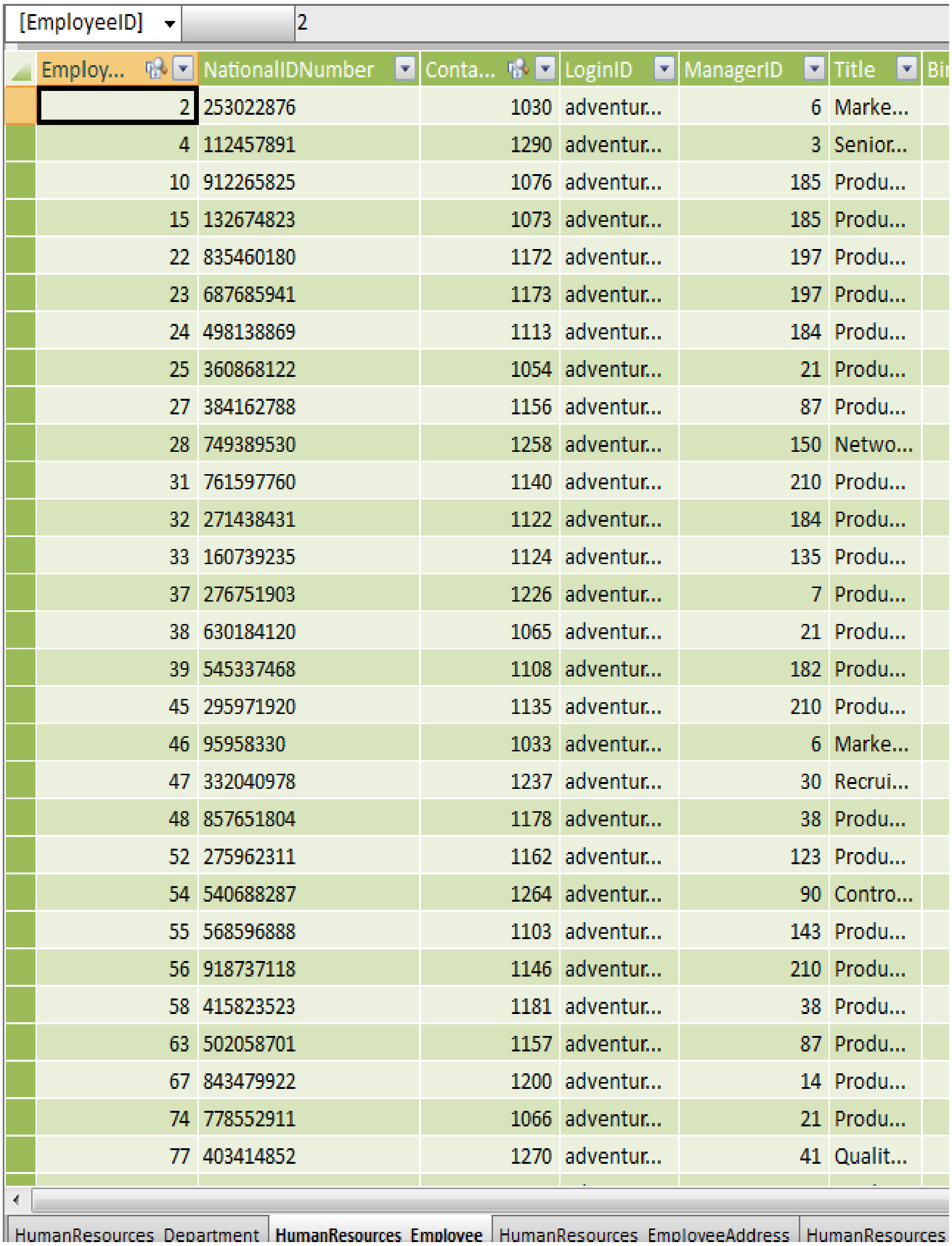
Total: 67 Cancelled: 0

Success: 67 Error: 0

Details:

	Work Item	Status	Message
✓	HumanResources_Depart...	Success. 16 rows transferred.	
✓	HumanResources_Employee	Success. 290 rows transferred.	
✓	HumanResources_Employ...	Success. 290 rows transferred.	
✓	HumanResources_Employ...	Success. 296 rows transferred.	
✓	HumanResources_Employ...	Success. 316 rows transferred.	
✓	HumanResources_JobCan...	Success. 13 rows transferred.	
✓	HumanResources_Shift	Success. 3 rows transferred.	
✓	Person_Address	Success. 19,614 rows transferred.	
✓	Person_AddressType	Success. 6 rows transferred.	
✓	Person_Contact	Success. 19,972 rows transferred.	
✓	Person_ContactType	Success. 20 rows transferred.	
✓	Person_CountryRegion	Success. 238 rows transferred.	

Once the Close button has been clicked, each table you selected will appear as a separate tab at the bottom of the PowerPivot window ready for analysis.



[EmployeeID] 2

Employ...	NationalIDNumber	Conta...	LoginID	ManagerID	Title	Bi
2	253022876	1030	adventur...	6	Marke...	
4	112457891	1290	adventur...	3	Senior...	
10	912265825	1076	adventur...	185	Produ...	
15	132674823	1073	adventur...	185	Produ...	
22	835460180	1172	adventur...	197	Produ...	
23	687685941	1173	adventur...	197	Produ...	
24	498138869	1113	adventur...	184	Produ...	
25	360868122	1054	adventur...	21	Produ...	
27	384162788	1156	adventur...	87	Produ...	
28	749389530	1258	adventur...	150	Netwo...	
31	761597760	1140	adventur...	210	Produ...	
32	271438431	1122	adventur...	184	Produ...	
33	160739235	1124	adventur...	135	Produ...	
37	276751903	1226	adventur...	7	Produ...	
38	630184120	1065	adventur...	21	Produ...	
39	545337468	1108	adventur...	182	Produ...	
45	295971920	1135	adventur...	210	Produ...	
46	95958330	1033	adventur...	6	Marke...	
47	332040978	1237	adventur...	30	Recrui...	
48	857651804	1178	adventur...	38	Produ...	
52	275962311	1162	adventur...	123	Produ...	
54	540688287	1264	adventur...	90	Contro...	
55	568596888	1103	adventur...	143	Produ...	
56	918737118	1146	adventur...	210	Produ...	
58	415823523	1181	adventur...	38	Produ...	
63	502058701	1157	adventur...	87	Produ...	
67	843479922	1200	adventur...	14	Produ...	
74	778552911	1066	adventur...	21	Produ...	
77	403414852	1270	adventur...	41	Qualit...	

Each Table that was selected appears as a tab at the bottom of the window. Click on each tab to see the different tables of data.

Add Data by Using Excel Linked Tables

A Linked Table is a table that has been created in Excel but is linked to a table in the PowerPivot window. The advantage of creating and maintaining the data in Excel, rather than importing it, is that you can continue to edit the values in the Excel worksheet, while using the data for analysis in PowerPivot.

In the PowerPivot window click on the File menu and choose 'Get External Data from Other Sources' and it will open the following window as displayed below:

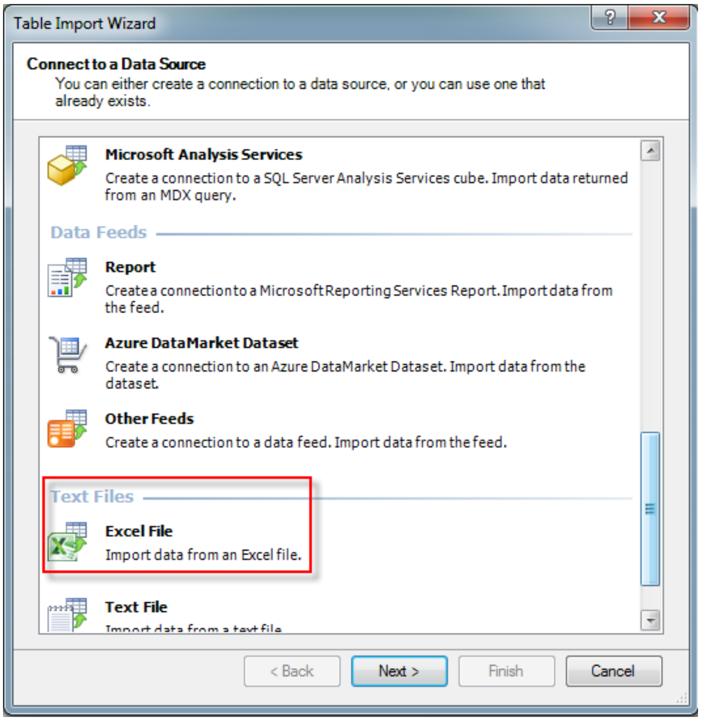


Table Import Wizard

Connect to a Data Source

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

Microsoft Analysis Services
Create a connection to a SQL Server Analysis Services cube. Import data returned from an MDX query.

Data Feeds

Report
Create a connection to a Microsoft Reporting Services Report. Import data from the feed.

Azure DataMarket Dataset
Create a connection to an Azure DataMarket Dataset. Import data from the dataset.

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

Select the Excel File option and click Next.

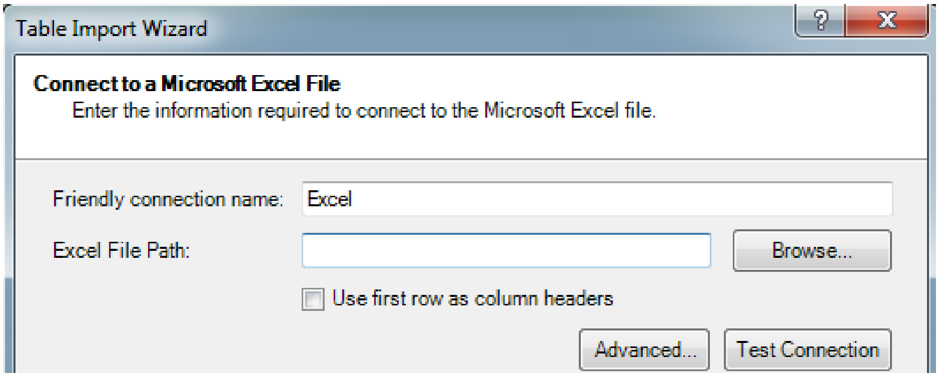


Table Import Wizard

Connect to a Microsoft Excel File

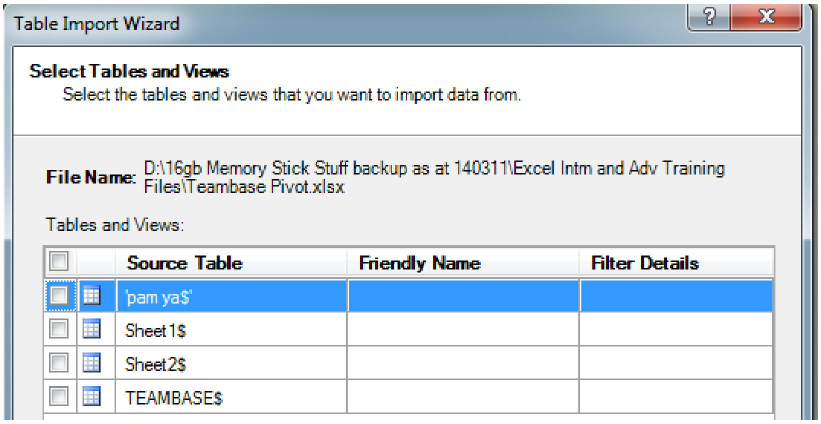
Enter the information required to connect to the Microsoft Excel file.

Friendly connection name: Excel

Excel File Path:

☐ Use first row as column headers

Click the Browse button to navigate to the Excel file and then choose the worksheets you want to link to and then each sheet will appear as a tab



Select Tables and Views

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

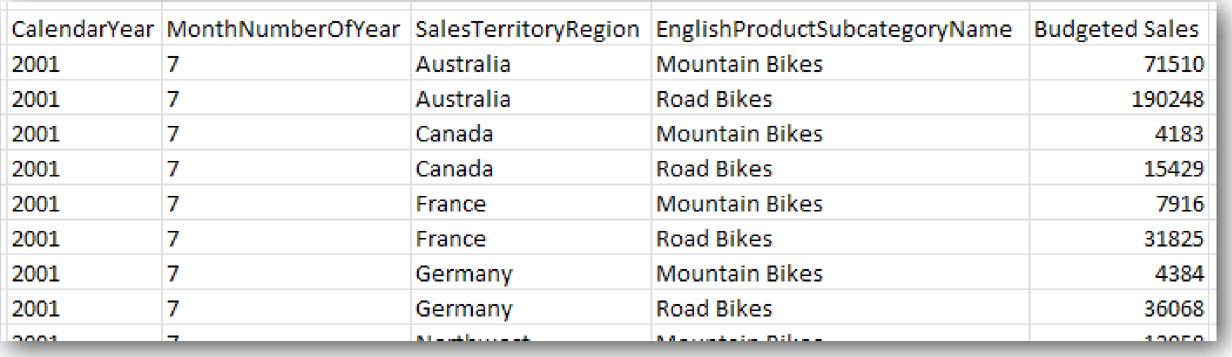
File Name: D:\16gb Memory Stick Stuff backup as at 140311\Excel Intm and Adv Training Files\Teambase Pivot.xlsx

Tables and Views:

☐		Source Table	Friendly Name	Filter Details
☐	▦	'pam ya$'		
☐	▦	Sheet1$		
☐	▦	Sheet2$		
☐	▦	TEAMBASE$		

Linked Tables (Data Source Type)

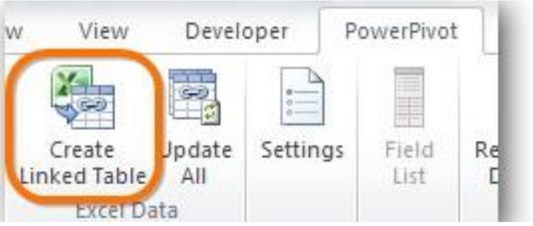
If you have a table of data in Excel like this:



CalendarYear	MonthNumberOfYear	SalesTerritoryRegion	EnglishProductSubcategoryName	Budgeted Sales
2001	7	Australia	Mountain Bikes	71510
2001	7	Australia	Road Bikes	190248
2001	7	Canada	Mountain Bikes	4183
2001	7	Canada	Road Bikes	15429
2001	7	France	Mountain Bikes	7916
2001	7	France	Road Bikes	31825
2001	7	Germany	Mountain Bikes	4384
2001	7	Germany	Road Bikes	36068

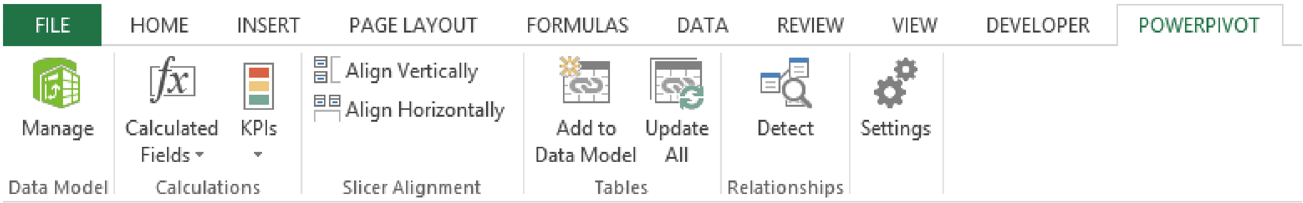
Excel 2010

You can quickly grab it into PowerPivot by using the “Create Linked Table” button on the PowerPivot ribbon tab:



Excel 2013

“Add to Data Model” button on the PowerPivot ribbon tab:



Advantages

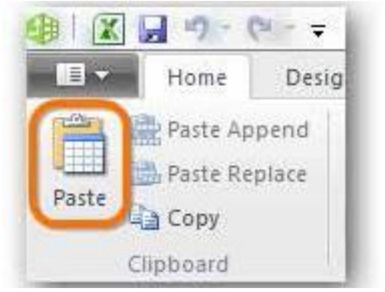
- This is the quickest way to get a table from Excel into PowerPivot.
- If you edit the data in Excel – change cells, add rows, etc. – PowerPivot will pick those changes up. So, this is a sneaky way to work around the "cannot edit in PowerPivot window" limitation.
- If you add columns, those will *also* be picked up. I call this out specifically because Copy/Paste (be- low) does *not* do this, and I frequently find myself wishing I had used Link rather than Copy/Paste for that reason.

Limitations

- You cannot link a table in Workbook A to the PowerPivot window from Workbook B. This only creates a linked table in the PowerPivot window "tied" to the XLSX where the table currently resides.
- This is *not* a good way to load *large* amounts of data into PowerPivot. A couple of thousand rows is fine, but ten thousand rows or more may cause you trouble and grind your computer to a halt.
- By default, PowerPivot will update its copy of this table *every* time you leave the PowerPivot window and come back to it. That happens whether you changed anything in Excel or not, and leads to a delay while PowerPivot reloads the same data.
- Linked Tables cannot be scheduled for auto-refresh on a PowerPivot server. They can only be up- dated on the desktop. (This is true for PowerPivot v1 and v2.
- You cannot subsequently change over to a different source type – this really isn't a limitation specifically for linked tables. This is true of every source type in this list: whatever type of data source is used to create a table, that table cannot later be changed over to use another type of data source. So, if you create a PowerPivot table via Linked Table, you cannot change it in the future to be sourced from a text file, database, or any other source. You will need to delete the table and re-create it from the new source.

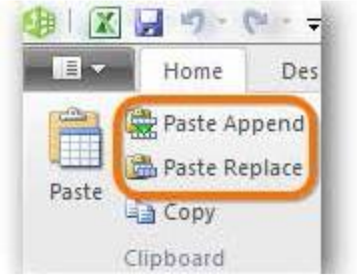
Pasting Data into PowerPivot (Data Source Type)

If you copy a table-shaped batch of data onto the Windows clipboard, this button in the PowerPivot window will light up:



Advantages

- You can paste from any table-shaped source and are not limited to using just Excel (unlike Linked Tables)
- You can paste from other workbooks and are not limited to the same workbook as your PowerPivot window
- Pasted tables support both “Paste/Replace” and “Paste/Append” as shown by the buttons below:

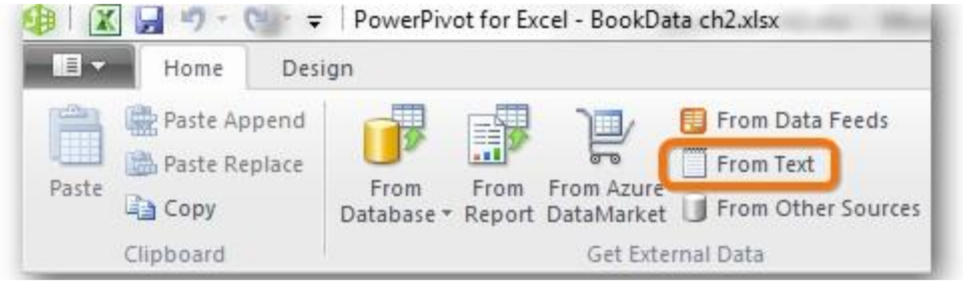


Limitations

- Suffers from the same “large data set” drawback as Linked Tables.
- You can never paste in an additional column. Once a table has been pasted, its columns are fixed.

- You can add a calculated column but can never change your mind and add that column you thought you omitted the first time you pasted. This becomes more of a drawback than you might expect.
- Not all apparently table-shaped sources are truly table-shaped. Tables on web pages are notorious for this. Sometimes you are lucky and sometimes you are not.
- Cannot be switched to another data source type (true of all data source types).

Importing from Text Files (Data Source Type)



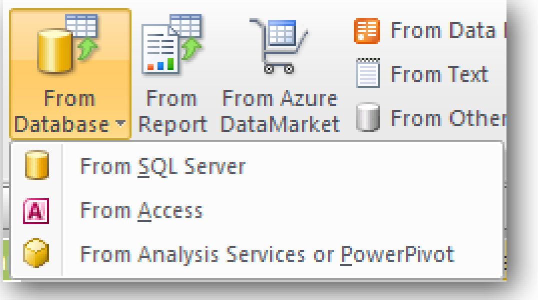
Advantages

- Can handle nearly limitless data volumes
- You can add new columns later (if you are a little careful about it, see below)
- Text files can be located anywhere on your hard drive or even on network drives (but not on web- sites, at least not in my experience). So, some backend process might update a text file every night in a fixed location (and filename), for example, and all you have to do is refresh the PowerPivot work- book the next day to pick up the new data.
- Can be switched to point at a different text file, but still cannot be switched to an entirely different source type (like database).

Limitations

- No reliable column names – unlike in a database, text files are not robust with regards to column names. If the order of columns in a CSV file gets changed, that will likely confuse PowerPivot on the next refresh.
- Cannot be switched to another data source type (true of all data source types).

Databases (Data Source Type)



Advantages

- Can handle nearly limitless data volumes.
- You can add new columns later.
- Can be switched to point at a different server, database, table, view, or query. Lots of "re-pointability" here, but you still can't switch to another data source type.
- Databases are a great place to add calculated columns. There are some significant advantages to building calculated columns in the database, and then importing them, rather than writing the calculated columns in PowerPivot itself. This is particularly true when your tables are quite large. We will talk about this later in this chapter.
- PowerPivot really shines when paired with a good database. There is just an incredible amount of flexibility available when your data is coming from a database.

Limitations

- Not always an option. Hey, not everyone has a SQL Server at their disposal, and/or not everyone knows how to work with databases.
- Cannot switch between database types. A table sourced from Access cannot later be switched over and pointed to SQL Server. So, in reality, these are separate data source types, but they are similar enough that I did not want to add a completely separate section for each.
- Cannot be switched to another data source type (true of all data source types).

Less Common Data Source Types

SharePoint Lists

These are great when you have a data source that is maintained and edited by human beings, especially if more than one person shares that editing duty. But if your company does not use SharePoint, this isn't terribly relevant to you.

Reporting Services (SSRS) Reports

This is another example of "if your company already uses it, it's a great data source," but otherwise, not relevant.

Data Feeds

Data Feeds are essentially a way in which a programmer can easily write an "adapter" that makes a particular data source available such that PowerPivot can pull data from it.

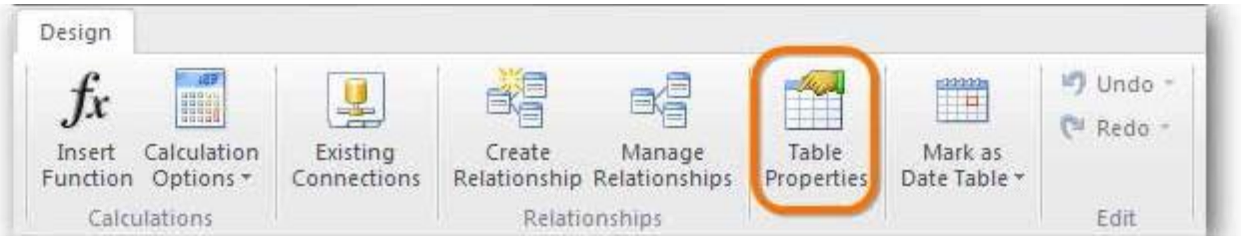
In fact, SharePoint and SSRS (and maybe DataMarket too) are exposed to PowerPivot via the Data Feed protocol – that is how that source types were enabled "under the hood."

So, I am mentioning this here in case your company has some sort of custom internal server application and you want to expose its data to PowerPivot. The quickest way to do that may be to expose that application's data as a data feed, as long as you have a programmer available to do the work.

Ribbon Buttons

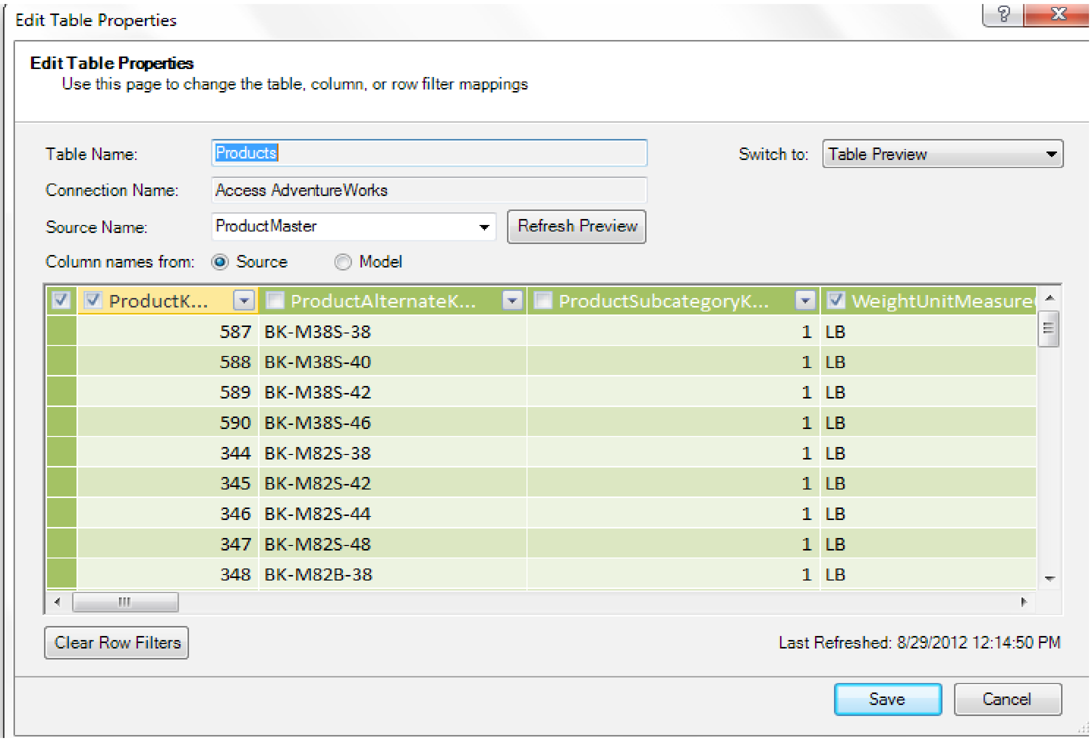
Table Properties Button

This is a very important button, but it is hiding on the second ribbon tab in the PowerPivot window:



This button is what allows you to modify the query behind an existing table. So, it's going be pretty important to you at some point. I know someone who used PowerPivot for two months before realizing that there was a second ribbon tab!

When you click it, it returns you to one of the dialogs you saw in the original import sequence:



Edit Table Properties

Edit Table Properties

Use this page to change the table, column, or row filter mappings

Table Name: Products

Connection Name: Access AdventureWorks

Source Name: ProductMaster

Switch to: Table Preview

Column names from: ● Source ○ Model

☑	☑ ProductK...	☐ ProductAlternateK...	☐ ProductSubcategoryK...	☑ WeightUnitMeasure
	587	BK-M38S-38	1	LB
	588	BK-M38S-40	1	LB
	589	BK-M38S-42	1	LB
	590	BK-M38S-46	1	LB
	344	BK-M82S-38	1	LB
	345	BK-M82S-42	1	LB
	346	BK-M82S-44	1	LB
	347	BK-M82S-48	1	LB
	348	BK-M82B-38	1	LB

Last Refreshed: 8/29/2012 12:14:50 PM

This dialog is important for two reasons:

1. The Edit button lets you modify existing connections. In the screenshot above, you see a path to an Access database. If I want to point to a different Access database, I would click Edit here. Same thing if I want to point to a different text file, or if I want to point to a different SQL Server database, etc.

