Leveraging the BI Features in Excel 2016

Lab Time: 60 minutes

Lab Folder: C:\Student\Modules\BiFeaturesInExcel2016

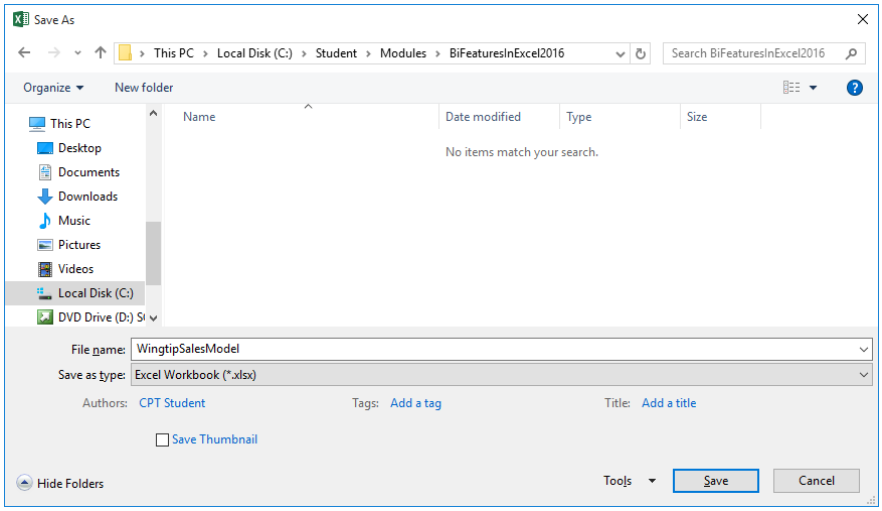
Lab Overview: In this lab, you will create a new Excel workbook and use the Power Pivot features to create a new data model that is identical to the data model in the **WingtipSalesAnalysis.pbix** project that you created with Power BI Desktop. As you go through this lab, you will notice that using the Power Query and Power Pivot features in Excel 2016 is a bit different from using the same features in Power BI Desktop. However, your basic knowledge of how Power Query and Power Pivot work should allow you to work much faster.

.Exercise 1: Creating a Workbook with a Data Model

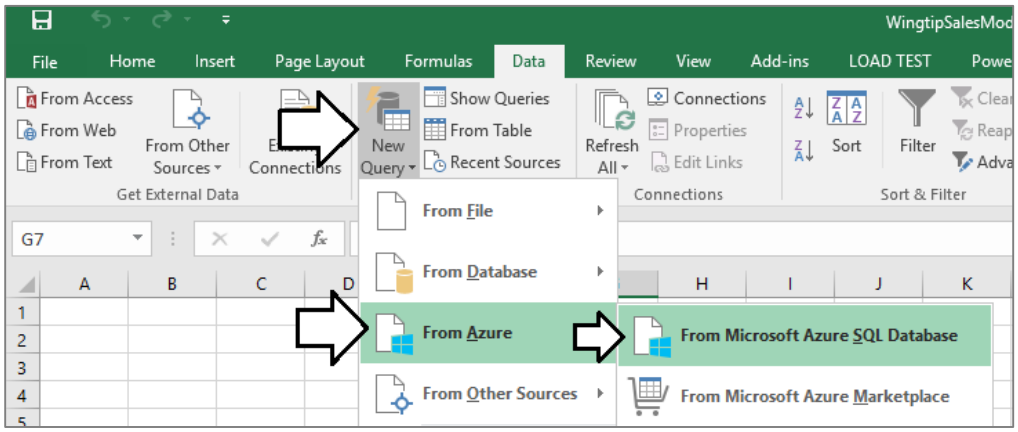
Exercise 1: Creating a New Excel Workbook and Importing Data into the Data Model

In this exercise you will launch Microsoft Excel 2013 and create a new workbook file. After that, you will use the Power Query features in Excel 2016 to import data from the Wingtip Sales database into the workbook's data model.

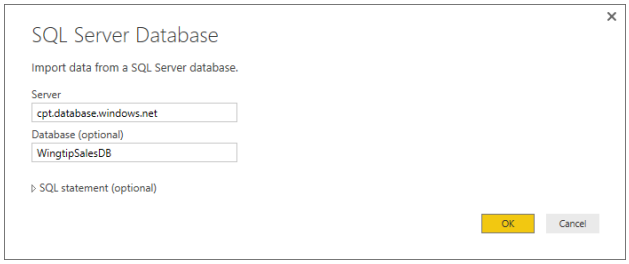
1. Launch Microsoft Excel 2016.
2. Create a new empty workbook.
3. Save the workbook file as **WingtipSalesModel.xlsx** to the path of **C:\Student\Modules\BiFeaturesInExcel2016**.



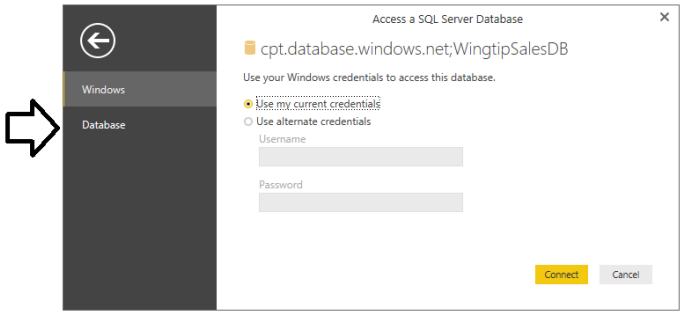
4. Import Wingtip sales data using Power Query.
   a) Activate the **Data** tab on the ribbon.
   b) Drop down the **New Query** menu and select From **Azure > From Microsoft Azure SQL Database**.



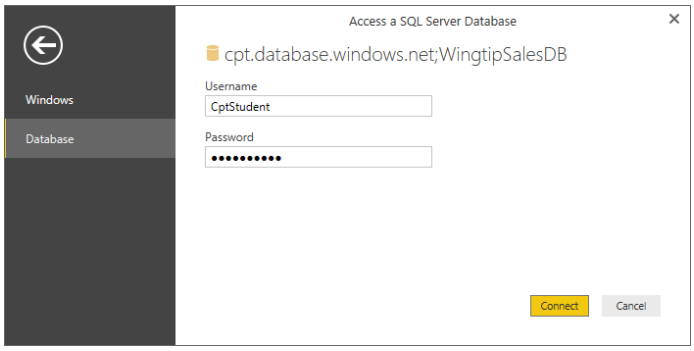
5. When you are prompted with the **SQL Server Database** dialog, enter the following values for the **Server** and **Database**.
   a) Server: **cpt.database.windows.net**
   b) Database: **WingtipSalesDB**
6. When the **SQL Server Database** dialog appears as it does in the following screenshot, click the **OK** button to continue.