2. The Open button lets you quickly import a new table from that existing connection. I highly recommend doing this rather than starting over from the "From Database" button on the first ribbon tab. You get to skip the first few screens of the wizard this way, AND you don't litter your workbook with a million connections pointing to the same exact source.

Unit 6 - Preparing data for analysis

In this unit, you will learn how to:

- Filter data before import
- Work with table & column options
- Create relationships

Filtering the data before it appears in the PowerPivot window

The data source you are connecting to can be filtered before analysis in the PowerPivot window using the Preview & Filter button after selecting the table in this case.

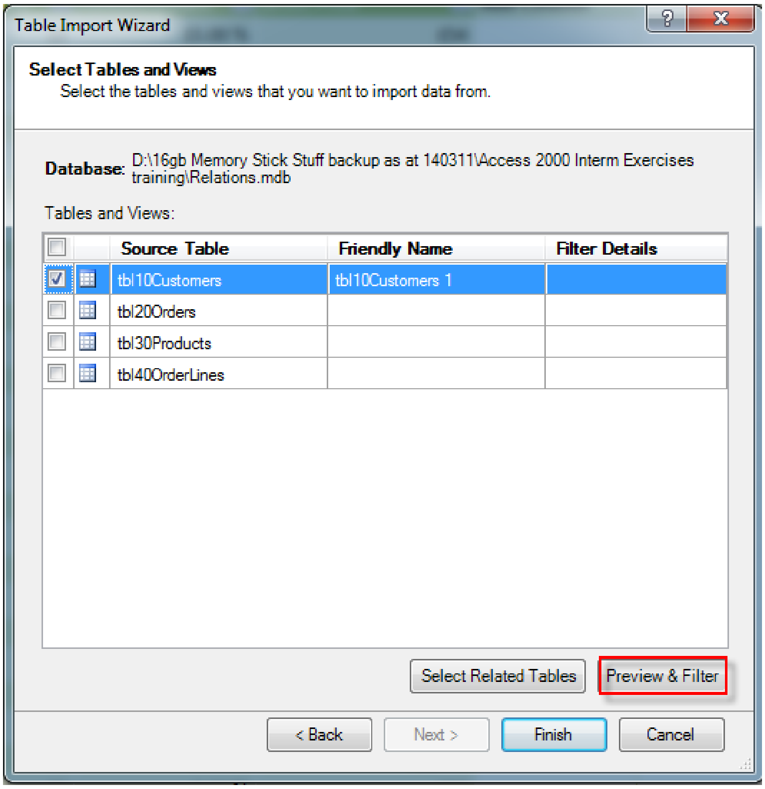


Table Import Wizard

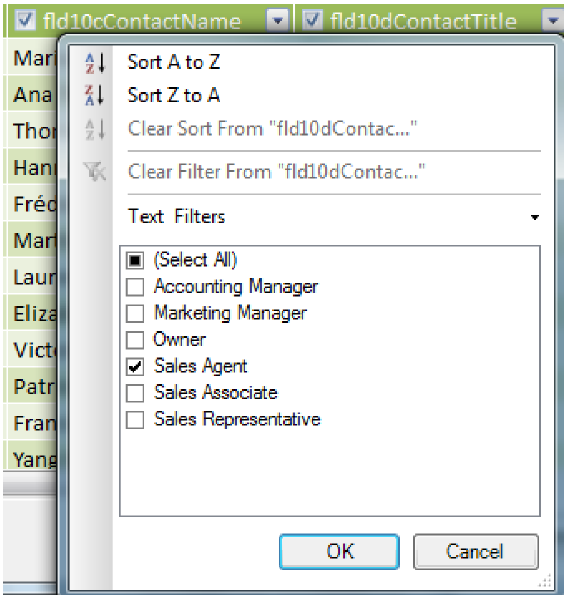
Select Tables and Views
Select the tables and views that you want to import data from.

Database: D:\16gb Memory Stick Stuff backup as at 140311\Access 2000 Interm Exercises training\Relations.mdb

Tables and Views:

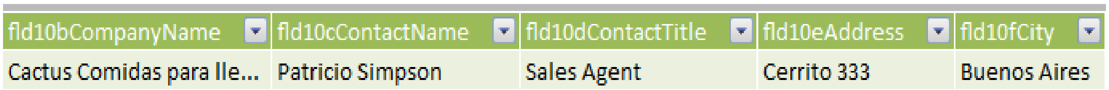
☐		Source Table	Friendly Name	Filter Details
☑		tbl10Customers	tbl10Customers 1	
☐		tbl20Orders		
☐		tbl30Products		
☐		tbl40OrderLines		

A new window appears with Filter buttons at the end of each field heading which can be used to extract data from a specific column or columns based on criteria you select. The Sales Agent Contact title has been chosen.



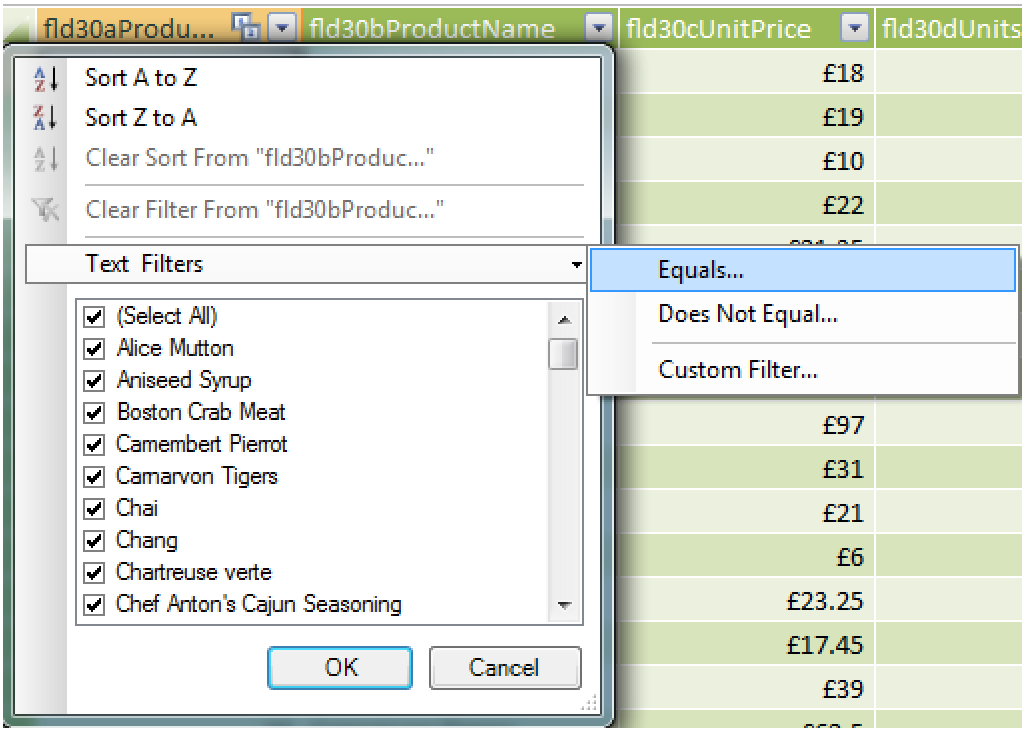
- ■ (Select All)
- ☐ Accounting Manager
- ☐ Marketing Manager
- ☐ Owner
- ☑ Sales Agent
- ☐ Sales Associate
- ☐ Sales Representative

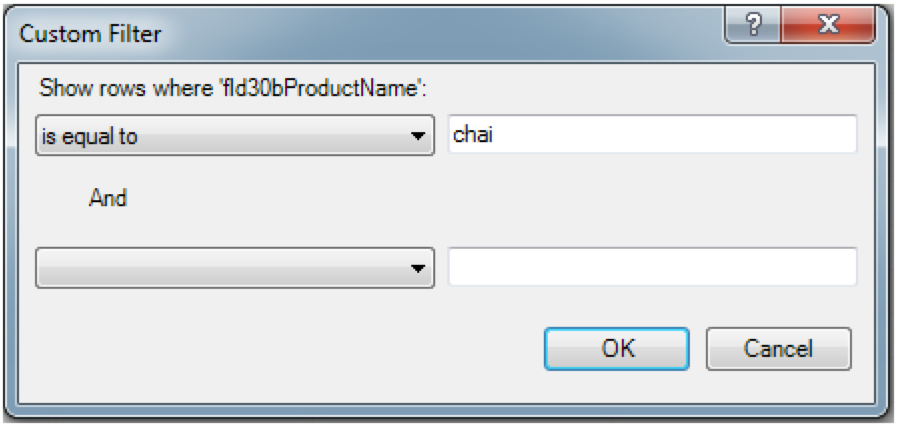
A new table is generated showing the record to the Sales Agent as shown below.



fld10bCompanyName	fld10cContactName	fld10dContactTitle	fld10eAddress	fld10fCity
Cactus Comidas para lle...	Patricio Simpson	Sales Agent	Cerrito 333	Buenos Aires

A text filter could also be used to extract data. In this example, a Text Filter will be used to extract the Chai Product record.



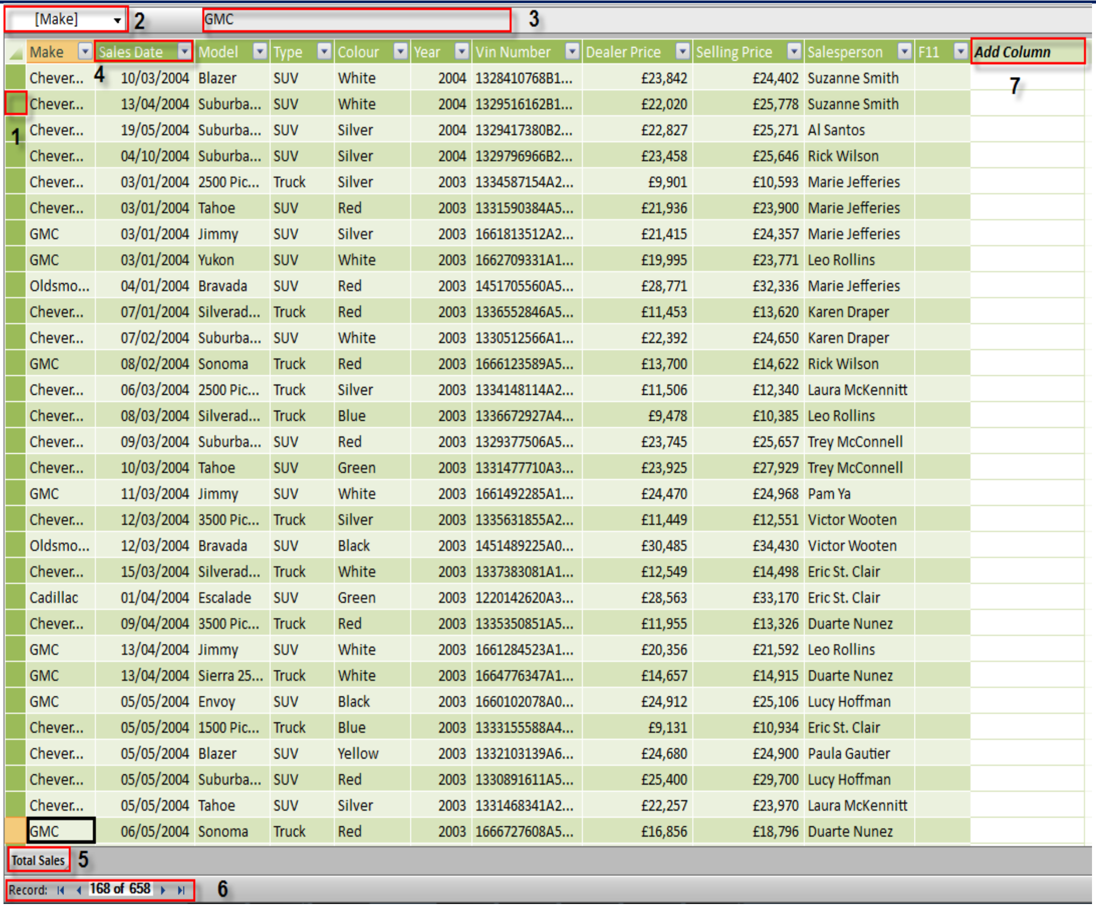


Show rows where 'fld30bProductName':

is equal to: chai

Use the Check boxes to select the Items you need. First unselect Select All and then click the appropriate checkboxes. Alternatively use the Text Filters for more customised filters and then click ok to apply the filter

Working with Tables and Columns



Make	Sales Date	Model	Type	Colour	Year	Vin Number	Dealer Price	Selling Price	Salesperson	F11	Add Column
Chever...	10/03/2004	Blazer	SUV	White	2004	1328410768B1...	£23,842	£24,402	Suzanne Smith		
Chever...	13/04/2004	Suburba...	SUV	White	2004	1329516162B1...	£22,020	£25,778	Suzanne Smith		
Chever...	19/05/2004	Suburba...	SUV	Silver	2004	1329417380B2...	£22,827	£25,271	Al Santos		
Chever...	04/10/2004	Suburba...	SUV	Silver	2004	1329796966B2...	£23,458	£25,646	Rick Wilson		
Chever...	03/01/2004	2500 Pic...	Truck	Silver	2003	1334587154A2...	£9,901	£10,593	Marie Jefferies		
Chever...	03/01/2004	Tahoe	SUV	Red	2003	1331590384A5...	£21,936	£23,900	Marie Jefferies		
GMC	03/01/2004	Jimmy	SUV	Silver	2003	1661813512A2...	£21,415	£24,357	Marie Jefferies		
GMC	03/01/2004	Yukon	SUV	White	2003	1662709331A1...	£19,995	£23,771	Leo Rollins		
Oldsmo...	04/01/2004	Bravada	SUV	Red	2003	1451705560A5...	£28,771	£32,336	Marie Jefferies		
Chever...	07/01/2004	Silverad...	Truck	Red	2003	1336552846A5...	£11,453	£13,620	Karen Draper		
Chever...	07/02/2004	Suburba...	SUV	White	2003	1330512566A1...	£22,392	£24,650	Karen Draper		
GMC	08/02/2004	Sonoma	Truck	Red	2003	1666123589A5...	£13,700	£14,622	Rick Wilson		
Chever...	06/03/2004	2500 Pic...	Truck	Silver	2003	1334148114A2...	£11,506	£12,340	Laura McKennitt		
Chever...	08/03/2004	Silverad...	Truck	Blue	2003	1336672927A4...	£9,478	£10,385	Leo Rollins		
Chever...	09/03/2004	Suburba...	SUV	Red	2003	1329377506A5...	£23,745	£25,657	Trey McConnell		
Chever...	10/03/2004	Tahoe	SUV	Green	2003	1331477710A3...	£23,925	£27,929	Trey McConnell		
GMC	11/03/2004	Jimmy	SUV	White	2003	1661492285A1...	£24,470	£24,968	Pam Ya		
Chever...	12/03/2004	3500 Pic...	Truck	Silver	2003	1335631855A2...	£11,449	£12,551	Victor Wooten		
Oldsmo...	12/03/2004	Bravada	SUV	Black	2003	1451489225A0...	£30,485	£34,430	Victor Wooten		
Chever...	15/03/2004	Silverad...	Truck	White	2003	1337383081A1...	£12,549	£14,498	Eric St. Clair		
Cadillac	01/04/2004	Escalade	SUV	Green	2003	1220142620A3...	£28,563	£33,170	Eric St. Clair		
Chever...	09/04/2004	3500 Pic...	Truck	Red	2003	1335350851A5...	£11,955	£13,326	Duarte Nunez		
GMC	13/04/2004	Jimmy	SUV	White	2003	1661284523A1...	£20,356	£21,592	Leo Rollins		
GMC	13/04/2004	Sierra 25...	Truck	White	2003	1664776347A1...	£14,657	£14,915	Duarte Nunez		
GMC	05/05/2004	Envoy	SUV	Black	2003	1660102078A0...	£24,912	£25,106	Lucy Hoffman		
Chever...	05/05/2004	1500 Pic...	Truck	Blue	2003	1333155588A4...	£9,131	£10,934	Eric St. Clair		
Chever...	05/05/2004	Blazer	SUV	Yellow	2003	1332103139A6...	£24,680	£24,900	Paula Gautier		
Chever...	05/05/2004	Suburba...	SUV	Red	2003	1330891611A5...	£25,400	£29,700	Lucy Hoffman		
Chever...	05/05/2004	Tahoe	SUV	Silver	2003	1331468341A2...	£22,257	£23,970	Laura McKennitt		
GMC	06/05/2004	Sonoma	Truck	Red	2003	1666727608A5...	£16,856	£18,796	Duarte Nunez		

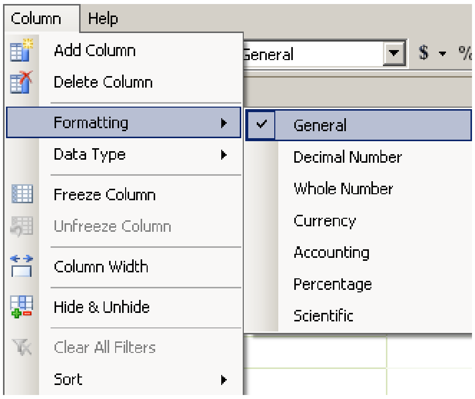
Once the data has imported into the PowerPivot window as mentioned earlier each worksheet or database table is displayed as a separate tab. The Tables has the following features and navigation tools:

1. Row Selector button used to select a record.
2. Column Selector button used to select a column.
3. Entry bar used to add new data to a column or edit data in the new column.
4. Column heading that incoprates Filtering and Sorting.
5. Sheet tab representing the Database table or linked excel worksheet.
6. Record selector tools.
7. Add a new column of data.

To navigate the table, you can you the following keyboard techniques:

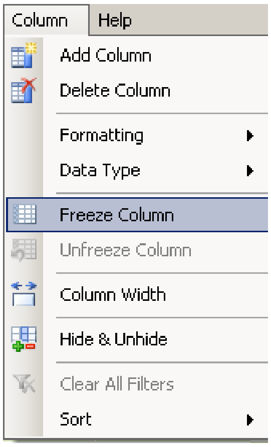
- Tab key to move across data in a row.
- Enter key to move down to the next data value in the column.
- Use the four arrow keys.
- Page Up and Page Down buttons to move to the next screen of results.

Formatting options



The Column menu allows you to apply or change a Data Type to an existing or new field. It also allows you to format the data to give it more meaning.

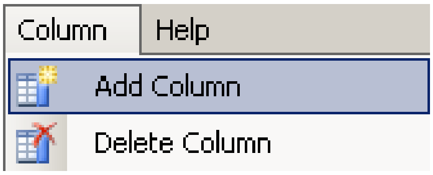
Freeze Columns

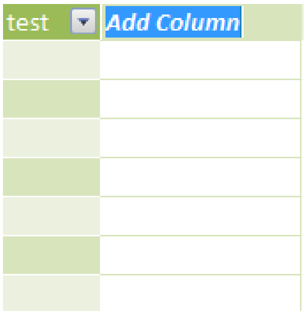


The Column menu allows you to Freeze the first column. However if you select the first two columns, then those selected columns will be frozen which is very useful when navigating further across a row of data.

Adding and Renaming Columns

There are two ways of adding columns. The first method is to click on the last column which is always ‘Add a new column’. Right click over the header and rename or double click on the header to rename the new column.



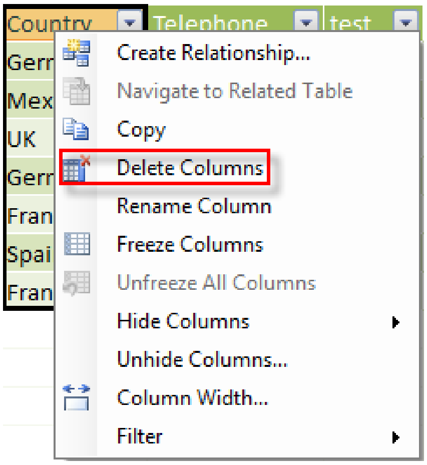


The second method is to click the Add Column button on the Column menu. The column is added at the end. This is the same as the first method as there will always be a blank column available. To rename the column, you can also double click on the column heading.

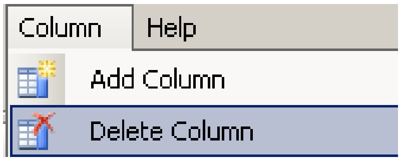
Other options available by right clicking are, Create Relationships, Copy, Delete Column, Rename Column, Freeze, Hide Columns, Column Widths and Filter.

Delete Columns

There are two methods to delete columns. The first method is to right click over the column and choose Delete Column.

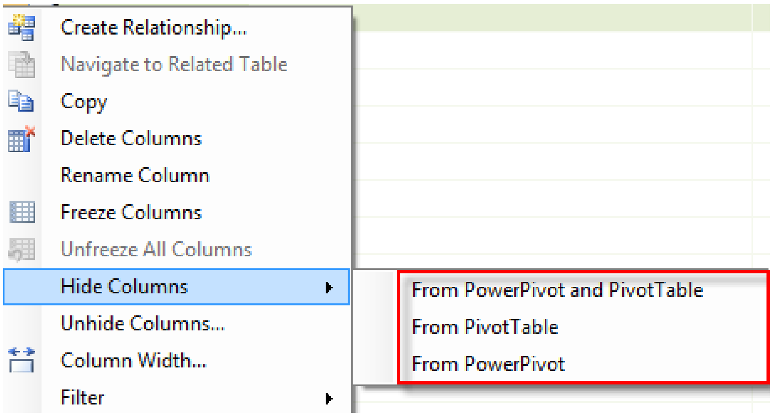


The second method is to click the Delete button on the Column menu.



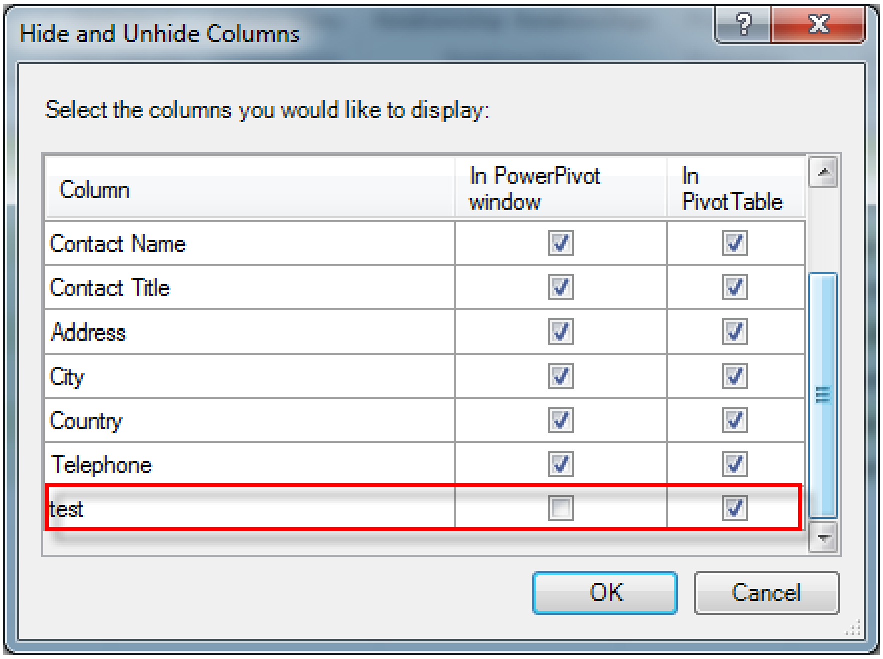
Hide and Unhide Columns

To hide columns, right click over the column and choose Hide Columns. You will be presented with these options:



Choose your option from the list.

To show the columns again, right click and choose Unhide Columns. Select the column from the list and click the appropriate checkbox as shown below:


Hide and Unhide Columns

Select the columns you would like to display:

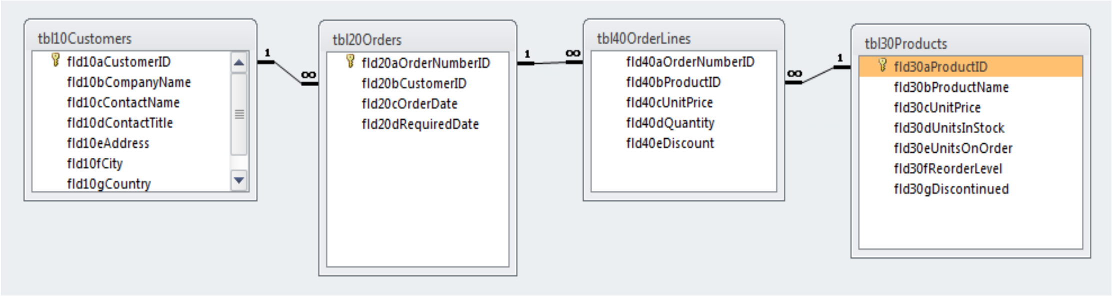
Column	In PowerPivot window	In PivotTable
Contact Name	☑	☑
Contact Title	☑	☑
Address	☑	☑
City	☑	☑
Country	☑	☑
Telephone	☑	☑
test	☐	☑

In Excel, there are fewer options to hide columns. The column can only be hidden from client's Tools.

Introduction to Relationships between Tables

Data Models

These two scenarios demonstrate that the Excel way does not need a data model whereas the PowerPivot way creates a data model. But one question is still open: what is a data model? A Data Model is nothing but a list of tables (sometimes referred to as entities) with arrows connecting them. A table, as its name suggests, is a list of columns that hold the real data. The arrows represent the existence of relationships between tables and are normally read as refers to. So, we might say that an order line refers to a product, meaning that the order line is about a specific product.



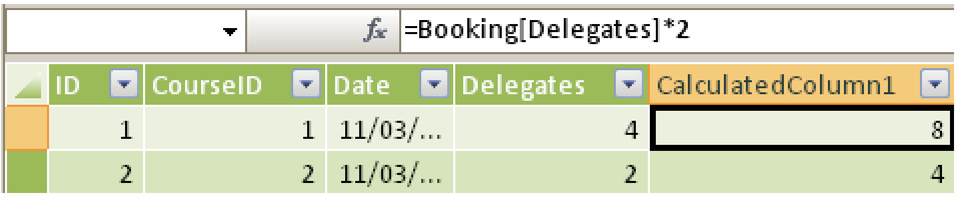
This data model shows that each Customer can have many orders. Each Order may have many order lines. Finally, a Product can be used many times on the Order Line. There is more detail on this later in the manual.

Introduction to calculations

The PowerPivot add-in uses DAX (Data Analysis Expression) language to create calculations in a table or Pivot table. This language is used as part of the Table function in Excel.

You can now write the formula for the new column in the formula bar of PowerPivot. The formula bar looks very similar to the formula bar of Excel. Nevertheless, formulas for PowerPivot are very different from formulas in Excel. PowerPivot does not use the Excel formula language. It uses DAX.

An example of a calculated column is shown below. The values in the Delegates column have been multiplied by 2 and display the results in the new column.



```
=Booking[Delegates]*2
```

ID	CourseID	Date	Delegates	CalculatedColumn1
1	1	11/03/...	4	8
2	2	11/03/...	2	4

Notice how the table name is displayed first followed by the field from the table. The DAX language is explored in more depth later in the manual.

Unit 7 – Data, formatting and layout

In this unit, you will learn how to:

- Create and delete tables
- Set the data type for
- Hide rows or columns of data
- Sort and filter data

Creating and deleting a table

Creating a Table

In PowerPivot, you can create tables of data to analyse data by connecting to various data sources as specified in Unit 5. However, you can copy and paste data from a table or a sheet and paste it into a blank table.

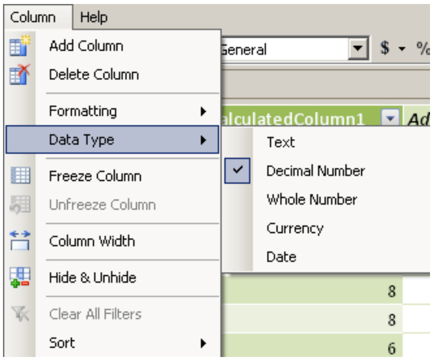
1. Copy some data from a table or a spreadsheet.
2. Switch to the PowerPivot window and select your sheet.
3. Use Edit menu and 'Paste Add New Rows' to add the rows of data to the existing table.
4. Use Edit menu and 'Paste Replace Existing Rows' to replace the existing rows of data.
5. Use Edit menu and 'Paste to a new table' to create a new table with the rows pasted in.

Delete a Table

To delete a table from the PowerPivot window, right click over the Table tab and select Delete. You will be asked to confirm the deletion. Click OK and the table will disappear.

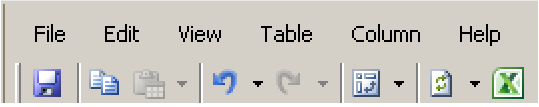
Set the Data Type of a Column

When you add a new column, you may want to set the Data Type to show the values in that column using a certain data format or change the existing data type for a column. Select the column heading and click on the Column menu and select Data Type.



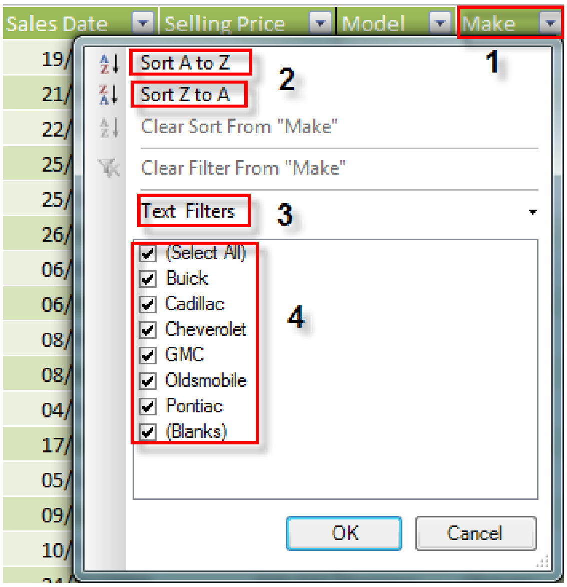
Undo or Redo an Action

To Undo and Redo actions that have been executed use the Undo and Redo buttons on the shortcut toolbar. Alternatively, you could use the Quick Access Toolbar or Ctrl+Z and Ctrl+Y.



Sorting and filtering data in a table

Before you create your Pivot table, the data can be filtered and sorted. Filtering is a method of extracting data from a data table based on criteria given. The filtering feature is available when the column heading filter arrow button is clicked.



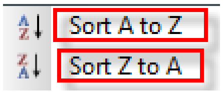
The filter and sorting box contains the following:

1. The column heading the filter is being applied to.
2. Sort the data either in ascending or descending order.
3. Text Filters box allows you to create your filter criteria.
4. Field list containing one occurrence of each Make of the car.

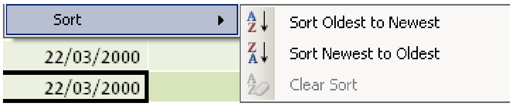
Sorting column data

There are two ways of sorting a column of a data:

1. Click on the filter button for the column that needs to be sorted and choose one of the following buttons:



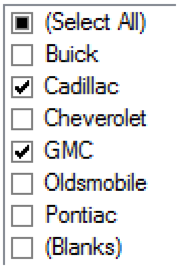
2. Place your cursor in one of the cells containing data and on the Column menu choose Sort and then your choice.



The wording on the two sorting buttons will change depending on the Data Type but in essence, do the same thing.

Filtering Data

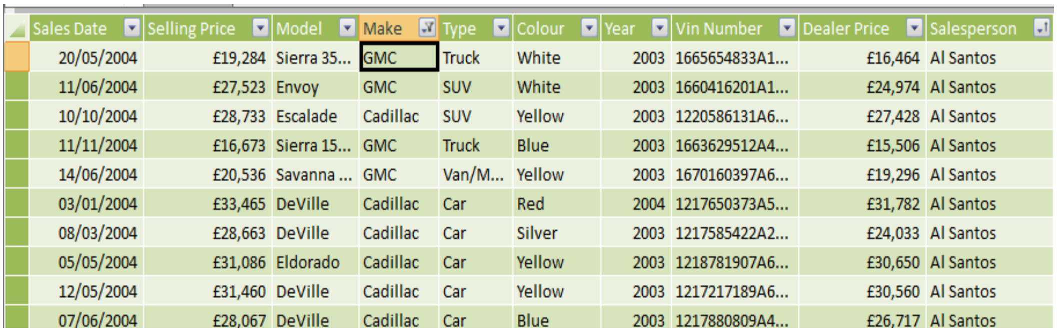
The Filtering functionality allows you to restrict the data records to those you are specifically looking for by applying criteria to the columns. The first method is to select the values from the column you are looking for using the checkboxes. The second way is apply a Text or Value filter to extract records.



■ (Select All)
☐ Buick
☑ Cadillac
☐ Cheverolet
☑ GMC
☐ Oldsmobile
☐ Pontiac
☐ (Blanks)

Click on the filter button for the column heading. Now remove the tick from (Select All). This now allows you to select multiple criteria as all the other ticks have now been removed.

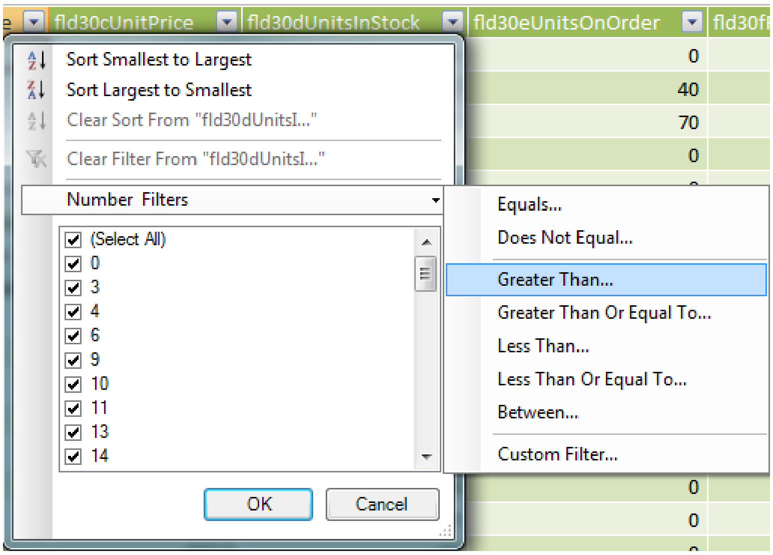
There are two values from the list selected so the results will be those relating to Cadillac and GMC as show in the next diagram displayed below. When the (Select All) checkbox is clicked on again, it will remove the filtered results and display all records.



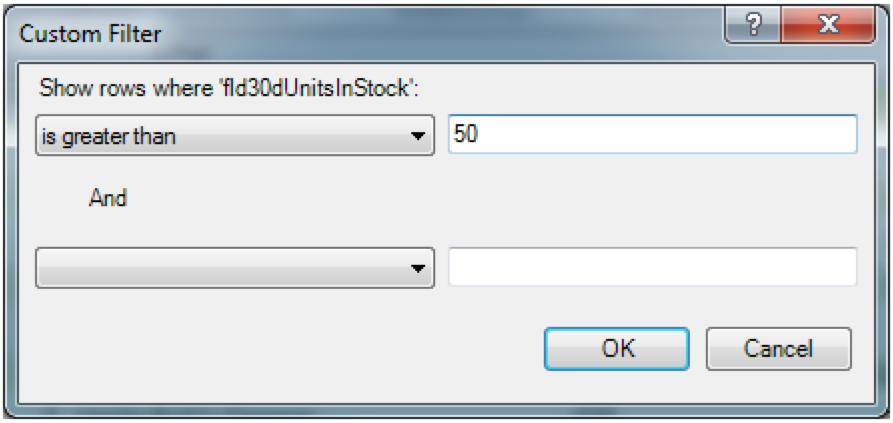
Sales Date	Selling Price	Model	Make	Type	Colour	Year	Vin Number	Dealer Price	Salesperson
20/05/2004	£19,284	Sierra 35...	GMC	Truck	White	2003	1665654833A1...	£16,464	Al Santos
11/06/2004	£27,523	Envoy	GMC	SUV	White	2003	1660416201A1...	£24,974	Al Santos
10/10/2004	£28,733	Escalade	Cadillac	SUV	Yellow	2003	1220586131A6...	£27,428	Al Santos
11/11/2004	£16,673	Sierra 15...	GMC	Truck	Blue	2003	1663629512A4...	£15,506	Al Santos
14/06/2004	£20,536	Savanna ...	GMC	Van/M...	Yellow	2003	1670160397A6...	£19,296	Al Santos
03/01/2004	£33,465	DeVille	Cadillac	Car	Red	2004	1217650373A5...	£31,782	Al Santos
08/03/2004	£28,663	DeVille	Cadillac	Car	Silver	2003	1217585422A2...	£24,033	Al Santos
05/05/2004	£31,086	Eldorado	Cadillac	Car	Yellow	2003	1218781907A6...	£30,650	Al Santos
12/05/2004	£31,460	DeVille	Cadillac	Car	Yellow	2003	1217217189A6...	£30,560	Al Santos
07/06/2004	£28,067	DeVille	Cadillac	Car	Blue	2003	1217880809A4...	£26,717	Al Santos

Value Filter

When you have numerical values and you are looking for a specific set of numbers, then a Number Filter is the quickest way to extract the data.



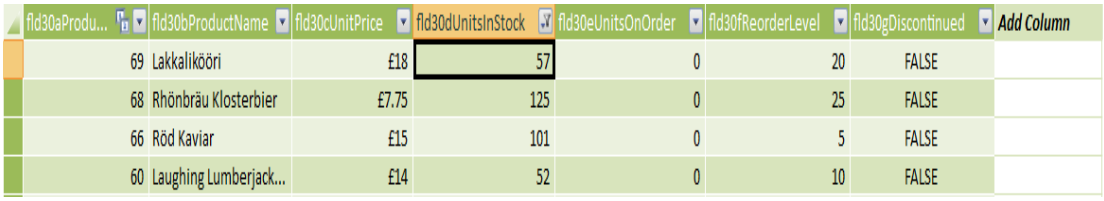
This filter is going to extract all the records where the Units in Stock are greater than 50.



Show rows where 'fld30dUnitsInStock':

is greater than 50

The results are shown below:



fld30aProdu...	fld30bProductName	fld30cUnitPrice	fld30dUnitsInStock	fld30eUnitsOnOrder	fld30fReorderLevel	fld30gDiscontinued	Add Column
69	Lakkaliköori	£18	57	0	20	FALSE	
68	Rhönbräu Klosterbier	£7.75	125	0	25	FALSE	
66	Röd Kaviar	£15	101	0	5	FALSE	
60	Laughing Lumberjack...	£14	52	0	10	FALSE	

Unit 8 - PowerPivot and relationships

In this unit, you will learn how to:

- Understand relationships
- Importing database tables with relationships already set up
- Create a Relationship between Two Tables
- View, Edit and Delete Relationships
- Troubleshoot Relationships
- Visualize Disconnected Tables

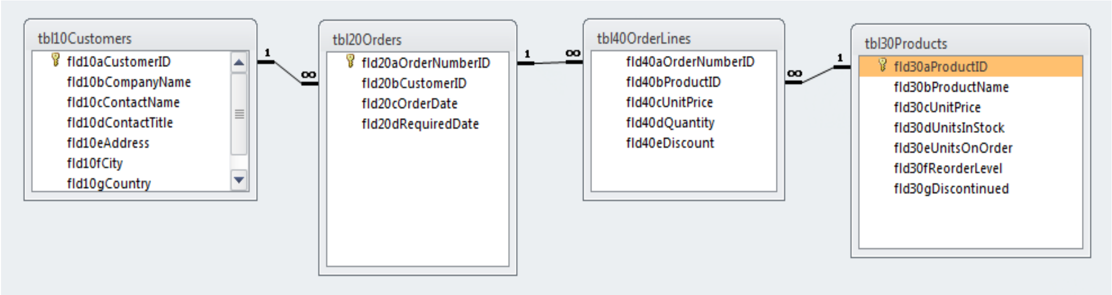
Understanding Relationships (Normalisation)

One of the goals of good database design is to remove redundant data (duplicate data). In order to do this, you divide your data into many tables so that each fact is represented only once. You then have to ensure that the database can link the disjointed tables of data. You do this by placing common fields in tables that are related.

These two scenarios demonstrate that the Excel way does not need a data model whereas the PowerPivot way creates a data model. But one question is still open: what is a data model? A Data Model is nothing but a list of tables (sometimes referred to as entities) with arrows connecting them. A table, as its name suggests, is a list of columns that hold the real data. The arrows represent the existence of relationships between tables and are normally read as refers to. So, we might say that an order line refers to a product, meaning that the order line is about a specific product.

In the example below, the Customer and Orders table have fields with the same name, CustomerID. Each table will usually have a field that uniquely identifies each record in the table. This is known as the Primary Key. There is a natural link between customers and orders table as a customer will make orders. So, we create a Lookup between the tables.

The first step is to add the CustomerID field into the Orders table which will be used to Lookup the information from the Primary key and make those values available in the Orders table. This second field in the Orders table is known as the Foreign field. The tables are then linked from the Primary to the Foreign key creating a One to Many relationship with the One part being attributed to the Unique field (Primary key).



This data model shows that each Customer can have many orders. Each Order may have many order lines. Finally, a Product can be used many times on the Order Line. There is more detail on this later in the manual.

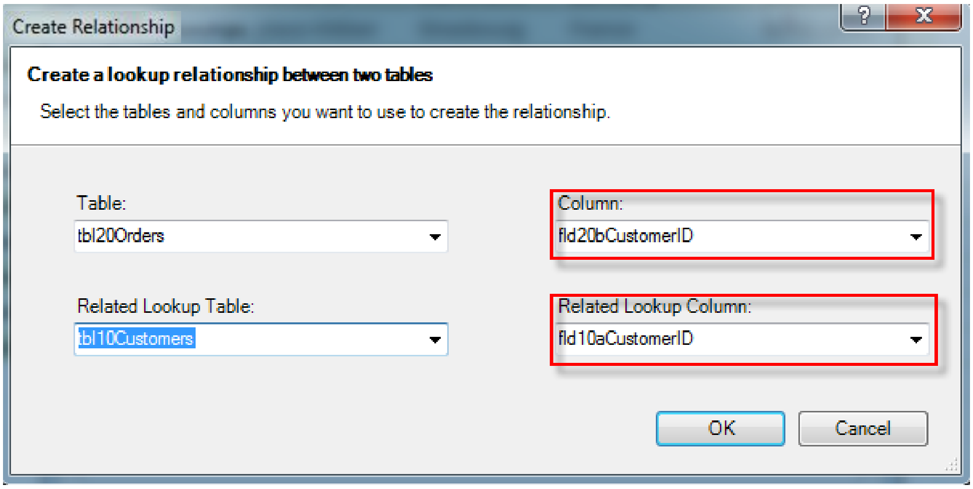
Importing database tables with relationships already set up

If the database you are importing already has had the data tables normalized, common fields in related tables and the relationships have been set up already in the database, then PowerPivot will pull through the relationships into the PowerPivot window. Now data from multiple linked tables can be analysed a massive plus point over Excel Pivot tables.

Create a Relationship between Two Tables

Once you have connected to the database and successfully imported the data, if these attributes mentioned in the last section are present then you can create the relationships between the tables using the Design Tab and selecting the Create Relationships option.

Go to the Design tab; select Relationships and then Create Relationship.



Create a lookup relationship between two tables

Select the tables and columns you want to use to create the relationship.

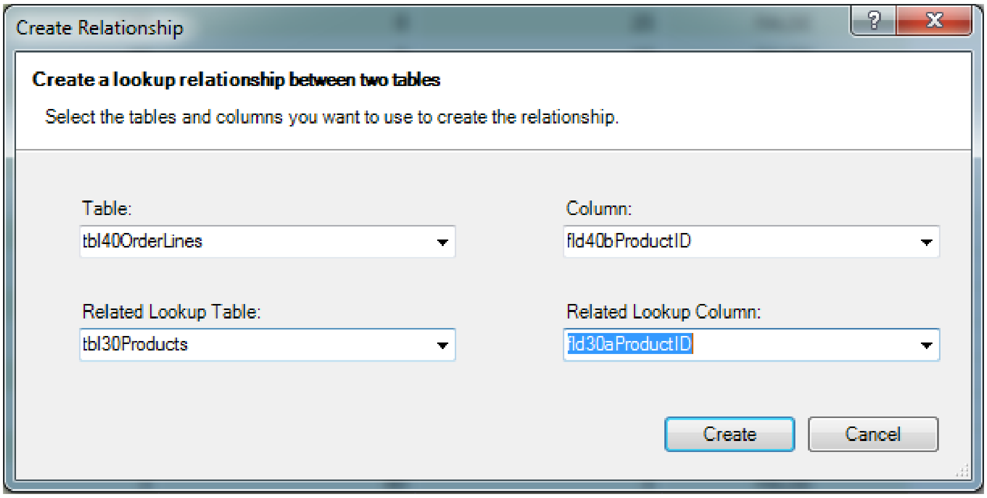
Table: tbl20Orders

Column: fld20bCustomerID

Related Lookup Table: tbl10Customers

Related Lookup Column: fld10aCustomerID

The relationship above has created a lookup from the Orders table to the Customers table and the one below from Order Lines to Products tables.



Create Relationship

Create a lookup relationship between two tables

Select the tables and columns you want to use to create the relationship.

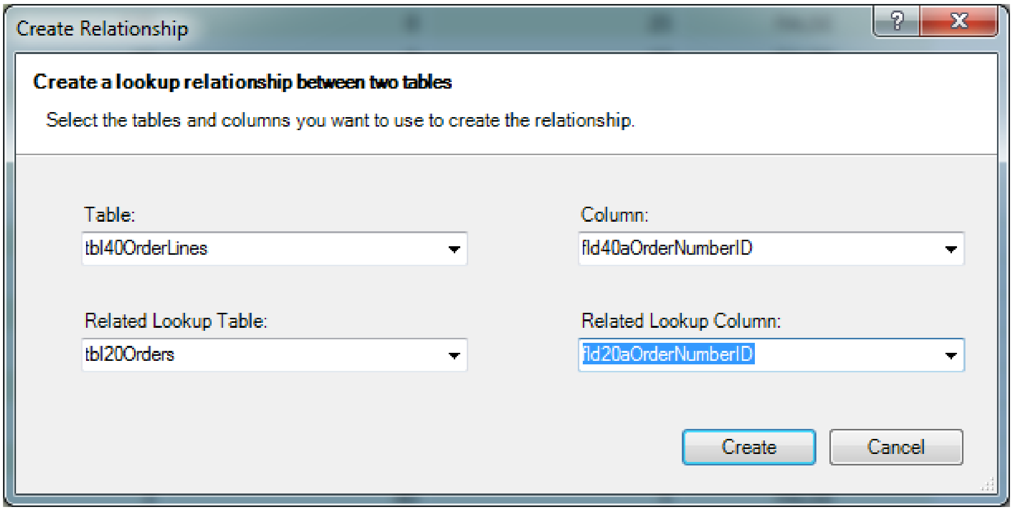
Table: tbl40OrderLines

Column: fld40bProductID

Related Lookup Table: tbl30Products

Related Lookup Column: fld30aProductID

The final relationship to create is the lookup from the Order Lines table to the Orders table using the Order Number ID field.



Create Relationship

Create a lookup relationship between two tables

Select the tables and columns you want to use to create the relationship.

Table: tbl40OrderLines

Column: fld40aOrderNumberID

Related Lookup Table: tbl20Orders

Related Lookup Column: fld20aOrderNumberID

View, Edit and Delete Relationships

To view relationships and edit them, go to the Design menu, Relationships and then Manage Relationships.

You will then be presented with the next image and lists all the relationships present between the tables.

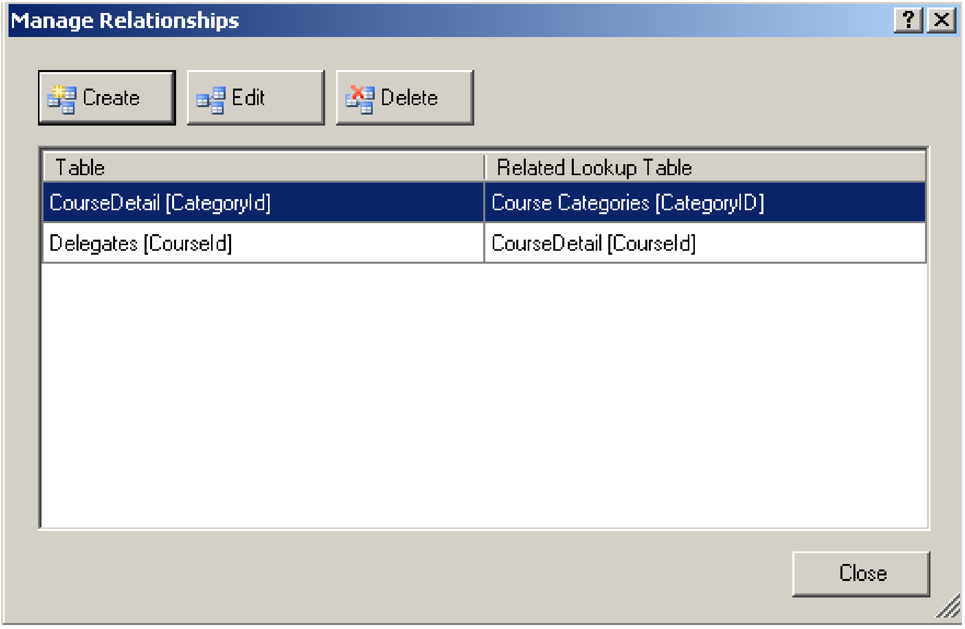
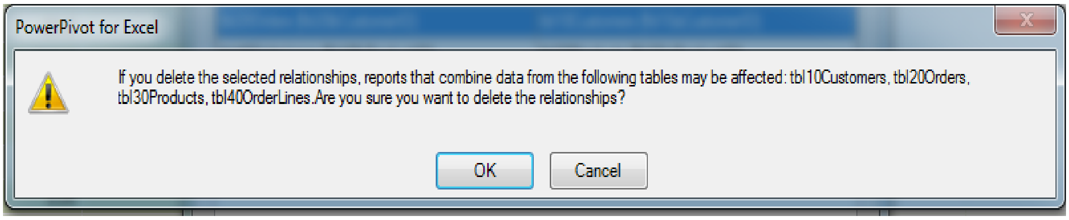


Table	Related Lookup Table
CourseDetail [CategoryId]	Course Categories [CategoryID]
Delegates [CourseId]	CourseDetail [CourseId]

If you have missed any relationships, use the Create button to create another relationship.

Use the Edit button if you have noticed you have used the wrong fields. This is important otherwise the Lookup from one table to another will fail and you won't get the related data and in most instances, you will get every single possible outcome between the tables. The Edit Relationships window appears which is identical to the Create Relationships window. Change the Tables and/or fields to recreate the relationship.

Use the Delete button to delete the relationship between tables. It will warn you that other tables may be affected.

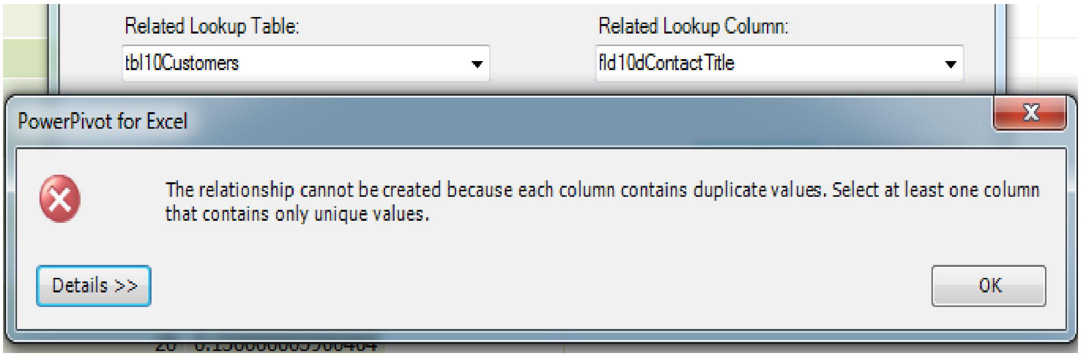


If you delete the selected relationships, reports that combine data from the following tables may be affected: tbl10Customers, tbl20Orders, tbl30Products, tbl40OrderLines.Are you sure you want to delete the relationships?

Troubleshoot Relationships

When creating the Lookups or relationships between the related tables the following signs may appear to warn you something is wrong or giving you information.

⊗ This symbol means the fields do not contain duplicate information.


Related Lookup Table: tbl10Customers

Related Lookup Column: fld10dContactTitle

The relationship cannot be created because each column contains duplicate values. Select at least one column that contains only unique values.

There is also an Information symbol ℹ that may appear. This means that your Lookup and Related Lookup tables are the wrong Way round. In the example below, it is trying to Lookup from the Primary table to the Foreign key table. This should be the other way round as the foreign key field looks up the data from the related table. The message is:

The relationship cannot be created in the requested direction. When you click create, the direction of the relationship will be reversed'.

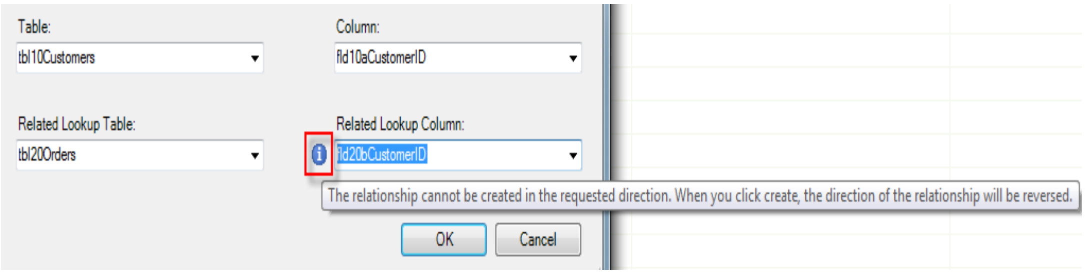


Table: tbl10Customers

Column: fld10aCustomerID

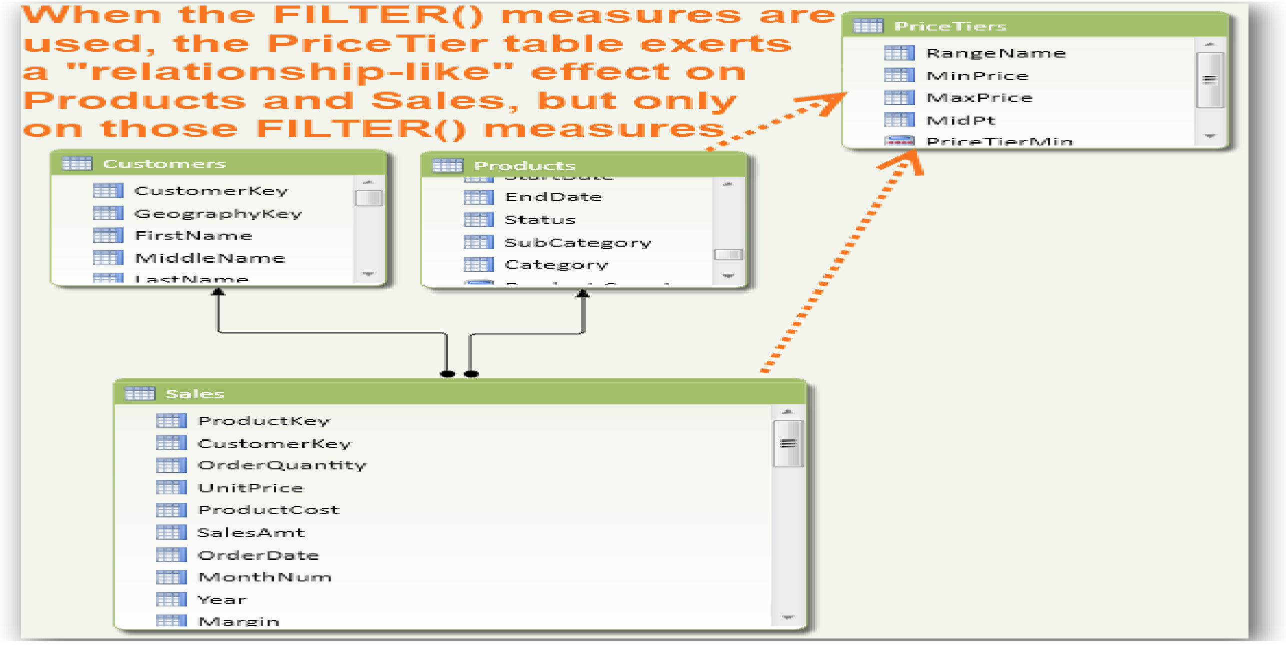
Related Lookup Table: tbl20Orders

Related Lookup Column: fld20bCustomerID

The relationship cannot be created in the requested direction. When you click create, the direction of the relationship will be reversed.

A Way to Visualize Disconnected Tables

The Diagram view allows you to look at the tables and their relationships. The Diagram view is accessed via the PowerPivot window, Home tab and then the Diagram View button.


When the FILTER() measures are used, the PriceTier table exerts a "relationship-like" effect on Products and Sales, but only on those FILTER() measures

Disconnected tables, by definition, have no relationships to other tables in the model. If we look at diagram view above, we see that the PriceTiers table, for instance, is an island like we expect:

But when we use the “MinMaxTier” measures that we wrote above, the PriceTiers table does act a lot like a Lookup table, since the PriceTiers filter context (such as user selections on the slicer) very much impacts the measure calculations and results.

So, I often like to say that disconnected tables have a “dotted line” relationship with the tables that contain the corresponding FILTER() measures. In your head, you might think of it like this:

Disconnected tables only impact the measures that are specifically written to “pay attention” to them – so the PriceTiers table impacts [ProductCount MinMaxTier] and [Total Sales MinMaxTier], but no other measures in the Products and Sales tables.

Unit 9 - Calculations in PowerPivot

In this unit, you will learn how to:

- Understand Data Analysis Expressions (DAX) language
- Build Formulas for Calculated Columns and Measures
- Understand the Use of Relationships and Lookups in Formulas
- Understand Aggregations in Formulas
- Recalculate Formulas

Overview of Data Analysis Expressions (DAX) language

Now that you have seen some examples of PowerPivot for Excel worksheets, it is time to learn the fundamentals of PowerPivot expressions. PowerPivot has its own syntax for defining calculation expressions. It is conceptually similar to an Excel expression, but it has specific functions that allow you to create more advanced calculation on data stored in multiple tables. The PowerPivot language is called Data Analysis Expressions, but we always use the shorter DAX acronym.

In this unit, you learn the basics of DAX and also discover how to use it to solve some typical problems in business scenarios.

Understanding Calculation in DAX

Just as it does in Excel, any calculation in DAX begins with the assignment operator. The main difference is that DAX never uses cell coordinates like A1, C2, and so on. In DAX, you always specify coordinates using column and table names. Moreover, DAX does not support the concept of range as Excel does: to use DAX efficiently, you need to learn to work with columns and tables.

Please note that in a DAX expression, you can get the value of a column only for a single row or for the whole table—that is, you cannot get access to a specific row inside a table. To get a range, you need to use DAX functions that filter a table, thus returning a subset of the rows of the original table, corresponding to the needed range.

To express complex formulas, you need to learn the basics of DAX, which includes the syntax, the different data types that DAX can handle, the basic operators, and how to refer to columns and tables. In the next few sections, we are going to introduce these concepts.

Building Formulas for Calculated Columns and Measures

Now that you know the basics of DAX syntax, you need to learn one of the most important concepts in DAX: the difference between calculated columns and measures. Even though they might appear similar at first sight because you can make some calculations both ways, you will see that you need to use measures to implement the most flexible calculations, and this is a key to becoming a powerful PowerPivot user.

Calculated Columns

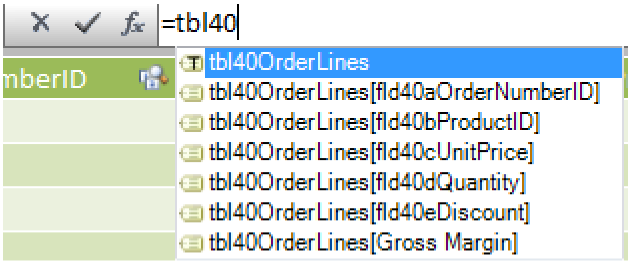
You can create a calculated column by using the Add Column button on the Columns group of the Design tab, or you can simply move to the last column, which is named Add Column, and start writing the formula. The DAX expression has to be inserted into the formula bar and IntelliSense helps you during the writing of the expression.

A calculated column is just like any other column in a PowerPivot table and can be used in rows, columns, filters, or values of a pivot table. The DAX expression defined for a calculated column operates in the context of the current row of the table it belongs to. Any reference to a column returns the value of that column in the current row. You cannot access directly the values of other rows.

In the next few lines, we will work out the Gross Margin in the Order Lines table. We will use the Quantity and Price fields.

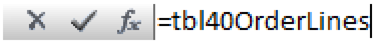
The formula in the new calculated column will be the following:

An example of IntelliSense and the steps to create the Gross Margin Calculated Column is outlined below:

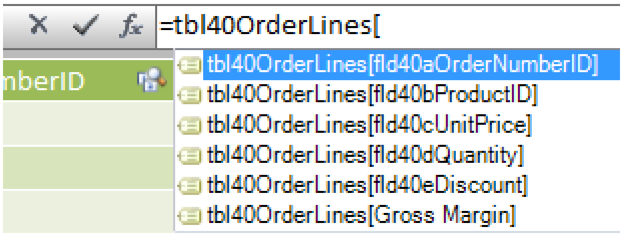

=tbl40

- tbl40OrderLines
- tbl40OrderLines[fld40aOrderNumberID]
- tbl40OrderLines[fld40bProductID]
- tbl40OrderLines[fld40cUnitPrice]
- tbl40OrderLines[fld40dQuantity]
- tbl40OrderLines[fld40eDiscount]
- tbl40OrderLines[Gross Margin]

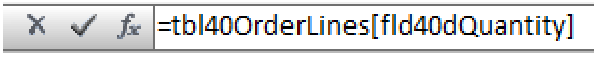
As soon as you start typing the table name, IntelliSense appears with a list of tables to choose from. Once the table is selected, press the tab key to complete the entry.


=tbl40OrderLines

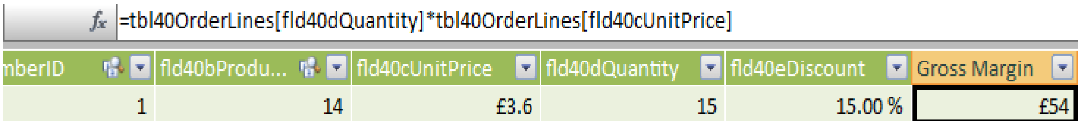
After the table name has been entered, you need to open a square bracket [and IntelliSense will once again give you a list of fields within the table. Once the field is selected or you pressed the tab key, the rest of the field is entered followed by a closing square bracket].


=tbl40OrderLines[

- tbl40OrderLines[fld40aOrderNumberID]
- tbl40OrderLines[fld40bProductID]
- tbl40OrderLines[fld40cUnitPrice]
- tbl40OrderLines[fld40dQuantity]
- tbl40OrderLines[fld40eDiscount]
- tbl40OrderLines[Gross Margin]


=tbl40OrderLines[fld40dQuantity]

The multiplication sign is then added *, followed by the Order Line table and Unit Price field. The final formula is shown below and it applies the same formula to each row.



```
=tbl40OrderLines[fld40dQuantity]*tbl40OrderLines[fld40cUnitPrice]
```

nberID	fld40bProdu...	fld40cUnitPrice	fld40dQuantity	fld40eDiscount	Gross Margin
1	14	£3.6	15	15.00 %	£54

An example of where a calculated field may not compute correctly is if you add a calculated column to a table to work out a percentage of one field against another. When the data is aggregated into the Pivot table, it uses the Sum option which is wrong. You could change the function applied to the data but it would do an average of the average which is not accurate.

Calculated columns are good at row level, however this calculation needs to be done at cell level. This is where you need to create a Measure.

Measures

A measure is a DAX expression that uses the same syntax as calculated columns; the difference is the context of evaluation. A measure is evaluated in the context of the cell of the pivot table, while a calculated column is computed at the row level. The cell context depends on the user selections on the pivot table. So, when you use SUM(SalesAmount) in a measure, you mean the sum of all the cells that are aggregated under this cell, whereas when you use [SalesAmount] in a calculated column, you mean the value of SalesAmount in this row.

When you create a measure, you can define a value that changes according to the filter that the user applies on a pivot table. In this way, you can solve the problem of calculating the gross margin percentage. To define a measure, you can click the New Measure button on the PowerPivot tab of the ribbon, shown in Figure 3-10, whenever a cell in a PivotTable is selected.

Differences Between Calculated Columns and Measures

Even if they look similar, there is a big difference between calculated columns and measures.

The value of a calculated column is calculated during data refresh and uses the current row as a context; it does not depend on user activity on the pivot table. A measure operates on aggregations of data defined by the context of the current cell: source tables are filtered according to the coordinates of the cell, and data is aggregated and calculated using this filter. In other words, a measure always operates on aggregations of data under the evaluation context and for this reason there is no way to reference a single row in a DAX expression.

Now that you have seen the difference between calculated columns and measures, you might be wondering when it is better to use calculated columns and when to use measures.

Sometimes, either is an option, but in most situations, your computation needs determine your choice.

You have to define a calculated column (in the PowerPivot table grid window) whenever you want to do the following:

- Place the calculated results in an Excel Slicer or see results in Rows or Columns in a pivot table (as opposed to the Values area).
- Define an expression that is strictly bound to the current row. (For example, Price * Quantity cannot work on an average of the two columns)
- Categorize text or numbers (for example, a range of values for a measure, a range of ages of customers, and so on).

On the other hand, you have to define a measure (in the PowerPivot Field List in the pivot table) whenever you want to display resulting calculation values that reflect pivot table selections made by the user and see them in the Values:

- When you calculate profit percentage of a pivot table selection
- When you calculate ratios of a product compared to all products but filter both by year or region

Some calculations can be covered both by calculated columns and measures, even if different DAX expressions have to be used in these cases. For example, you can define the GrossMargin as a calculated column:

= Sales[SalesAmount] - Sales[TotalProductCost]

However, it can be defined as a measure too:

= SUM(Sales[SalesAmount]) – SUM(Sales[TotalProductCost])

Understanding the Use of Relationships and Lookups in Formulas

One of the most powerful features in PowerPivot for Excel is the ability to create relationships between tables and then use the related tables to look up or filter related data. You retrieve related values from tables by using the formula language provided with PowerPivot for Excel, Data Analysis Expressions (DAX). DAX uses a relational model and therefore can easily and accurately retrieve related or corresponding values in another table or column.

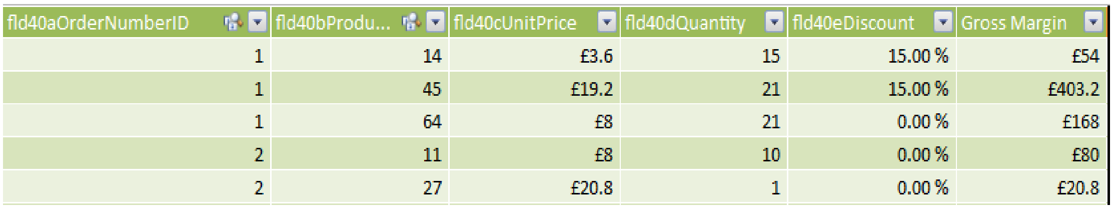
The ability to look up matching or related data from another table is particularly useful in situations where the current table has only an identifier of some kind, but the data that you need (such as product price, name) is stored in a related table. It is useful when there are multiple rows in another table that are related to the current row or current value. For example, you can easily retrieve all the sales that are tied to a particular region.

In contrast to Excel lookup functions such as VLOOKUP, which are based on arrays, or LOOKUP, which gets the first of multiple matching values, DAX follows existing relationships among tables joined by keys to get the single related value that matches exactly. DAX can also retrieve a table of records that are related to the current record.

Creating a Calculated column using the Relationships and Lookup formula

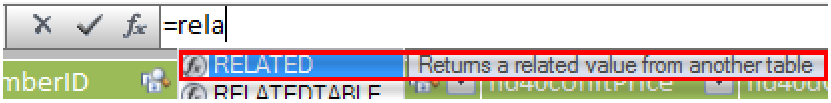
This example is following on from the Gross Margin. We are now going to add a calculated column and use a formula that analyses the relationships and lookups and will return a related value from another table. In this example, we are going to use the Related function to return the related value from the Customers table.

So, the result of this function will be to display the customer name from the customers table.



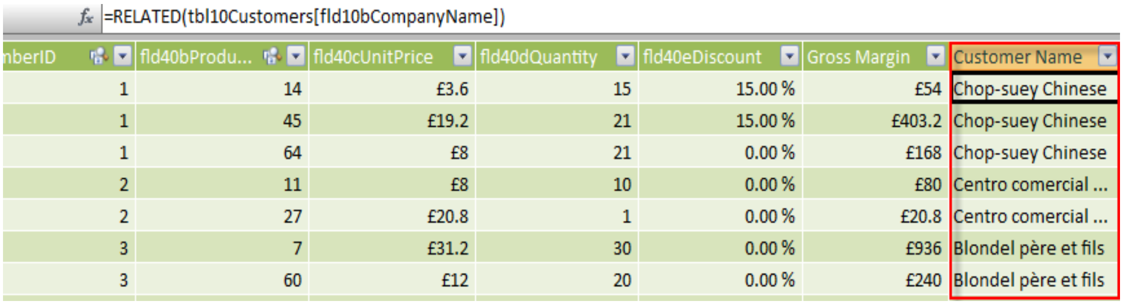
fld40aOrderNumberID	fld40bProdu...	fld40cUnitPrice	fld40dQuantity	fld40eDiscount	Gross Margin
1	14	£3.6	15	15.00 %	£54
1	45	£19.2	21	15.00 %	£403.2
1	64	£8	21	0.00 %	£168
2	11	£8	10	0.00 %	£80
2	27	£20.8	1	0.00 %	£20.8

A new column has been added and called Customer Name. The following formula was built in the first cell.



=rela

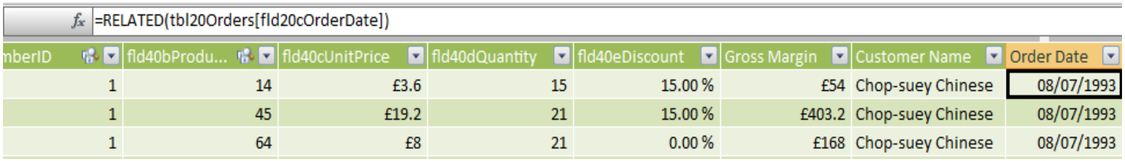
Like in Excel, when you start typing a function name, IntelliSense will display a list of functions that match the data entry. Press the tab key to complete the function.



=RELATED(tbl10Customers[fld10bCompanyName])

nberID	fld40bProdu...	fld40cUnitPrice	fld40dQuantity	fld40eDiscount	Gross Margin	Customer Name
1	14	£3.6	15	15.00 %	£54	Chop-suey Chinese
1	45	£19.2	21	15.00 %	£403.2	Chop-suey Chinese
1	64	£8	21	0.00 %	£168	Chop-suey Chinese
2	11	£8	10	0.00 %	£80	Centro comercial ...
2	27	£20.8	1	0.00 %	£20.8	Centro comercial ...
3	7	£31.2	30	0.00 %	£936	Blondel père et fils
3	60	£12	20	0.00 %	£240	Blondel père et fils

The completed function and the results are shown in the image above. The image belows shows a second example of adding the Order Date from the Orders table.



```
=RELATED(tbl20Orders[fld20cOrderDate])
```

mberID	fld40bProdu...	fld40cUnitPrice	fld40dQuantity	fld40eDiscount	Gross Margin	Customer Name	Order Date
1	14	£3.6	15	15.00 %	£54	Chop-suey Chinese	08/07/1993
1	45	£19.2	21	15.00 %	£403.2	Chop-suey Chinese	08/07/1993
1	64	£8	21	0.00 %	£168	Chop-suey Chinese	08/07/1993

Understanding Aggregations in Formulas

This topic introduces aggregations and provides an overview of the types of aggregations that are possible with PowerPivot for Excel. PowerPivot for Excel contains these tools for building aggregations:

You can build PivotTables and PivotCharts that are based on PowerPivot data. Excel PivotTables are a popular tool for grouping and summarizing data in worksheets. PowerPivot is integrated with the PivotTable features in Excel and provides many enhancements.

You can use the DAX formula language to design custom aggregations. DAX can be used to create calculated columns in PowerPivot tables and to create measures in PivotTables and PivotCharts.

Aggregations are a way of collapsing, summarizing, or grouping data. When you start with raw data from tables or other data sources, the data is appears quite flat meaning there is lots of detail but it has not been organized in any way.

Choosing Groups for Aggregation

When you aggregate data, you group data by attributes such as product, price, region, or date and then define a formula that works on all the data in the group. For example, when you create a total for a year, you are creating an aggregation. If you then create a ratio of this year over the previous year and present those as percentages, it is a different type of aggregation.

The decision of how to group the data is driven by the business question. Here are some sample questions:

Count function - How many transactions were there in a month?

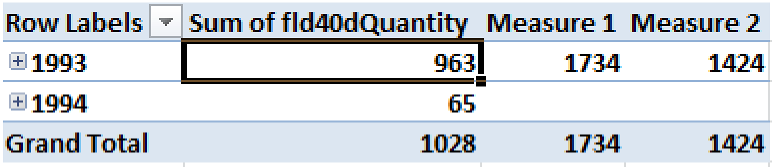
Average function - What were the mean sales in this month, by salesperson?

Max Function - Which sales districts were the top five in terms of units sold?

If you have Dates, you can aggregate by using various DAX functions. There are the following functions that can be used. These functions include Year and Month functions. An example of these formulas would be:

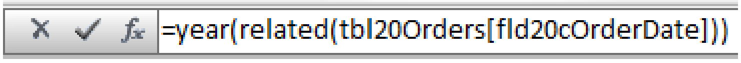
Order Year = Year(SalesOrderHeader[OrderDate]) – this will return the year so it allows you to aggregate at Year level.

Order Month = Month(SalesOrderHeader[OrderDate]) – this will return the year so it allows you to aggregate at Month level.



Row Labels	Sum of fld40dQuantity	Measure 1	Measure 2
⊞1993	963	1734	1424
⊞1994	65		
Grand Total	1028	1734	1424

The first step to achieving this aggregated result, is to create a Calculated column in the PowerPivot window with the following formula:



```
=year(related(tbl20Orders[fld20cOrderDate]))
```

This formula looks up the data in the related Orders table and returns the four digit year from the full date. Then add the new calculated column into the Pivot table for the aggregated values.

This final example returns the number of orders in each year for the month of November.

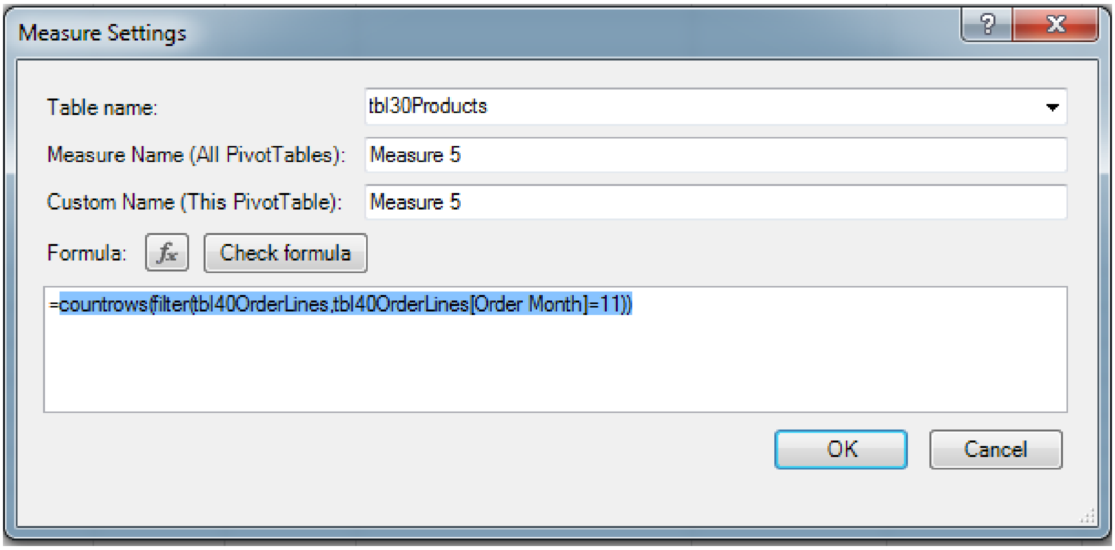


Table name: tbl30Products

Measure Name (All PivotTables): Measure 5

Custom Name (This PivotTable): Measure 5

Formula:

```
=countrows(filter(tbl40OrderLines,tbl40OrderLines[Order Month]=11))
```

This formula counts the rows that match the filter of month 11 which is November.

Filtering Data in Formulas

You can create filters in formulas to restrict the data being returned from the data source that are used in calculations. These results will automatically adjust as Pivot table is rearranged.

There are DAX functions that can be used to give you aggregated totals based on filtered data. The following selection of functions can be used:

Countx, Averagex, Maxx, Minx, Sumx

The formula syntax is the following:

=Sumx(Table,Filter),Column to be aggregated

The && constitutes 'AND' logic so both sets of criteria must match.

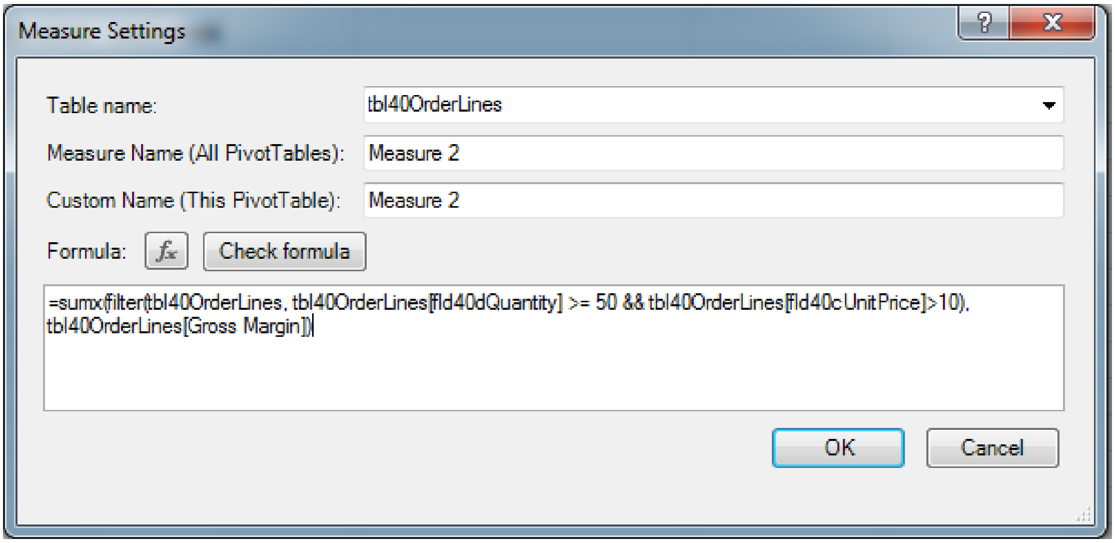


Table name: tbl40OrderLines

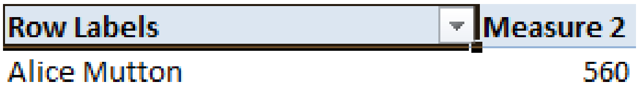
Measure Name (All PivotTables): Measure 2

Custom Name (This PivotTable): Measure 2

Formula:

```
=sumx(filter(tbl40OrderLines, tbl40OrderLines[fld40dQuantity] >= 50 && tbl40OrderLines[fld40cUnitPrice]>10),
tbl40OrderLines[Gross Margin])
```

The formula listed above, is doing a sum of filtered results based on 2 conditions using 'And' logic, which means both conditions have to be met. So, for each Product it would show the Sum of the Gross Margin for those products where the Quantity is >=50 and Unit Price > 10 as shown below:



Row Labels	Measure 2
Alice Mutton	560

Recalculating Formulas

When you are working with data in a PowerPivot for Excel workbook, you may need to refresh the data from the source. There will also be a need to recalculate the formulas that you have created in calculated columns, or make sure that the data presented in a PivotTable is the most recent.

Data refresh

This term means obtaining up-to-date data from external data sources. PowerPivot does not automatically detect changes in external data sources, but data can be refreshed manually from the PowerPivot workbook or automatically if the workbook is shared on SharePoint.

Recalculation

This term means updating all the columns, tables, charts, and PivotTables in your workbook that contain formulas. Because recalculation of a formula incurs a performance cost, it is important to understand the dependencies associated with each calculation.

Using Automatic Recalculation

When you use automatic recalculation mode, any changes to data in the workbook that would cause the result of any formula to change will trigger recalculation of the entire column that contains a formula. The following always require recalculation of formulas:

- Values from an external data source have been refreshed.
- The definition of the formula changes.
- Names of tables or columns that are referenced in a formula have been edited.
- Relationships between tables have been added, modified, or removed.
- New measures or calculated columns have been added.
- Changes made to other formulas within the PowerPivot workbook, so columns or calculations that depend on that formula should be refreshed.

- Rows have been added or deleted.
- You applied a filter that requires execution of a query to update the data set. The filter could have been applied either in a formula or as part of a PivotTable or PivotChart.

Using Manual Recalculation

You can use manual recalculation to avoid incurring the cost of computing formula results until you are ready. Manual mode is particularly useful in these instances:

- You are designing a formula by using a template and want to change the names of the columns and tables used in the formula before you validate it.
- You know that some data in the workbook has changed but you are working with a different column that has not changed so you want to postpone a recalculation.
- You are working in a workbook that has many dependencies and want to defer recalculation till you are sure all the necessary changes have been made.

Unit 10 - Introduction to DAX Measures

In this unit, you will learn how to:

- Understand measures

Recalculate Formulas

“The Best Thing to Happen to Excel in 20 Years”

That’s a quote from Mr. Excel himself, Bill Jelen. He was talking about PowerPivot in general, but specifically measures. So, what are measures?

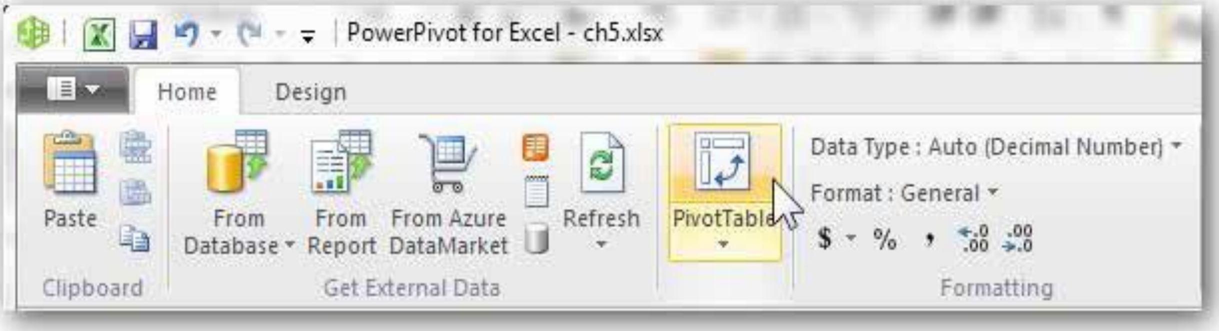
On the surface, you can think of Measures as “formulas that you add to a pivot.” But they offer you unprecedented power and flexibility, and their benefits extend *well* beyond the first impression.

Measures

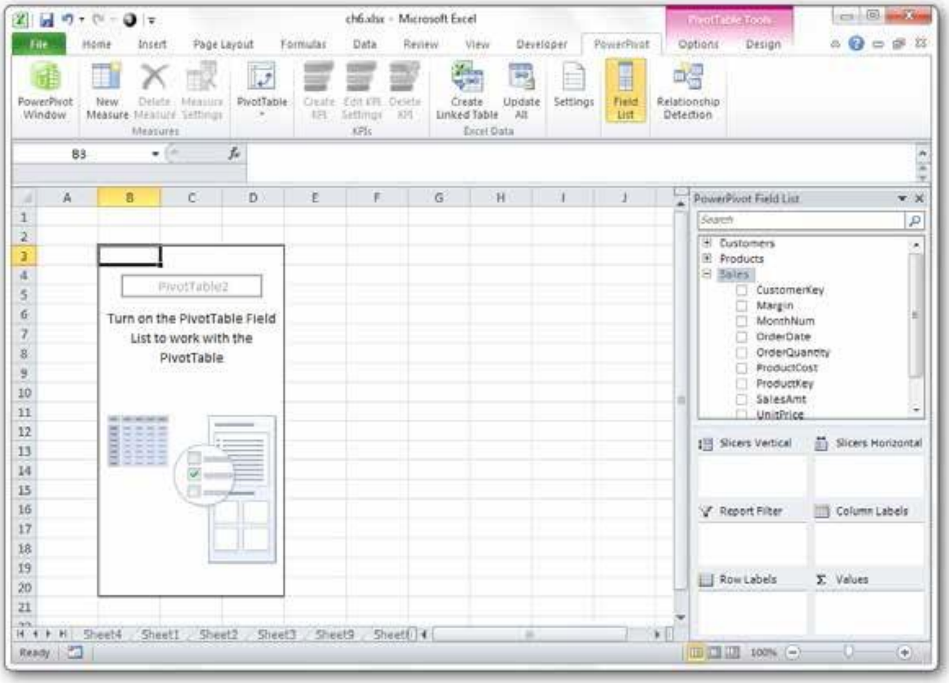
There are two places where you can add a measure: In the Excel window (attached to a pivot) or in PowerPivot window (in the measure grid).

I highly recommend starting out with the first option – in the Excel window, attached to a pivot, because that gives you the right context for validating whether your formula is correct.

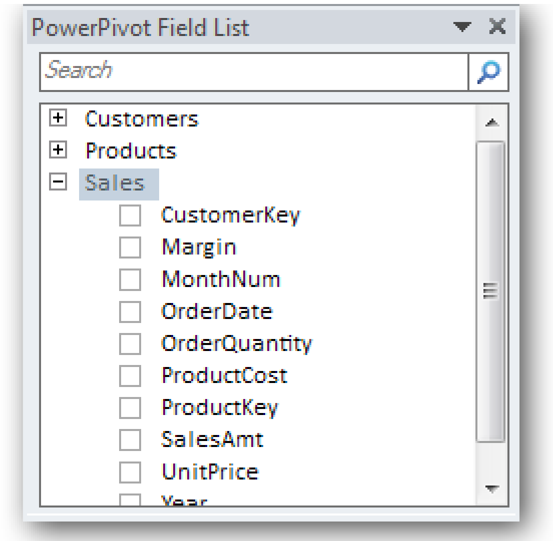
Create a Pivot table as follows: Use the pivot button on the ribbon in the PowerPivot window.



This yields a blank pivot on a new worksheet:

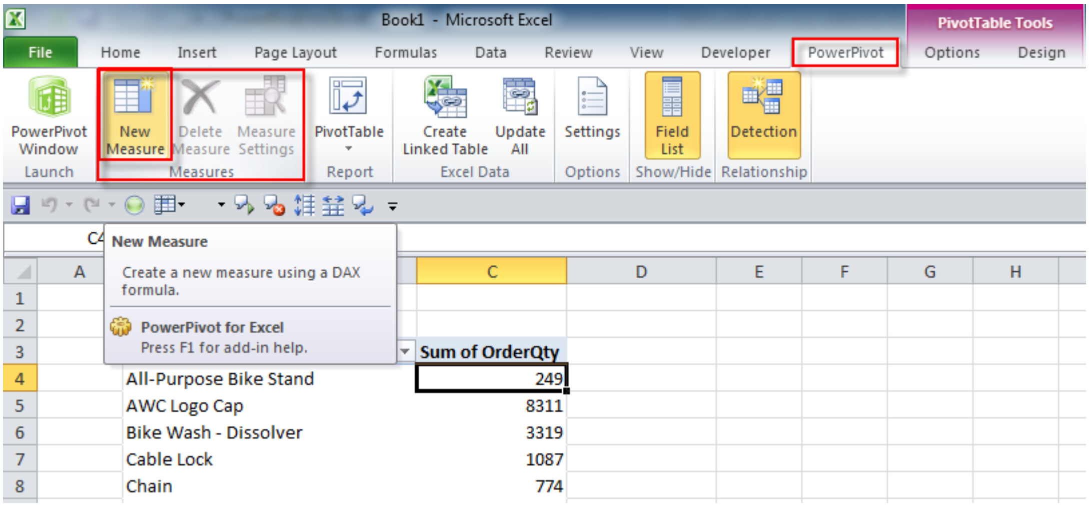


Notice how the pivot field list contains all three tables from the PowerPivot window.



Add a Measure

Click the New Measure button on the PowerPivot ribbon tab in Excel (Excel 2013 it is called calculated field):



Create a new measure using a DAX formula.

	A	B	C	D	E	F	G	H
1								
2								
3			Sum of OrderQty					
4		All-Purpose Bike Stand	249					
5		AWC Logo Cap	8311					
6		Bike Wash - Dissolver	3319					
7		Cable Lock	1087					
8		Chain	774					

This brings up the Measure Settings dialog, which I will often refer to as the measure editor, or often as just "the editor."

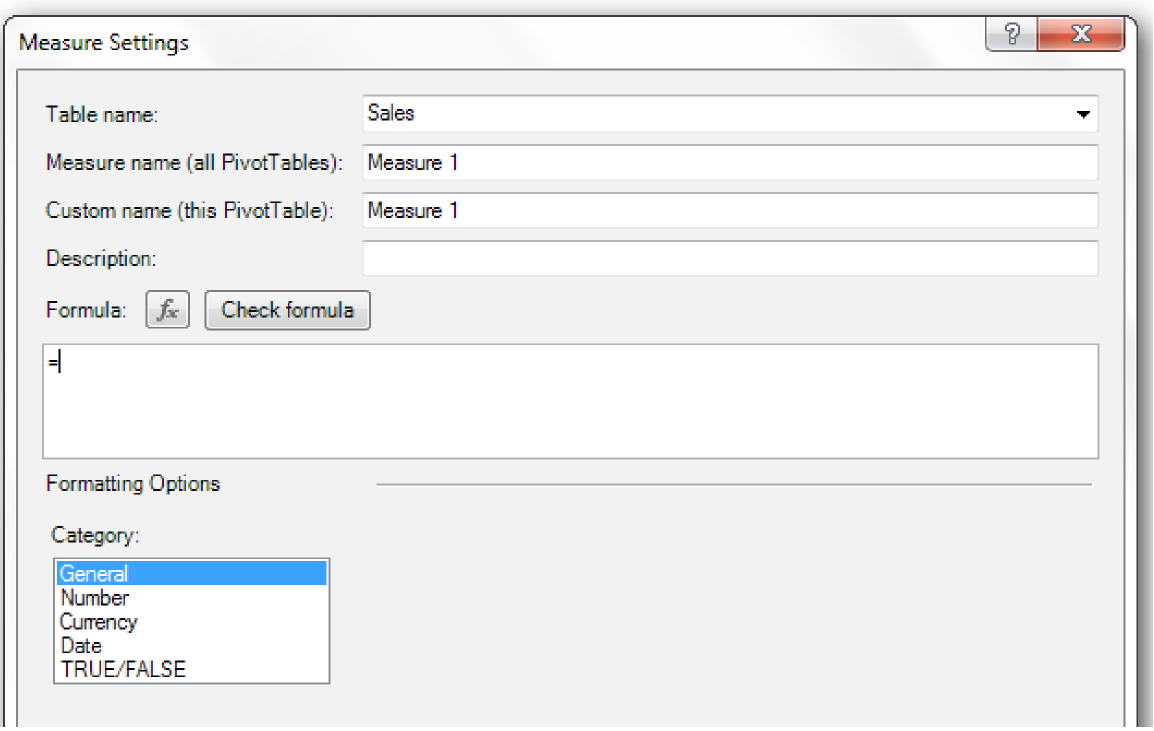

Measure Settings

Table name: Sales

Measure name (all PivotTables): Measure 1

Custom name (this PivotTable): Measure 1

Description:

Formula:

=

Formatting Options

Category: General

There is a lot going on in this dialog, but for now let's ignore most of it and just write a simple formula:

=SUM(Sales[SalesAmt])

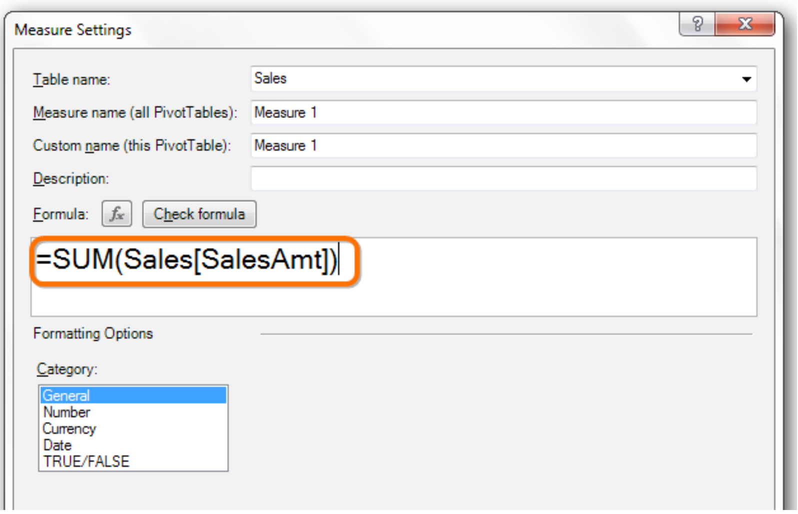

Measure Settings

Table name: Sales

Measure name (all PivotTables): Measure 1

Custom name (this PivotTable): Measure 1

Description:

Formula:

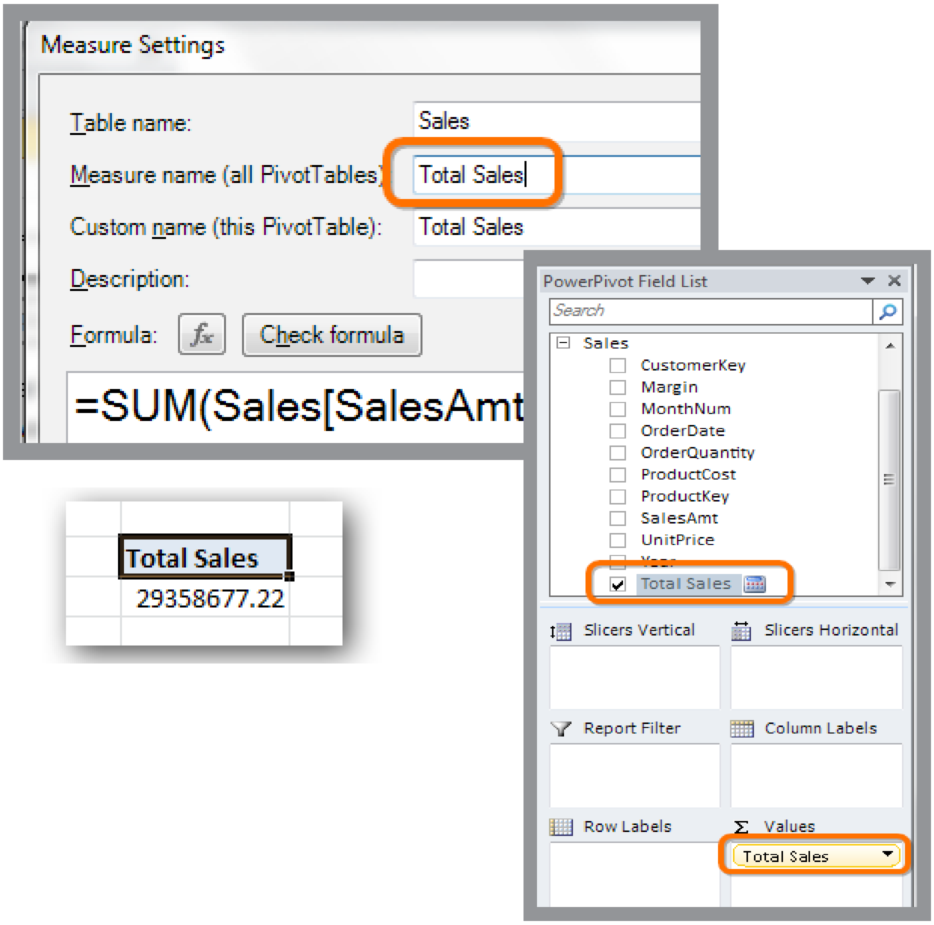
=SUM(Sales[SalesAmt])

Formatting Options

Category: General

Name the Measure

Before clicking OK, I will give the measure a name. This is just as important as giving sensible names to tables and columns.



Measure Settings

Table name: Sales

Measure name (all PivotTables): Total Sales

Custom name (this PivotTable): Total Sales

Description:

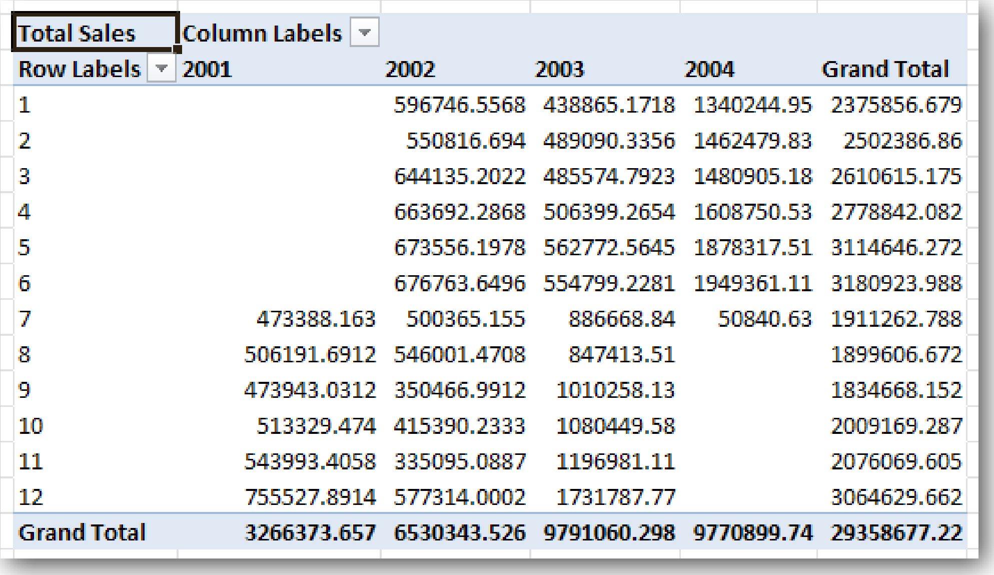
Formula:

=SUM(Sales[SalesAmt

Total Sales
29358677.22

After clicking OK the result is shown as below:

Let's do some "normal pivot" stuff. By dragging MonthNum to Rows and Year to Columns, which then breaks down the sales total by year and month number as shown below:



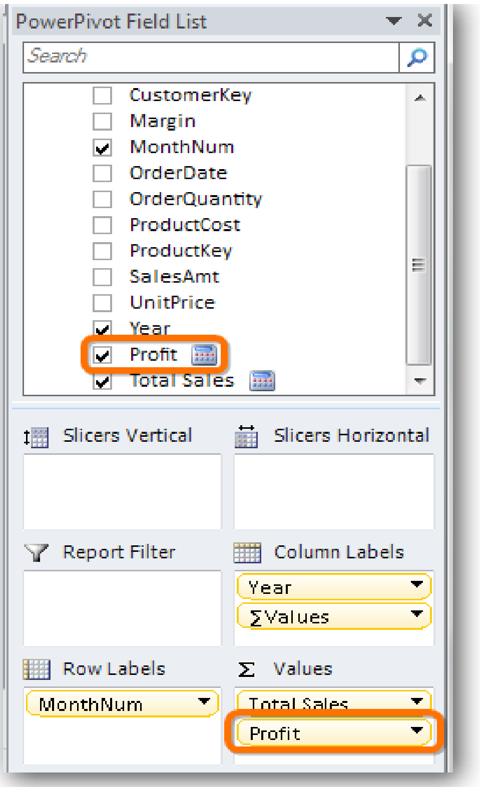
Total Sales	Column Labels				
Row Labels	2001	2002	2003	2004	Grand Total
1		596746.5568	438865.1718	1340244.95	2375856.679
2		550816.694	489090.3356	1462479.83	2502386.86
3		644135.2022	485574.7923	1480905.18	2610615.175
4		663692.2868	506399.2654	1608750.53	2778842.082
5		673556.1978	562772.5645	1878317.51	3114646.272
6		676763.6496	554799.2281	1949361.11	3180923.988
7	473388.163	500365.155	886668.84	50840.63	1911262.788
8	506191.6912	546001.4708	847413.51		1899606.672
9	473943.0312	350466.9912	1010258.13		1834668.152
10	513329.474	415390.2333	1080449.58		2009169.287
11	543993.4058	335095.0887	1196981.11		2076069.605
12	755527.8914	577314.0002	1731787.77		3064629.662
Grand Total	**3266373.657**	**6530343.526**	**9791060.298**	**9770899.74**	**29358677.22**

Referencing Measures in Other Measures

First, let’s create another simple SUM measure, for Margin:

=SUM(Sales[Margin])

Now add the new measure to the pivot field to show both the measures as shown below:



☐ CustomerKey
☐ Margin
☑ MonthNum
☐ OrderDate
☐ OrderQuantity
☐ ProductCost
☐ ProductKey
☐ SalesAmt
☐ UnitPrice
☑ Year
☑ Profit
☑ Total Sales

Column Labels: Year, ∑Values

Row Labels: MonthNum

Values: Total Sales, Profit

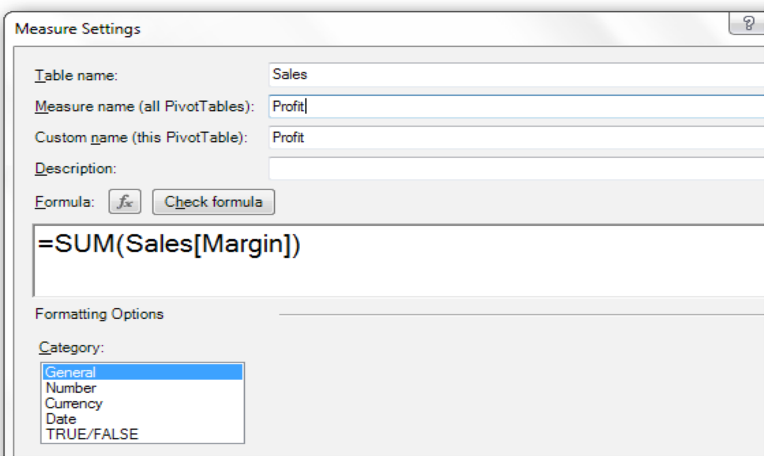


Table name: Sales

Measure name (all PivotTables): Profit

Custom name (this PivotTable): Profit

Description:

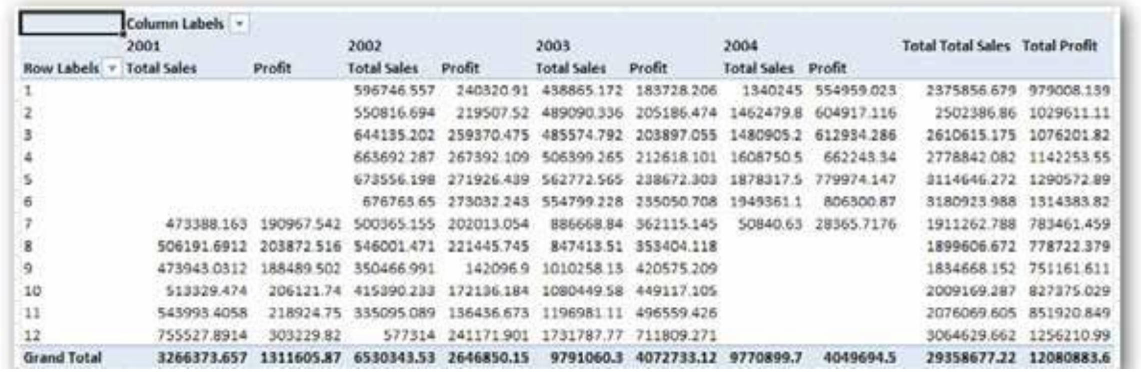
Formula:

```
=SUM(Sales[Margin])
```

Formatting Options

Category:

- General
- Number
- Currency
- Date
- TRUE/FALSE



	Column Labels									
	2001		2002		2003		2004		Total Total Sales	Total Profit
Row Labels	Total Sales	Profit	Total Sales	Profit	Total Sales	Profit	Total Sales	Profit		
1			596746.557	240320.91	438865.172	183728.206	1340245	554959.023	2375856.679	979008.139
2			550816.694	219507.52	489090.336	205186.474	1462479.8	604917.116	2502386.86	1029611.11
3			644135.202	259370.475	485574.792	203897.055	1480905.2	612934.286	2610615.175	1076201.82
4			663692.287	267392.109	506399.265	212618.101	1608750.5	662243.34	2778842.082	1142253.55
5			673556.198	271926.439	562772.565	238672.303	1878317.5	779974.147	3114646.272	1290572.89
6			676765.65	273032.243	554799.228	235050.708	1949361.1	806300.87	3180925.988	1314383.82
7	473388.163	190967.542	500365.155	202013.054	886668.84	362115.145	50840.63	28365.7176	1911262.788	783461.459
8	506191.6912	203872.516	546001.471	221445.745	847413.51	353404.118			1899606.672	778722.379
9	473943.0312	188489.502	350466.991	142096.9	1010258.13	420575.209			1834668.152	751161.611
10	513329.474	206121.74	415390.233	172136.184	1080449.58	449117.105			2009169.287	827375.029
11	543993.4058	218924.75	335095.089	136436.673	1196981.11	496559.426			2076069.605	851920.849
12	755527.8914	303229.82	577314	241171.901	1731787.77	711809.271			3064629.662	1256210.99
Grand Total	**3266373.657**	**1311605.87**	**6530343.53**	**2646850.15**	**9791060.3**	**4072733.12**	**9770899.7**	**4049694.5**	**29358677.22**	**12080883.6**

Let us now look at creating a Ratio measure. The Profit measure will be used in the new measure we are about to create. Create a new measure:

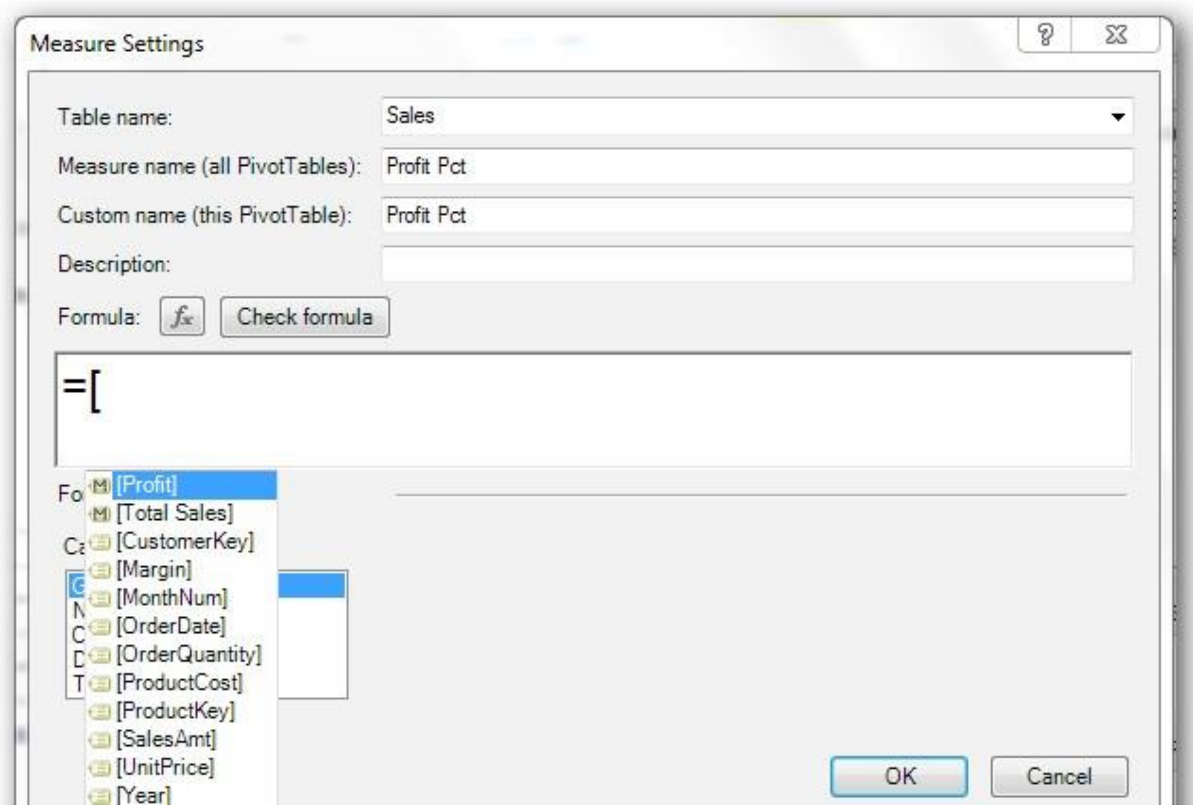


Table name: Sales

Measure name (all PivotTables): Profit Pct

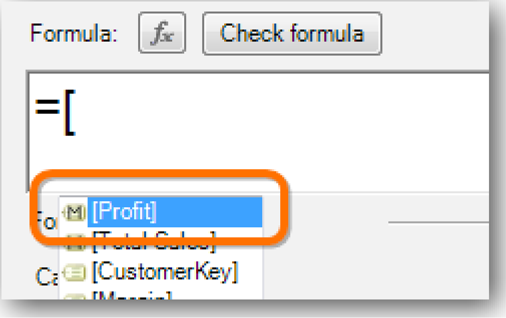
Custom name (this PivotTable): Profit Pct

Description:

Formula:

```
=[
```

Do you see the first item in the autocomplete list? This is the M symbol which represents the Profit measure. Zooming in:



Formula:

```
=[
```

There's even a little "M" icon, for measure, next to [Profit] in the autocomplete. [Total Sales] is also in there, so let's try:

=[Profit] / [Total Sales]

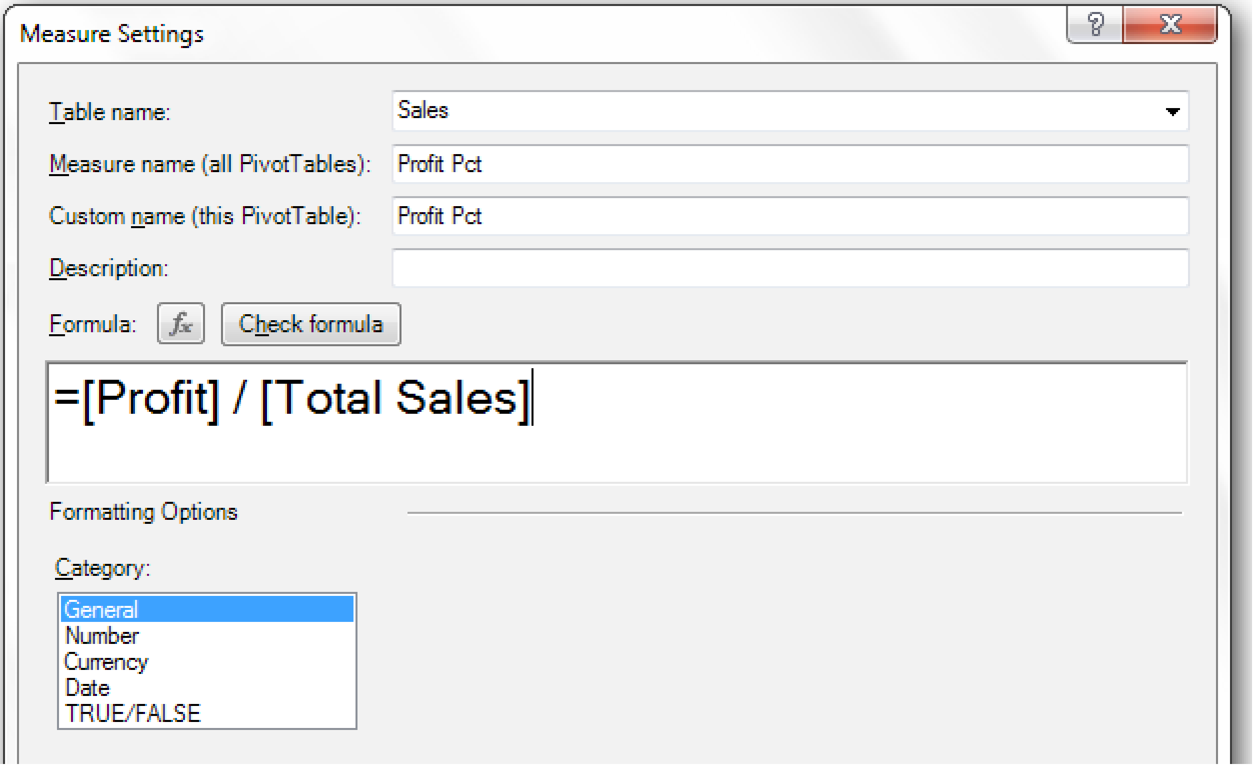
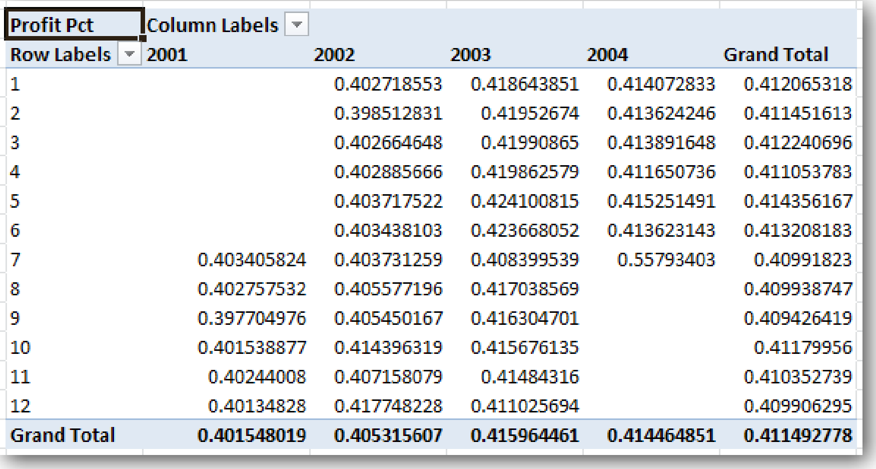


Table name: Sales
Measure name (all PivotTables): Profit Pct
Custom name (this PivotTable): Profit Pct
Description:

=[Profit] / [Total Sales]

Category: General

Now we will create a new [Profit Pct] measure, but then I'll uncheck the other two measures so we just see [Profit Pct] in the pivot:



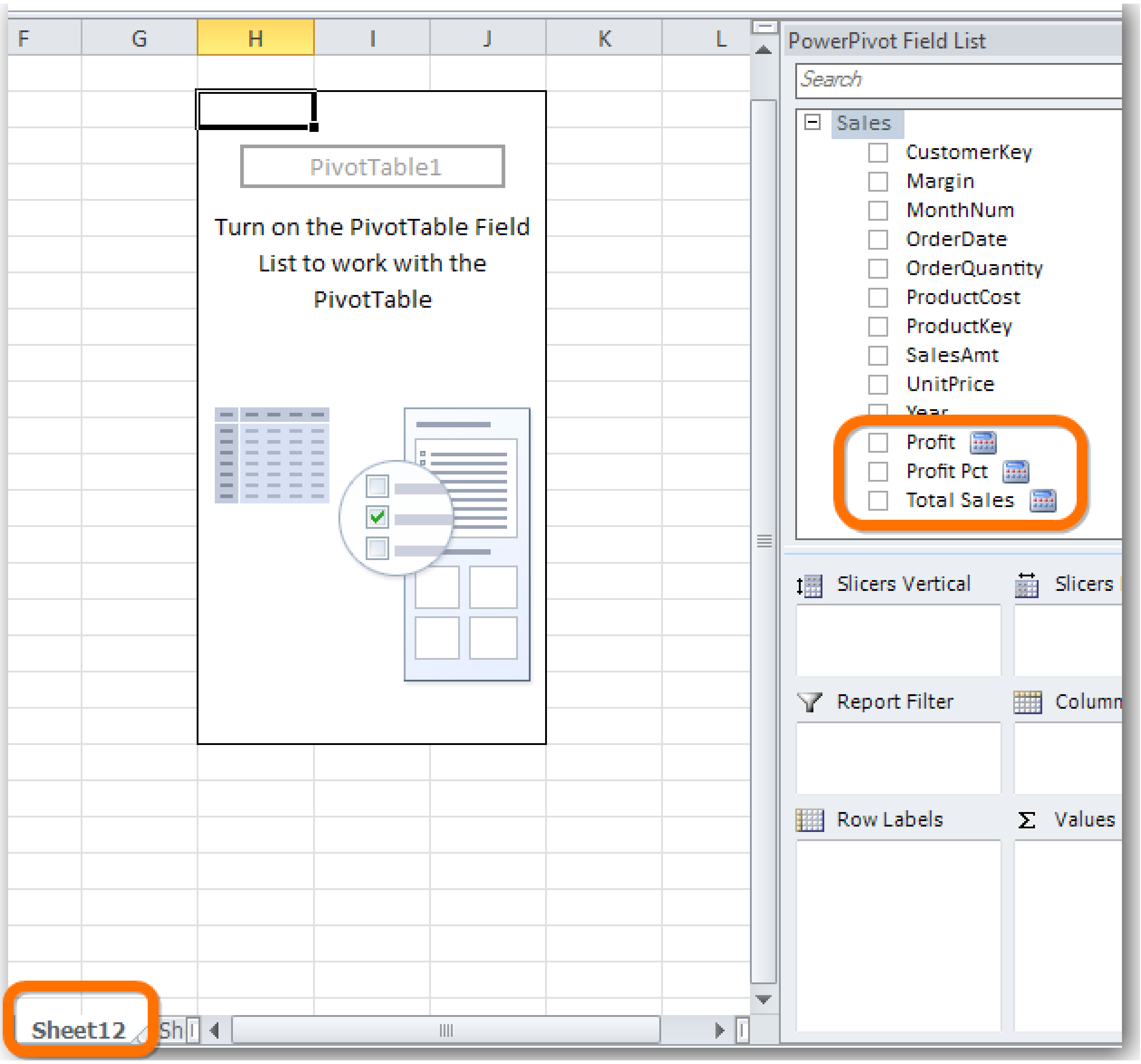
Profit Pct	Column Labels				
Row Labels	2001	2002	2003	2004	Grand Total
1		0.402718553	0.418643851	0.414072833	0.412065318
2		0.398512831	0.41952674	0.413624246	0.411451613
3		0.402664648	0.41990865	0.413891648	0.412240696
4		0.402885666	0.419862579	0.411650736	0.411053783
5		0.403717522	0.424100815	0.415251491	0.414356167
6		0.403438103	0.423668052	0.413623143	0.413208183
7	0.403405824	0.403731259	0.408399539	0.55793403	0.40991823
8	0.402757532	0.405577196	0.417038569		0.409938747
9	0.397704976	0.405450167	0.416304701		0.409426419
10	0.401538877	0.414396319	0.415676135		0.41179956
11	0.40244008	0.407158079	0.41484316		0.410352739
12	0.40134828	0.417748228	0.411025694		0.409906295
Grand Total	**0.401548019**	**0.405315607**	**0.415964461**	**0.414464851**	**0.411492778**

Other Fundamental Benefits of Measures

There are two more benefits that no chapter titled “Intro to Measures” would be complete without. Let’s cover those quickly before continuing.

Use in Any Pivot

Up until now I have just been working with a single pivot. But if I create a brand-new pivot, guess what? All of the measures I created on that first pivot are still available in my new pivot!

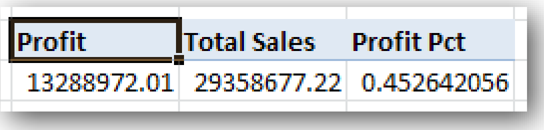


PivotTable1

Turn on the PivotTable Field List to work with the PivotTable

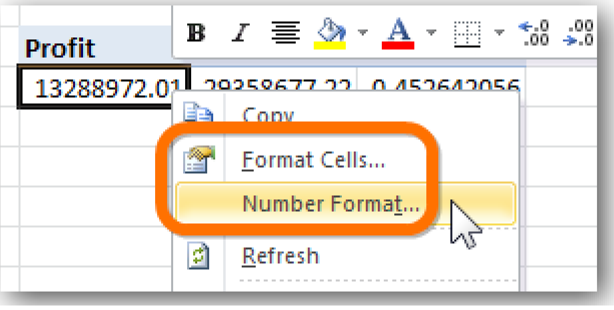
Centrally-defined Number Formatting

So far, we've been looking at ugly-formatted measures. Let's add all three measures to this new pivot to illustrate:



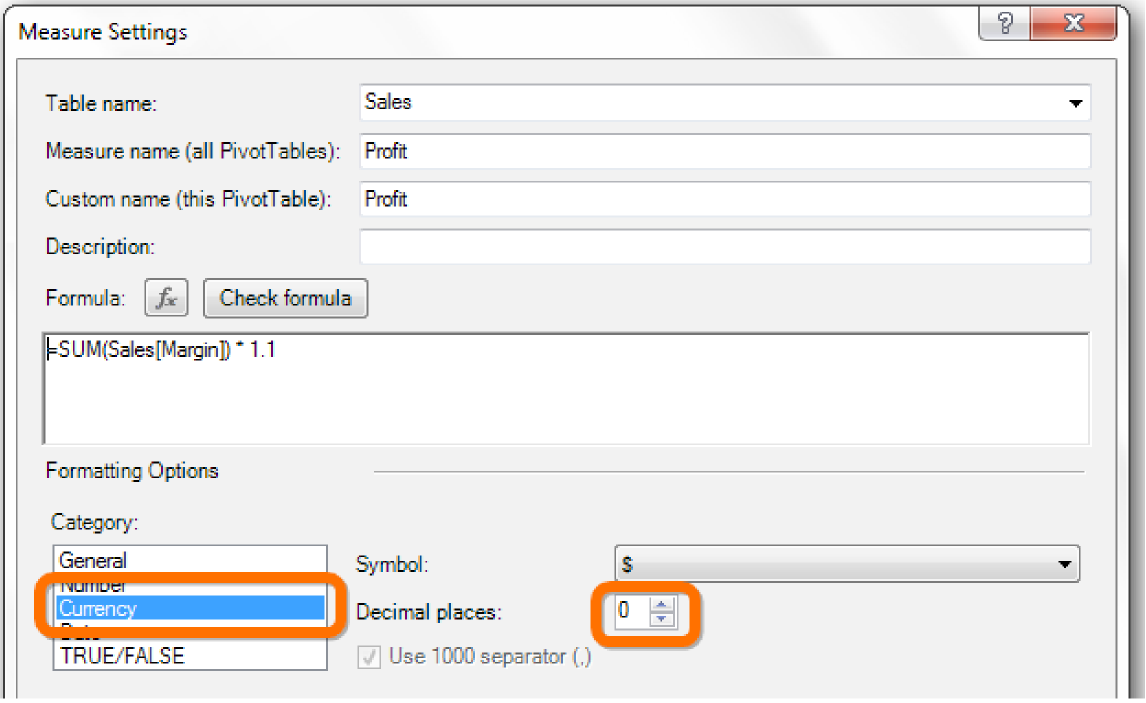
Profit	Total Sales	Profit Pct
13288972.01	29358677.22	0.452642056

I can always use Format Cells, or even better, right click on Number Format, to change this:



Profit		
13288972.01	29358677.22	0.452642056

Instead, let's bring up the measure editor for one of these measures:



Measure Settings

Table name: Sales

Measure name (all PivotTables): Profit

Custom name (this PivotTable): Profit

Description:

Formula:

```
=SUM(Sales[Margin]) * 1.1
```

Formatting Options

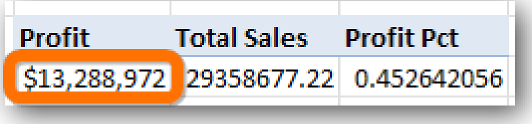
Category: Currency

Symbol: $

Decimal places: 0

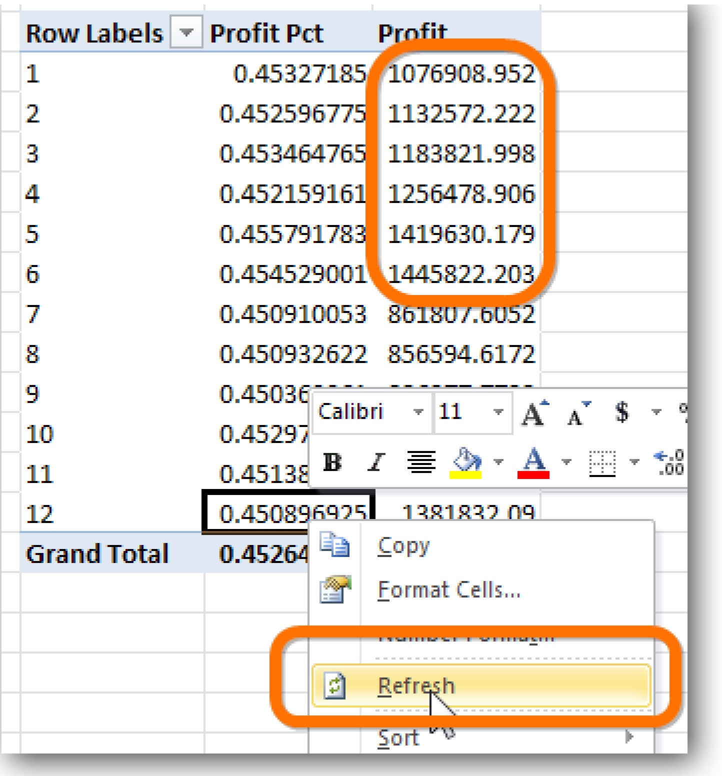
☑ Use 1000 separator (,)

The results are the same as if I had used Format Cells or Number Format:



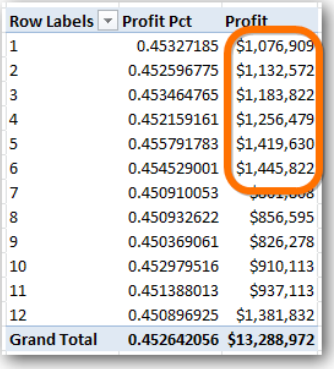
Profit	Total Sales	Profit Pct
$13,288,972	29358677.22	0.452642056

But that format now applies everywhere! Let's return to my previous pivot and Refresh it:



Row Labels	Profit Pct	Profit
1	0.45327185	1076908.952
2	0.452596775	1132572.222
3	0.453464765	1183821.998
4	0.452159161	1256478.906
5	0.455791783	1419630.179
6	0.454529001	1445822.203
7	0.450910053	861807.6052
8	0.450932622	856594.6172
9	0.45036	[illegible]
10	0.45297	
11	0.45138	
12	0.450896925	1381832.09
Grand Total	0.45264	

The pivot picks up the new formatting!



Row Labels	Profit Pct	Profit
1	0.45327185	$1,076,909
2	0.452596775	$1,132,572
3	0.453464765	$1,183,822
4	0.452159161	$1,256,479
5	0.455791783	$1,419,630
6	0.454529001	$1,445,822
7	0.450910053	[illegible]
8	0.450932622	$856,595
9	0.450369061	$826,278
10	0.452979516	$910,113
11	0.451388013	$937,113
12	0.450896925	$1,381,832
Grand Total	**0.452642056**	**$13,288,972**

Now let's set a percentage format on the [Profit Pct] measure:

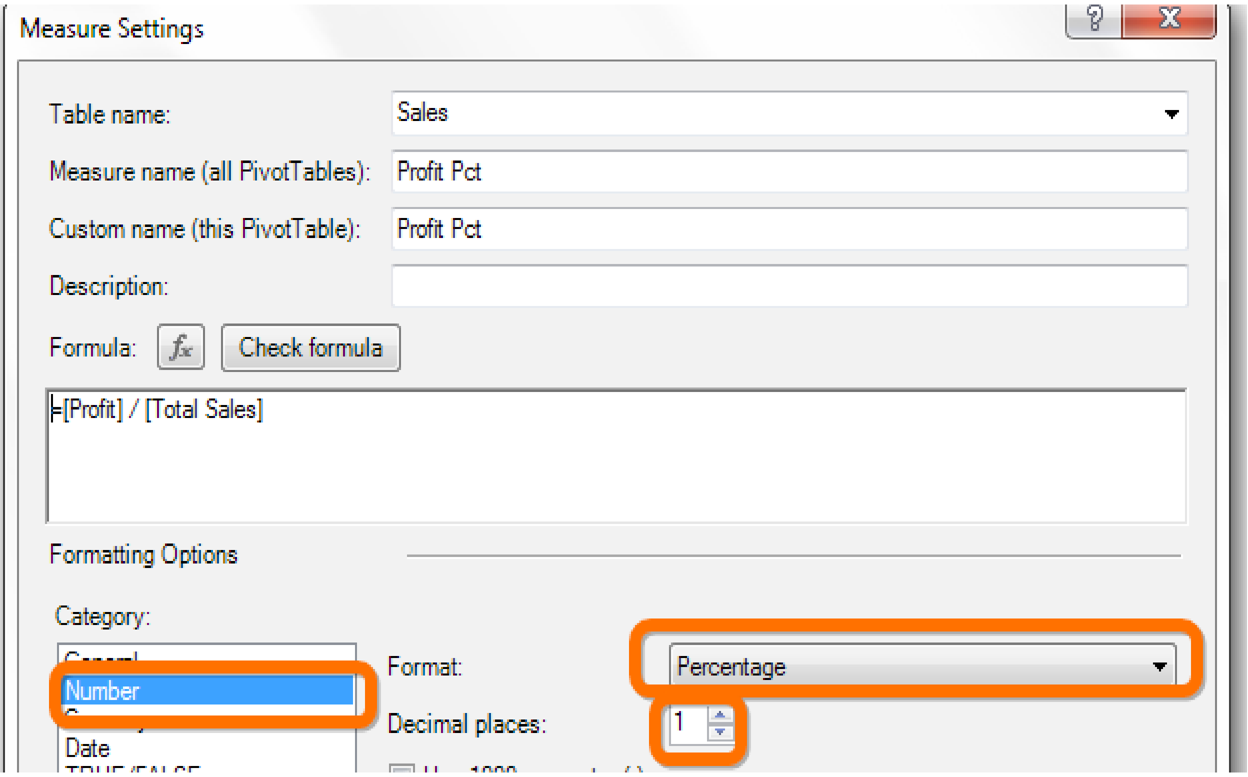


Table name: Sales

Measure name (all PivotTables): Profit Pct

Custom name (this PivotTable): Profit Pct

Description:

Formula:

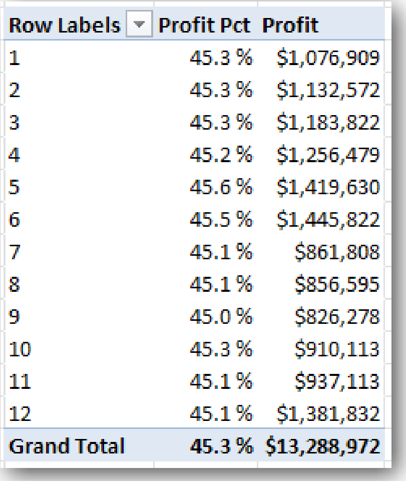
```
=[Profit] / [Total Sales]
```

Category: Number

Format: Percentage

Decimal places: 1

The results are as expected:



Row Labels	Profit Pct	Profit
1	45.3 %	$1,076,909
2	45.3 %	$1,132,572
3	45.3 %	$1,183,822
4	45.2 %	$1,256,479
5	45.6 %	$1,419,630
6	45.5 %	$1,445,822
7	45.1 %	$861,808
8	45.1 %	$856,595
9	45.0 %	$826,278
10	45.3 %	$910,113
11	45.1 %	$937,113
12	45.1 %	$1,381,832
Grand Total	**45.3 %**	**$13,288,972**

Measures Are “Portable Formulas”

Stop and think about that “rearrange the pivot and the formulas still work” point for a moment. Let’s say your workgroup originally requested a report that displayed Sales per Day and Sales per Transaction, grouped by Month. The formulas will update automatically.

How would you build that report in normal Excel? You couldn’t just write formulas in a pivot. You’d have to do some pretty serious formula alchemy to get it working.

Unit 11 – A selection of DAX Functions

In this unit, you will learn how to:

- Understand the DAX function CALCULATE()
- Understand the DAX function ALL()
- Understand the DAX function ALLEXCEPT()
- Understand the DAX function RELATED()
- Understand the DAX function SUMX()
- Understand the DAX function COUNTX()
- Understand the DAX function COUNTAX()

CALCULATE()

Have you ever used the Excel function SUMIF(), or perhaps its newer cousin, SUMIFS()?

I describe CALCULATE() as "the SUMIF/SUMIFS you always wish you'd had." You are going to love this function, because it works wonders.

In case you are one of the pivot pros who managed to skip SUMIF() and SUMIFS() in normal Excel, they are both very useful functions: they sum up a column you specify, but filter out rows that don't fit the filter criteria you specify in the formula. So, for instance, you can use SUMIF to sum up a column of Sales figures, but only for rows in the table where the Year column contains 2012.

Does that sound familiar? It sounds a lot like the Golden Rules from the prior chapter – "filter, then arithmetic." An interesting similarity, and CALCULATE() continues in that same tradition.

Anyway, CALCULATE() is superior to SUMIF() and SUMIFS() in three fundamental ways:

1. It has cleaner syntax. This is the smallest of the three advantages, but it feels good.
2. It is an "anything" IF, and not limited to SUM/COUNT/AVERAGE. There is no MAXIF() function in Excel for instance. It is literally unlimited – it allows you to take any aggregation function (or even a complex multi-function expression!) and quickly produce an IF version of it.

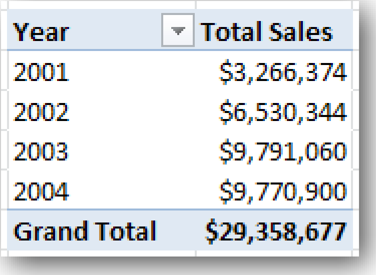
3. It can be used in pivots (as part of a measure), which normal SUMIF() cannot.

CALCULATE(<measure expression>, <filter1>, <filter2>, ...)

Ex: CALCULATE(SUM(Sales[Margin]), Sales[Year]=2001)

Ex: CALCULATE([Sales per Day], Sales[Year]=2002, Sales[ProductKey]=313)

Start with a simple pivot. Year on rows, [Total Sales] measure on values:

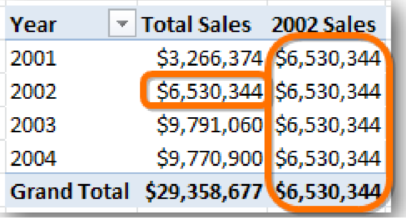


Year	Total Sales
2001	$3,266,374
2002	$6,530,344
2003	$9,791,060
2004	$9,770,900
Grand Total	**$29,358,677**

OK, let's add a new measure, one that is always filtered to Year=2002:

CALCULATE([Total Sales], Sales[Year]=2002)

The results are shown below;

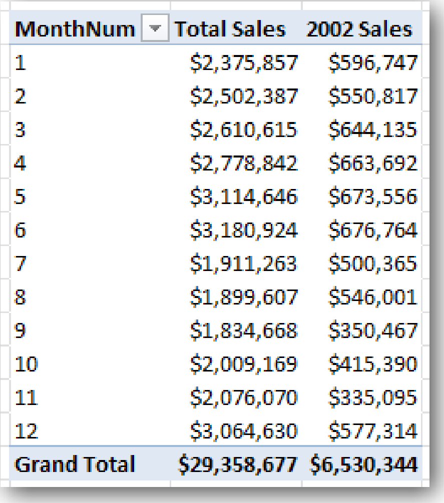


Year	Total Sales	2002 Sales
2001	$3,266,374	$6,530,344
2002	$6,530,344	$6,530,344
2003	$9,791,060	$6,530,344
2004	$9,770,900	$6,530,344
Grand Total	**$29,358,677**	**$6,530,344**

Do those results surprise you? I bet they are close to what you expected, but maybe not exactly. You might have expected years 2001 and 2003 to display zeroes for our new measure, and you might be scratching your head a bit about the grand total cell, but otherwise, having the new measure always return the 2002 value from the original measure is probably pretty instinctive.

It's not very often that I write a CALCULATE measure that filters against a column that is also on the pivot (Sales[Year] in this case). That seldom makes any real-world sense. I just started out like this so you can see that the $6,530,344 number matches up.

So, to make this a bit more realistic, let's take Year off of the pivot and put MonthNum on there instead:



MonthNum	Total Sales	2002 Sales
1	$2,375,857	$596,747
2	$2,502,387	$550,817
3	$2,610,615	$644,135
4	$2,778,842	$663,692
5	$3,114,646	$673,556
6	$3,180,924	$676,764
7	$1,911,263	$500,365
8	$1,899,607	$546,001
9	$1,834,668	$350,467
10	$2,009,169	$415,390
11	$2,076,070	$335,095
12	$3,064,630	$577,314
Grand Total	$29,358,677	$6,530,344

This probably makes even more sense than the prior pivot. The grand total is still that $6.5M number, but every other cell returns a distinct number – the sales from 2002 matching the MonthNum from the pivot.

How CALCULATE() Works

Now that we've looked at a couple of examples, let's examine how CALCULATE() truly works, because that will clear up the handful of somewhat unexpected results in that first example.

There are three key points to know about CALCULATE(), specifically about the <filter> arguments:

1. The <filter> arguments operate during the "filter" phase of measure calculation. They modify the filter context provided by the pivot – this happens before the filters are applied to the source tables, and therefore also before the arithmetic phase.

2. If a <filter> argument acts on a column that IS already on the pivot, it will override the pivot context for that column. So, in our first example above, the pivot is "saying" that Sales[Year]=2001, but I have Sales[Year]=2002 in my CALCULATE(), so the pivot's "opinion" of 2001 is completely overridden by CALCULATE(), and becomes 2002. That is why even the 2001 and 2003 cells (and the grand total cell) in the first example returned the 2002 sales number.

3. If a <filter> argument acts on a column that is NOT already on the pivot, that <filter> will purely *add* to the filter context. In our second example, where we had Sales[MonthNum] on the pivot but not Sales[Year], the Sales[Year]=2002 filter was applied on top of the Month context coming in

from the pivot, and so we received the intersection – 2002 sales for month 1, 2002 sales for month 2, etc.

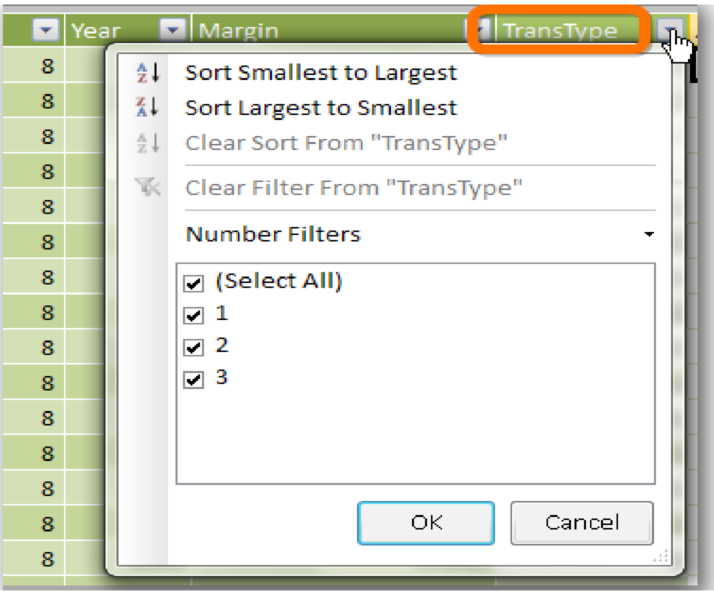
Examples of CALCULATE()

The [2002 Sales] measure that I have been using as an example so far is a good way to show you how CALCULATE() works, but it might not seem terribly useful. So, let me show you two quick examples that are much more broadly applicable.

Transactions of a Certain Type

Here is one that I see all the time in the retail sales business: not all transactions are normal sales. Some businesses record many different transaction types including “Normal Transaction,” “Refund,” and “Promotional Sales Transaction.”

My database has a column for that, so I went ahead and imported it into my Sales table (using Table Properties). Here, we see that it has three values:



I now want to write four new measures, defined here in English:

- "Regular" Sales – Just transactions of type 1
- "Promotional" Sales – Just transaction of type 3
- "Refunds" – transactions of type 2, expressed as a negative number
- "Net Sales" – Regular plus Promotional sales, less Refunds

Now, here are the formulas for each:

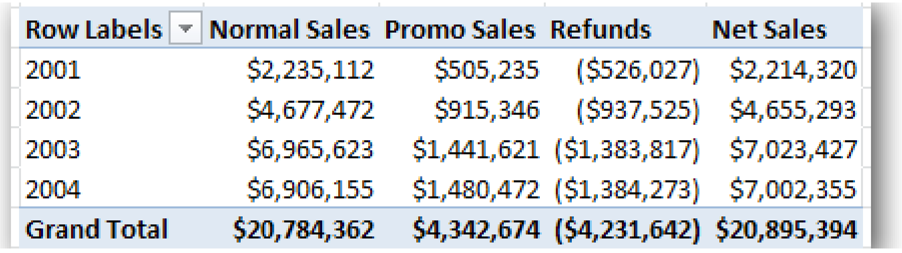
[Regular Sales] =CALCULATE([Total Sales], Sales[TransType]=1)

[Promo Sales] =CALCULATE([Total Sales], Sales[TransType]=3)

[Refunds] =CALCULATE([Total Sales], Sales[TransType]=2) * -1

[Net Sales] =[Regular Sales] + [Promotional Sales] + [Refunds]

The results are displayed below:

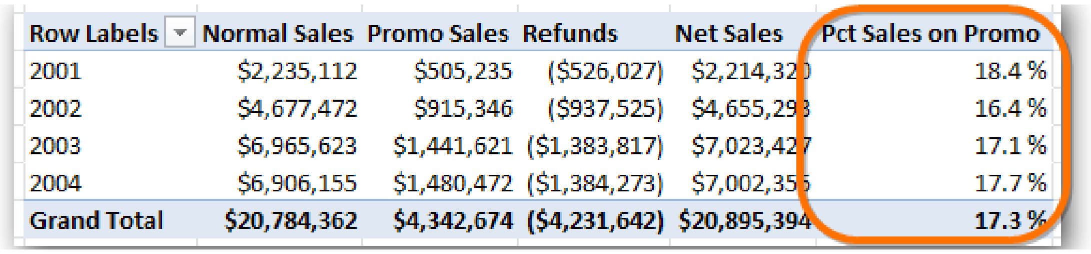


Row Labels	Normal Sales	Promo Sales	Refunds	Net Sales
2001	$2,235,112	$505,235	($526,027)	$2,214,320
2002	$4,677,472	$915,346	($937,525)	$4,655,293
2003	$6,965,623	$1,441,621	($1,383,817)	$7,023,427
2004	$6,906,155	$1,480,472	($1,384,273)	$7,002,355
Grand Total	**$20,784,362**	**$4,342,674**	**($4,231,642)**	**$20,895,394**

Let us continue down the Practical Road, let's see what percentage of our sales are due to us running promotional campaigns:

[Pct Sales on Promo] =[Promo Sales] / ([Regular Sales] + [Promotional Sales])

Results:



Row Labels	Normal Sales	Promo Sales	Refunds	Net Sales	Pct Sales on Promo
2001	$2,235,112	$505,235	($526,027)	$2,214,320	18.4 %
2002	$4,677,472	$915,346	($937,525)	$4,655,293	16.4 %
2003	$6,965,623	$1,441,621	($1,383,817)	$7,023,427	17.1 %
2004	$6,906,155	$1,480,472	($1,384,273)	$7,002,355	17.7 %
Grand Total	**$20,784,362**	**$4,342,674**	**($4,231,642)**	**$20,895,394**	**17.3 %**

Mathematical Operator in <filters>

In a <filter> argument to CALCULATE(), you are not limited to the "=" operator. You can also use:

- < (Less than)
- >(Greater than)
- <= (Less than or equal to)
- >= (Greater than or equal to)
- <>(Not equal to)

Evaluation of Multiple <filters> in a Single CALCULATE()

All of the <filter> arguments in a single CALCULATE() behave as if they are wrapped in an AND() function.

In other words, a row must match every <filter> argument in order to be included in the calculation.

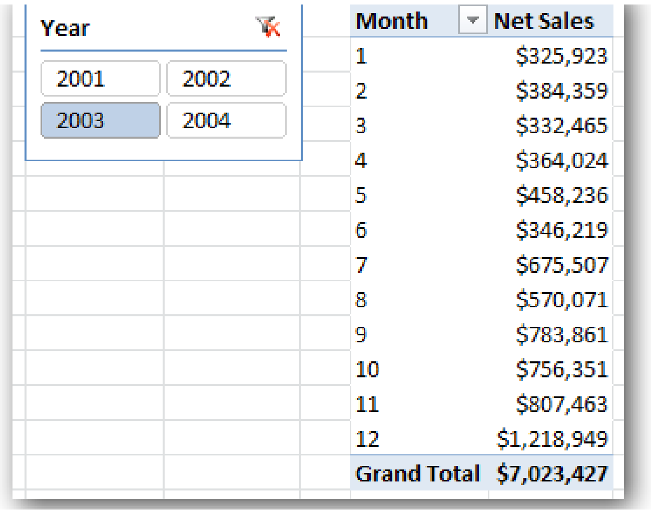
If you need an "OR()" style of operation, you can use the "||" operator. For instance:
CALCULATE([Total Sales], Sales[TransType]=1 || Sales[TransType]=3)

ALL() – The "Remove a Filter" Function

The Basics

The ALL() function is used within a CALCULATE(), as one of the <filter> arguments, to remove a filter from the filter context. Maybe we want to analyse each monthly net sales figure against the total sales ignoring the year filter applied by the year slicer.

Let's jump straight to an example. Consider the following pivot: [Net Sales] displayed by MonthNum, with Year on a slicer:

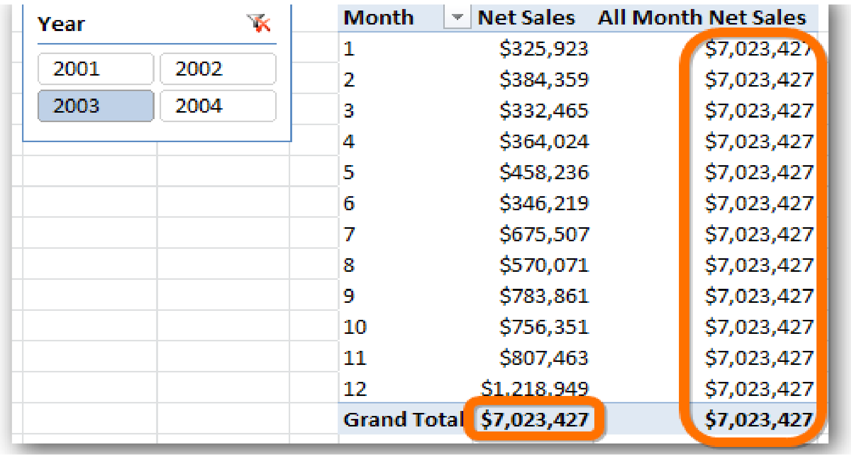


Month	Net Sales
1	$325,923
2	$384,359
3	$332,465
4	$364,024
5	$458,236
6	$346,219
7	$675,507
8	$570,071
9	$783,861
10	$756,351
11	$807,463
12	$1,218,949
Grand Total	$7,023,427

OK, time for a new measure:

[All Month Net Sales] =CALCULATE([Net Sales], ALL(Sales[MonthNum]))

And the results:



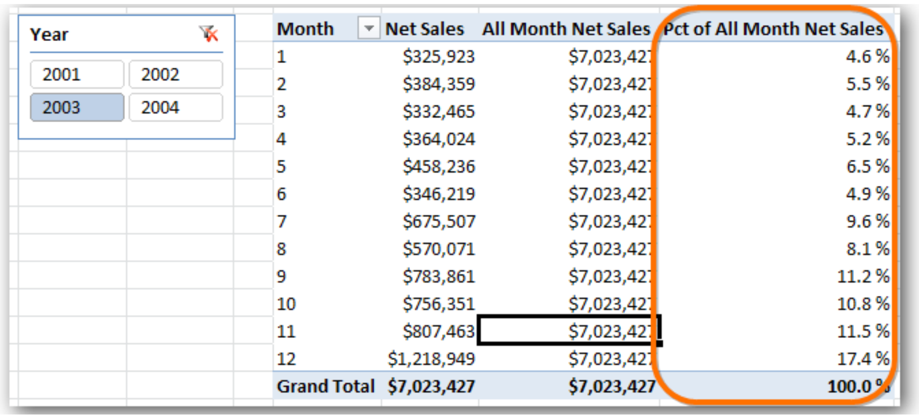
Month	Net Sales	All Month Net Sales
1	$325,923	$7,023,427
2	$384,359	$7,023,427
3	$332,465	$7,023,427
4	$364,024	$7,023,427
5	$458,236	$7,023,427
6	$346,219	$7,023,427
7	$675,507	$7,023,427
8	$570,071	$7,023,427
9	$783,861	$7,023,427
10	$756,351	$7,023,427
11	$807,463	$7,023,427
12	$1,218,949	$7,023,427
Grand Total	$7,023,427	$7,023,427

The Practical Basics

Let's do a simple ratio of the two measures already on the pivot:

[Pct of All Month Net Sales] = [Net Sales] / [Month Net Sales]

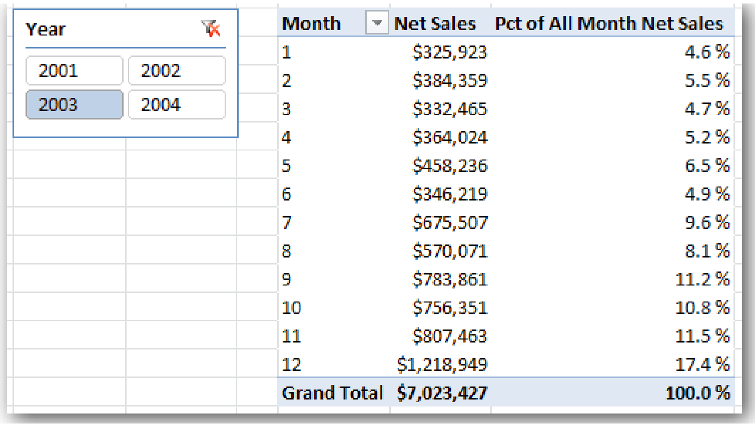
Results:



Month	Net Sales	All Month Net Sales	Pct of All Month Net Sales
1	$325,923	$7,023,427	4.6 %
2	$384,359	$7,023,427	5.5 %
3	$332,465	$7,023,427	4.7 %
4	$364,024	$7,023,427	5.2 %
5	$458,236	$7,023,427	6.5 %
6	$346,219	$7,023,427	4.9 %
7	$675,507	$7,023,427	9.6 %
8	$570,071	$7,023,427	8.1 %
9	$783,861	$7,023,427	11.2 %
10	$756,351	$7,023,427	10.8 %
11	$807,463	$7,023,427	11.5 %
12	$1,218,949	$7,023,427	17.4 %
Grand Total	$7,023,427	$7,023,427	100.0 %

We can remove the original ALL measure from the pivot and the new "pct of total" measure still works:

Negating a Slicer



Month	Net Sales	Pct of All Month Net Sales
1	$325,923	4.6 %
2	$384,359	5.5 %
3	$332,465	4.7 %
4	$364,024	5.2 %
5	$458,236	6.5 %
6	$346,219	4.9 %
7	$675,507	9.6 %
8	$570,071	8.1 %
9	$783,861	11.2 %
10	$756,351	10.8 %
11	$807,463	11.5 %
12	$1,218,949	17.4 %
Grand Total	$7,023,427	100.0 %

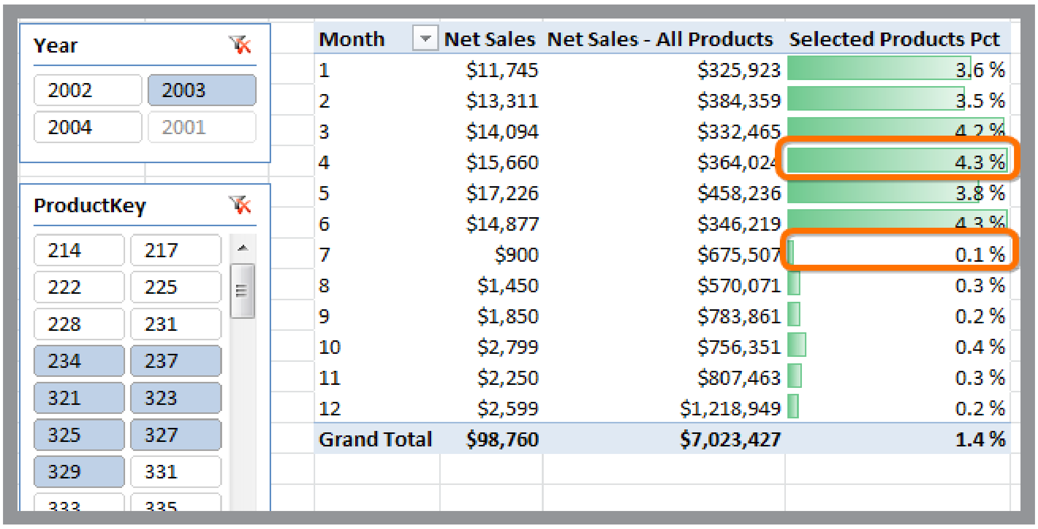
This one is useful, but also a lot of fun. Let's start with the following pivot (we just added ProductKey as a slicer, and made a few selections).

Now add a measure that ignores any filters on ProductKey:

[Net Sales - All Products] =CALCULATE([Net Sales], ALL(Sales[ProductKey]))

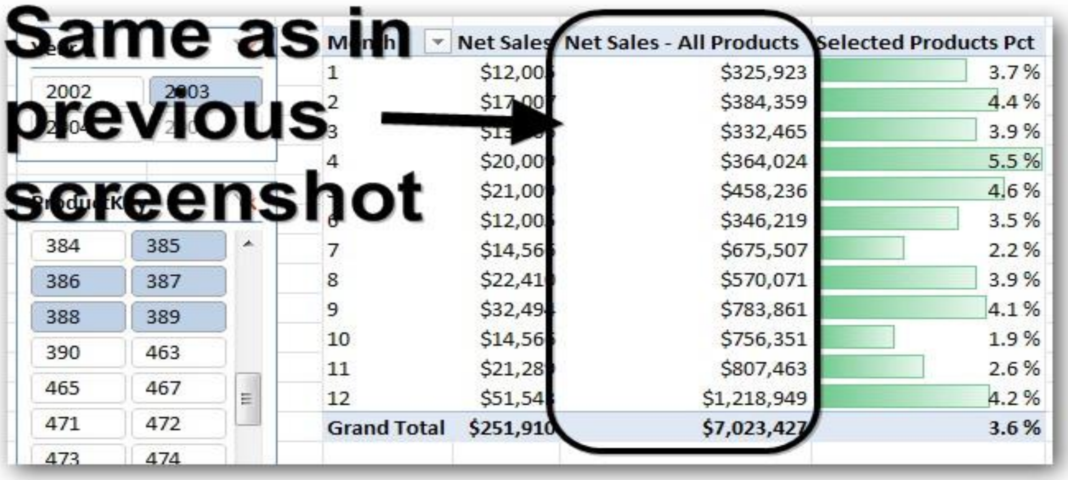
And a measure that is the ratio of that to the original [Net Sales]:

[Selected Products Pct] =[Net Sales] / [Net Sales - All Products]



Month	Net Sales	Net Sales - All Products	Selected Products Pct
1	$11,745	$325,923	3.6 %
2	$13,311	$384,359	3.5 %
3	$14,094	$332,465	4.2 %
4	$15,660	$364,024	4.3 %
5	$17,226	$458,236	3.8 %
6	$14,877	$346,219	4.3 %
7	$900	$675,507	0.1 %
8	$1,450	$570,071	0.3 %
9	$1,850	$783,861	0.2 %
10	$2,799	$756,351	0.4 %
11	$2,250	$807,463	0.3 %
12	$2,599	$1,218,949	0.2 %
Grand Total	**$98,760**	**$7,023,427**	**1.4 %**

Results:



Month	Net Sales	Net Sales - All Products	Selected Products Pct
1	$12,00	$325,923	3.7 %
2	$17,00	$384,359	4.4 %
3	$13	$332,465	3.9 %
4	$20,00	$364,024	5.5 %
	$21,00	$458,236	4.6 %
6	$12,00	$346,219	3.5 %
7	$14,56	$675,507	2.2 %
8	$22,41	$570,071	3.9 %
9	$32,49	$783,861	4.1 %
10	$14,56	$756,351	1.9 %
11	$21,28	$807,463	2.6 %
12	$51,54	$1,218,949	4.2 %
Grand Total	**$251,910**	**$7,023,427**	**3.6 %**

ALLEXCEPT()

Let's say you have 12 columns in a table, and you want to apply ALL() to 11 of the 12, but leave one of them alone.

You can then use ALLEXCEPT(<Table>, <col1 to leave alone>, <col2 to leave alone>...)

ALLEXCEPT(Sales, Sales[ProductKey])

Is the same as listing out *every* column in the Sales table *except* ProductKey:

ALL Sales[OrderQuantity], Sales[UnitPrice], Sales[ProductCost], Sales[CustomerKey], Sales[OrderDate], Sales[MonthNum], ...

RELATED()

The RELATED function is one of the most common and popular of all the dax filter functions. This table must be on the Many side of a one to many relationship. It returns a single value that is related to the current row. The syntax of the function is as follows:

=RELATED(Column)

COUNTX()

The COUNTX function counts the number of rows in a column that contain numbers or dates.

=COUNTX(Table,Expression) – This function works with Numerical data and dates.

SUMX()

The Sumx function is used when you want to sum the values in a column based on specific row criteria. For example, you may want to know the shipping costs for a particular export. The syntax for the formula is as follows:

=Sumx(Table, Expression) – Table is the table that contains the rows that need to be evaluated and the expression is the expression or filter that needs to be evaluated for each row of the table. The Sumx function will commonly use Calculatetable or Relatedtable functions in the first argument.

COUNTAX()

The COUNTAX function counts nonblank results when evaluating the result of an expression over a table. It is used to iterate through the rows in a table and count rows where the specified expressions results in a nonblank result. The syntax is as follows:

=COUNTAX(Table,Expression)

Unit 12 - Creating PivotTables, Charts & Reports

In this unit, you will learn how to:

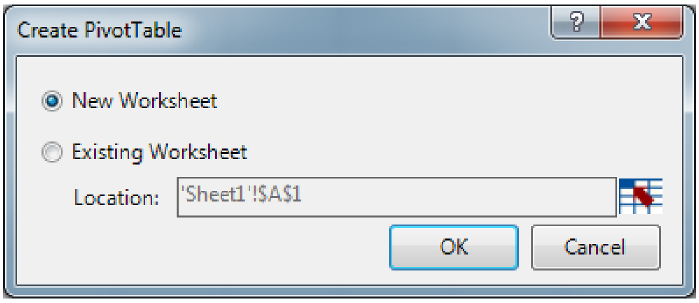
- Create a PivotTable or PivotChart Report
- Create and Change the Field Layout in a PivotTable or PivotChart Report
- Delete a PivotTable or PivotChart Report
- Filter Data using Slicers

Create a PivotTable or PivotChart Report

Once you have connected to your data source, you are now ready to create the Pivot report or chart.

The steps to create a pivot table are as follows:

1. On the Home tab, click Pivot Table and on the appropriate option
2. The PivotTable option has been selected.

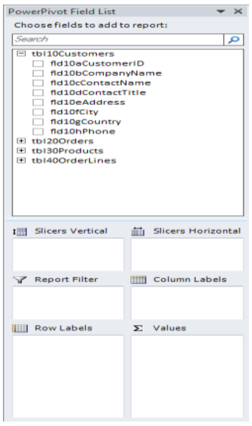

Create PivotTable

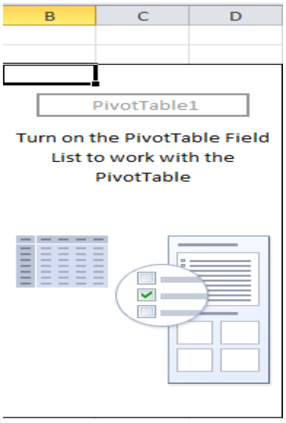
● New Worksheet

○ Existing Worksheet

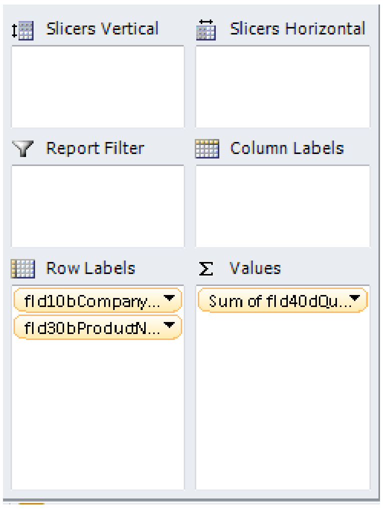
Location: 'Sheet1'!A1

3. The PivotTable will appear on the sheet with the field list on the right hand side.



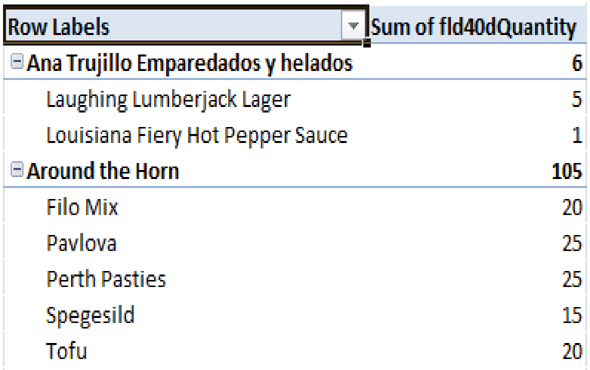

Turn on the PivotTable Field List to work with the PivotTable

4. The different areas of the PivotTable are as follows:
    1. Slicers Vertical
        - i. Used to filter the pivot table. The Slicer box appears vertically by the side of the Row headings.
    2. Slicers Horizontal
        - i. Used to filter the pivot table. The Slicer box appears horizontally above the column headings.
    3. Report Filter
        - i. This area appears in the top left of the report and is used to Filter the pivot table.
    4. Column Labels
        - i. The column labels will appear across a row in a horizontal fashion.
    5. Row Labels
        - i. The row labels will appear down a column in a vertical fashion.
    6. Values
        - i. These values will appear at the intersection point of the row and column labels displayed.
5. Drag fields from the Field List into the areas defined above to analyse the data. If you tick the checkbox next to a field in the Field list that contains text, it will be added to the Row Labels area of the Pivot table. If the field has values it will added to the Values area.



Row Labels: fld10bCompany..., fld30bProductN...

Values: Sum of fld40dQu...

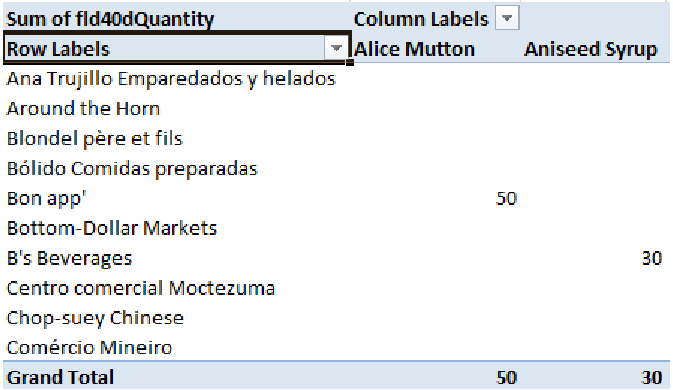


Row Labels	Sum of fld40dQuantity
⊟ Ana Trujillo Emparedados y helados	**6**
Laughing Lumberjack Lager	5
Louisiana Fiery Hot Pepper Sauce	1
⊟ Around the Horn	**105**
Filo Mix	20
Pavlova	25
Perth Pasties	25
Spegesild	15
Tofu	20

Create and Change the Field Layout in a PivotTable or PivotChart Report

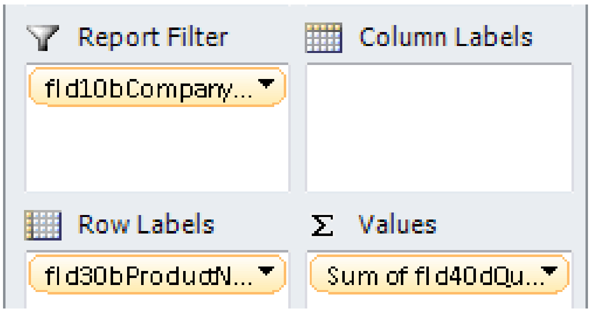
The fields that are being displayed in the Pivot can be rearranged. For instance, in the previous image, there were two row headings. It was decided to change the layout of the Pivot table to show Product Name as a column heading.

To rearrange the Pivot layout, hover over a field in one of the areas, hold the left mouse button down and drag the field to another area of the Pivot. In the example below, Product Name has been moved from the Row area to the Column area which changes the way the data is show.



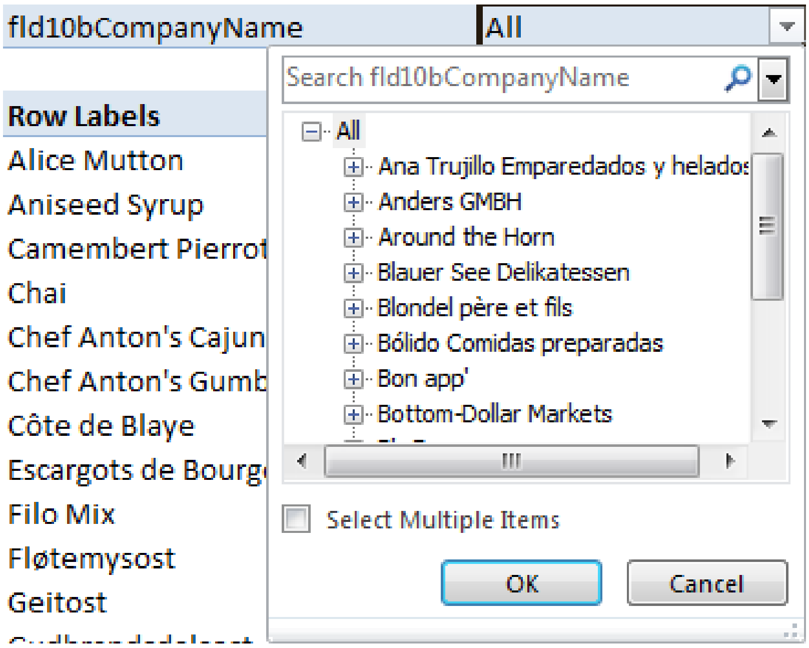
Sum of fld40dQuantity	Column Labels ▾	
Row Labels ▾	Alice Mutton	Aniseed Syrup
Ana Trujillo Emparedados y helados		
Around the Horn		
Blondel père et fils		
Bólido Comidas preparadas		
Bon app'	50	
Bottom-Dollar Markets		
B's Beverages		30
Centro comercial Moctezuma		
Chop-suey Chinese		
Comércio Mineiro		
Grand Total	50	30

If you need to filter the pivot table, you can add a Report Filter as shown below. Whatever value is selected from the Report Filter, will filter the Pivot table and show the records based on the criteria chosen.



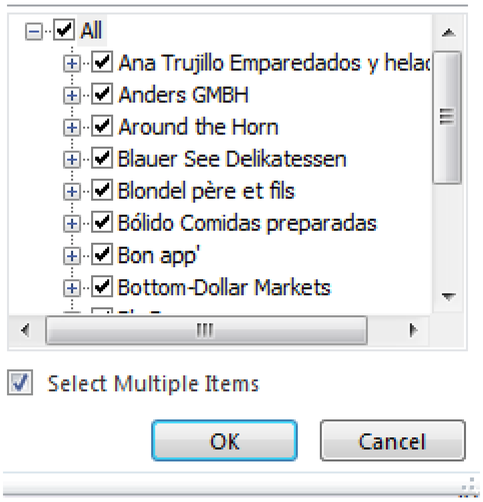
Company Name has been moved to the Report Filter field and Product Name as the Row field. When a value is selected from the Report Filter, the Pivot table headings and Data will change based on the criteria selected.

To select a value from the Report filter field, click on the filter arrow ▾ and then expand ⊞ All the list of the Company Names to select from outlined below:



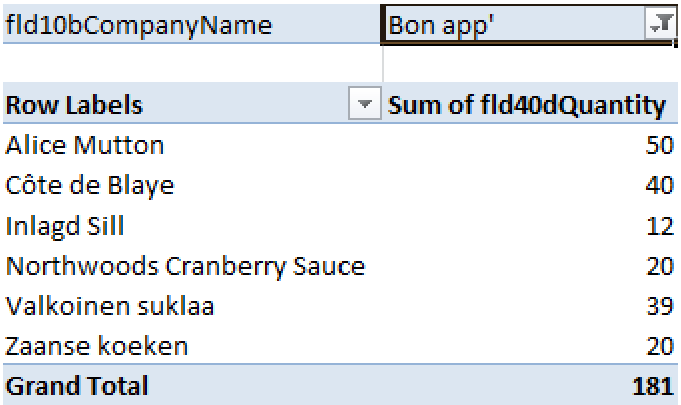
Select a value from the list by clicking once on it and then click OK. Alternatively you can use the Search box and type the name of the company you wish to filter the pivot table for.

If you need to select multiple items, click the check box for Select Multiple Items in the bottom left. This will then give you checkboxes for each company name as shown below:



Untick Select All and select the checkboxes for the companies you require.

Once you have clicked OK, the Pivot table will be updated as the image shows below:



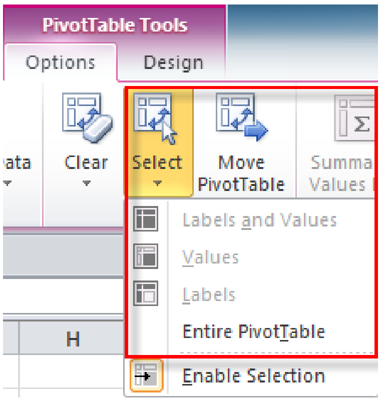
fld10bCompanyName	Bon app'
Row Labels	**Sum of fld40dQuantity**
Alice Mutton	50
Côte de Blaye	40
Inlagd Sill	12
Northwoods Cranberry Sauce	20
Valkoinen suklaa	39
Zaanse koeken	20
Grand Total	**181**

So the final result shows the sum of the quantities ordered for Bon app'.

Delete a PivotTable or PivotChart Report

If you need to delete your Pivot table go to the Options tab of the Pivot table tools tab. Follow the instructions below to delete the table:

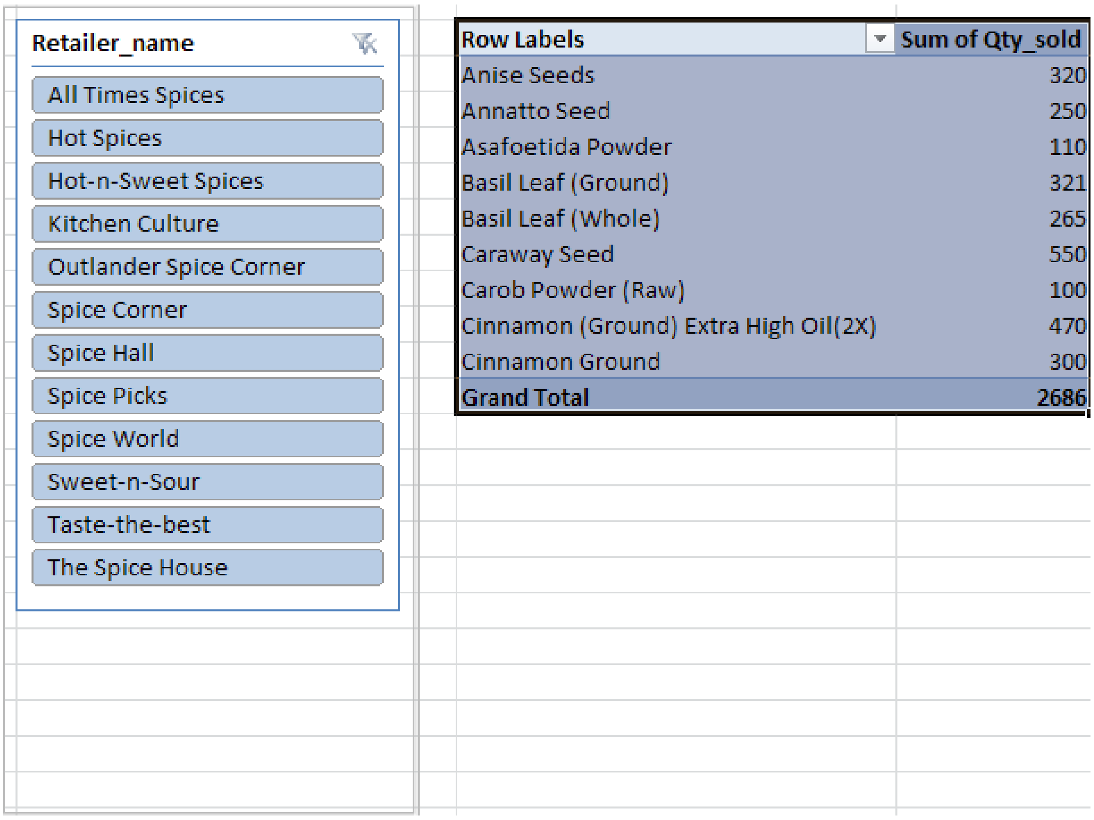
1. Go to the Actions group.



2. Click the Select button and then Entire PivotTable.
3. Press delete on the keyboard.

Filter Data using Slicers

Slicers are a new feature that was introduced in Excel 2010. Slicers are like tables containing items that are contained in a particular field and is used to filter the main pivot table based on the item or items selected.



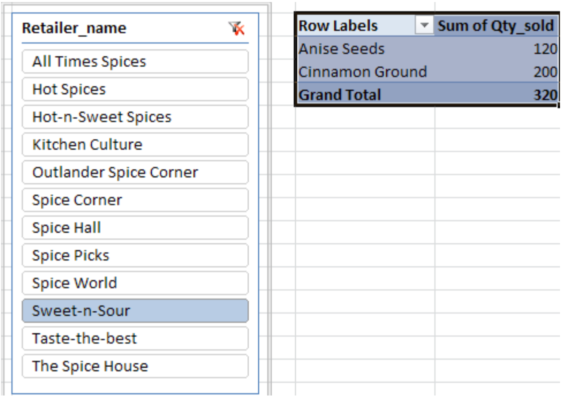
Retailer_name

All Times Spices
Hot Spices
Hot-n-Sweet Spices
Kitchen Culture
Outlander Spice Corner
Spice Corner
Spice Hall
Spice Picks
Spice World
Sweet-n-Sour
Taste-the-best
The Spice House

Row Labels	Sum of Qty_sold
Anise Seeds	320
Annatto Seed	250
Asafoetida Powder	110
Basil Leaf (Ground)	321
Basil Leaf (Whole)	265
Caraway Seed	550
Carob Powder (Raw)	100
Cinnamon (Ground) Extra High Oil(2X)	470
Cinnamon Ground	300
Grand Total	**2686**

The Retailer_name field has been added in the Slicers Vertical box.

Once a value in that box has been selected, then the Pivot table data will change based on the criteria selected (Sweet-n-Sour) as shown below:



Retailer_name

All Times Spices
Hot Spices
Hot-n-Sweet Spices
Kitchen Culture
Outlander Spice Corner
Spice Corner
Spice Hall
Spice Picks
Spice World
Sweet-n-Sour
Taste-the-best
The Spice House

Row Labels	Sum of Qty_sold
Anise Seeds	120
Cinnamon Ground	200
Grand Total	**320**

Appendix 1 – PowerView

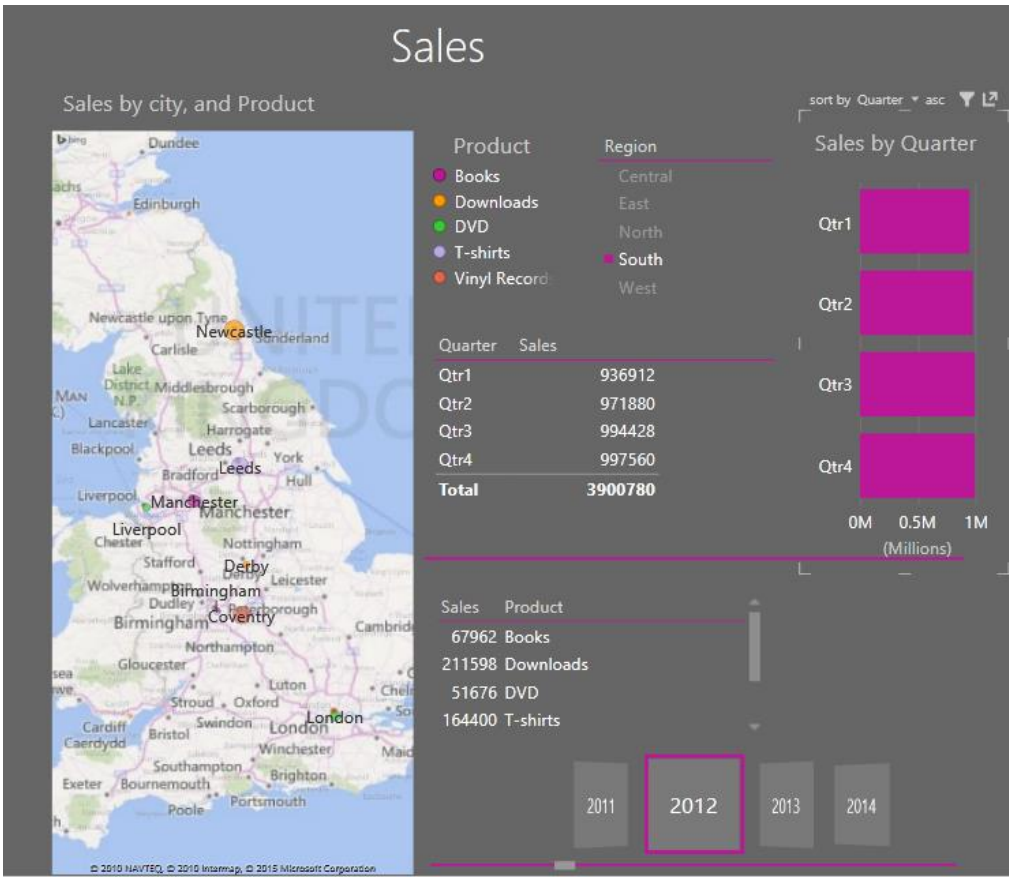
PowerView is a data visualization tool for Excel. The PowerView tool is an integrated COM Add-in in Excel 2013 but for Excel 2010 it needs to be downloaded from Microsoft's website.

Start Excel & launch the PowerView

To add the PowerView to the Insert Tab Ribbon. To achieve this, follow the instructions below:

1. Go to File and then Options.
2. On the left-hand panel, select the Add-Ins PowerView from the Manage box and click OK.
3. Tick the checkbox for the PowerView for Excel.

On the Insert tab PowerView tool will placed in the report group. A PowerView report can be generated from a single list or from a PowerPivot data model. The screenshot below is created from a single flat list in Excel. PowerView is a fast tool to create dashboard reports (BI) and the example below can be done in 10 minutes if the user has the necessary knowledge.



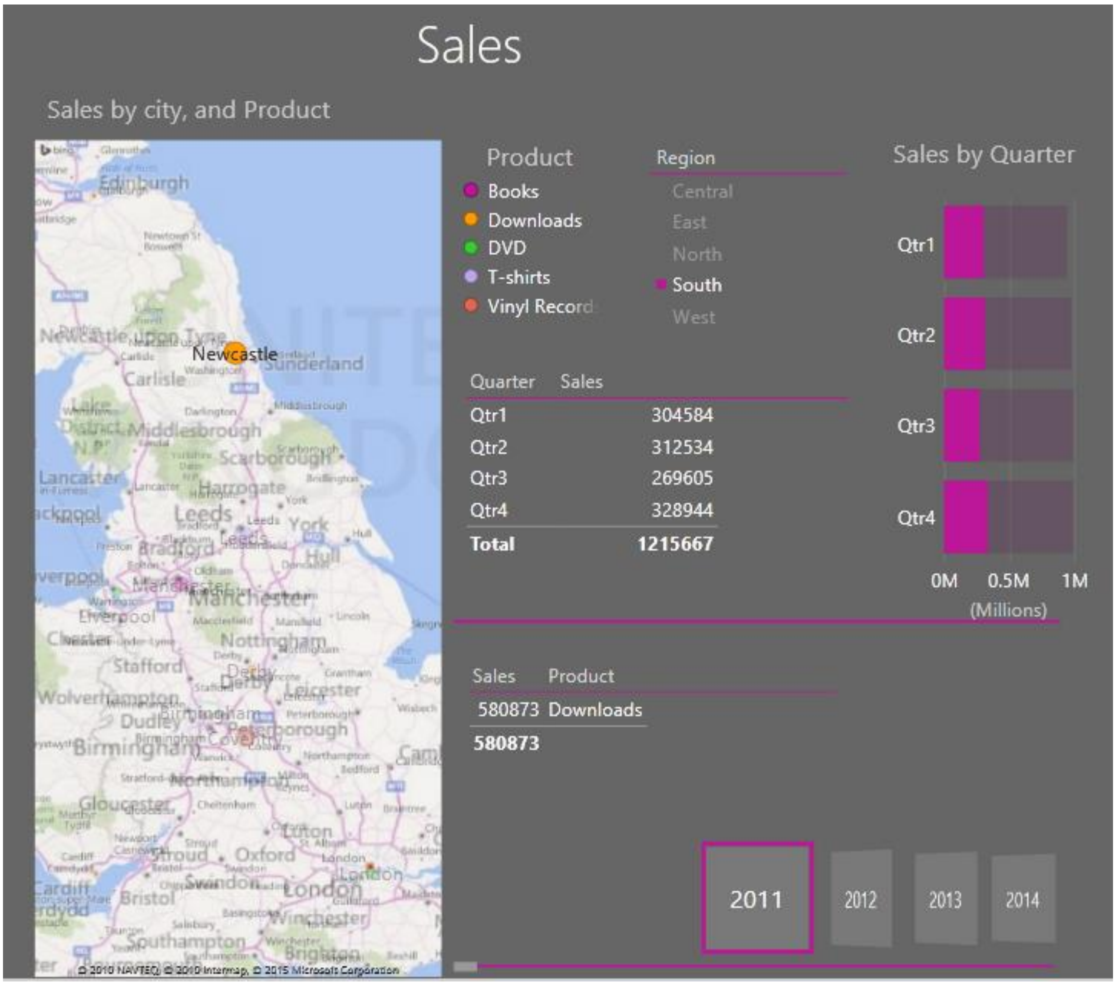
Product
- ● Books
- ● Downloads
- ● DVD
- ● T-shirts
- ● Vinyl Record

Region
- Central
- East
- North
- ■ South
- West

Quarter	Sales
Qtr1	936912
Qtr2	971880
Qtr3	994428
Qtr4	997560
Total	**3900780**

Sales	Product
67962	Books
211598	Downloads
51676	DVD
164400	T-shirts

If the source data include addresses, city names or country the PowerView tool can map the data. The PowerView tool integrate a high level of interactivity in the Report. By a click on a city the report will be filtered to only show data from the selected city. By click on a series in a chart the whole report will be filtered by the series. Slicers can add more filtering options to the report.

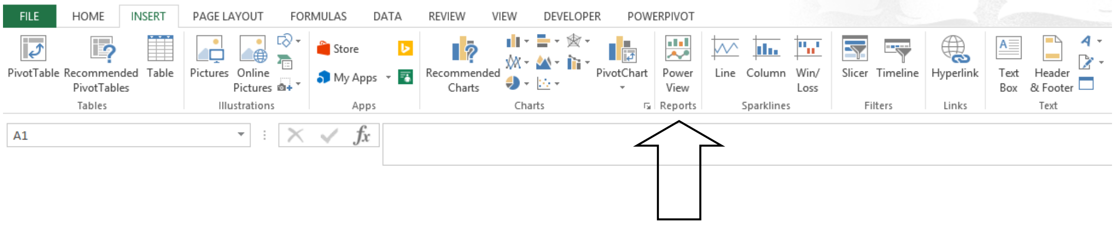


Quarter	Sales
Qtr1	304584
Qtr2	312534
Qtr3	269605
Qtr4	328944
Total	**1215667**

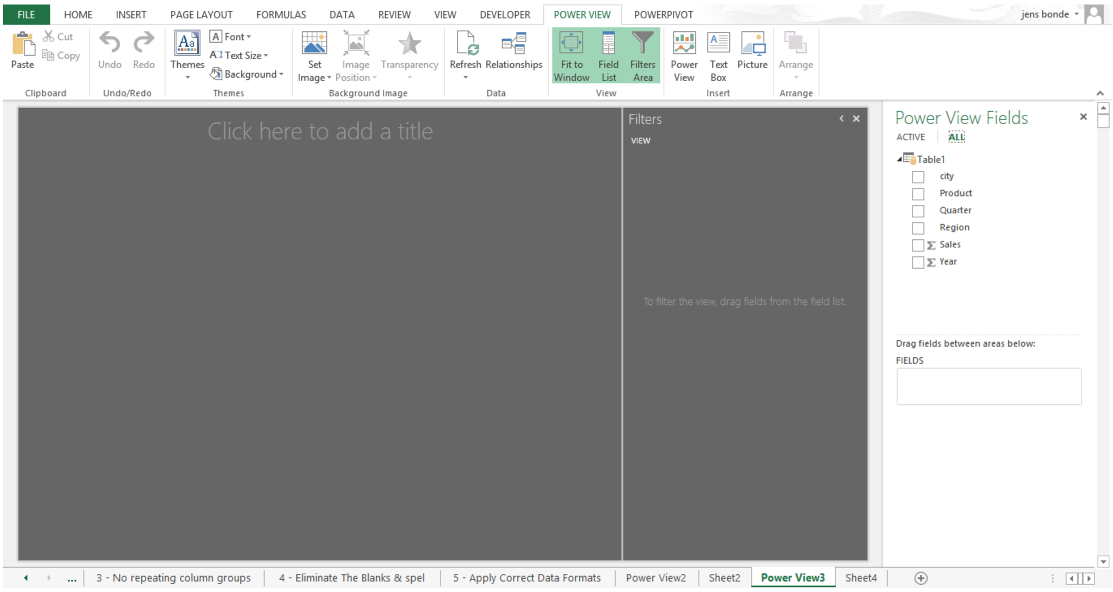
Sales	Product
580873	Downloads
580873	

How to create PowerView Report

Click the insert tab and click PowerView.

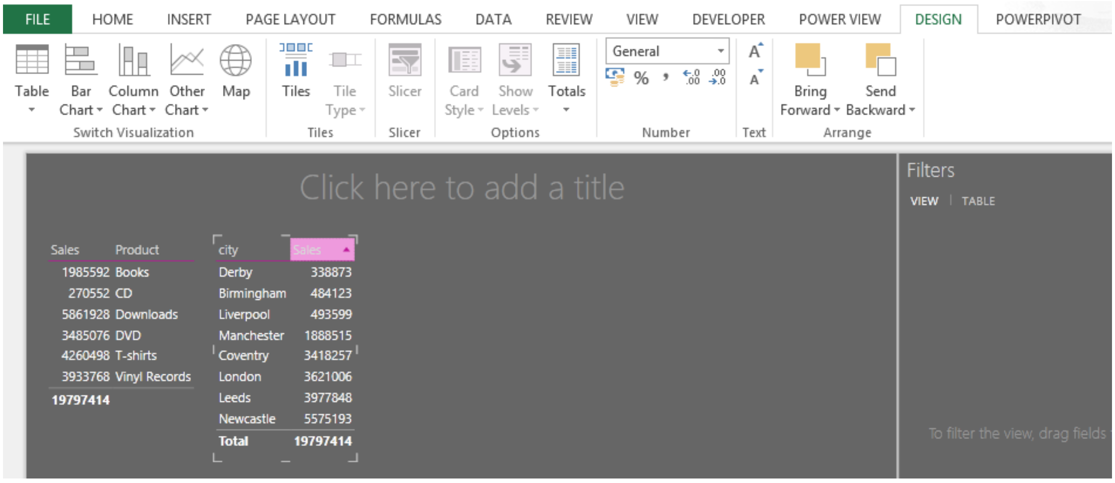


Excel will create a new worksheet and create the template for the report. The available tables will be placed to the right. Now you can drag in any field name to the report.



To filter the view, drag fields from the field list.

When fields are added to the report the Design tab will appear in the ribbon.



Sales	Product
1985592	Books
270552	CD
5861928	Downloads
3485076	DVD
4260498	T-shirts
3933768	Vinyl Records
19797414	

city	Sales
Derby	338873
Birmingham	484123
Liverpool	493599
Manchester	1888515
Coventry	3418257
London	3621006
Leeds	3977848
Newcastle	5575193
Total	**19797414**

To filter the view, drag fields

From the Design tab charts, slicers and maps can created,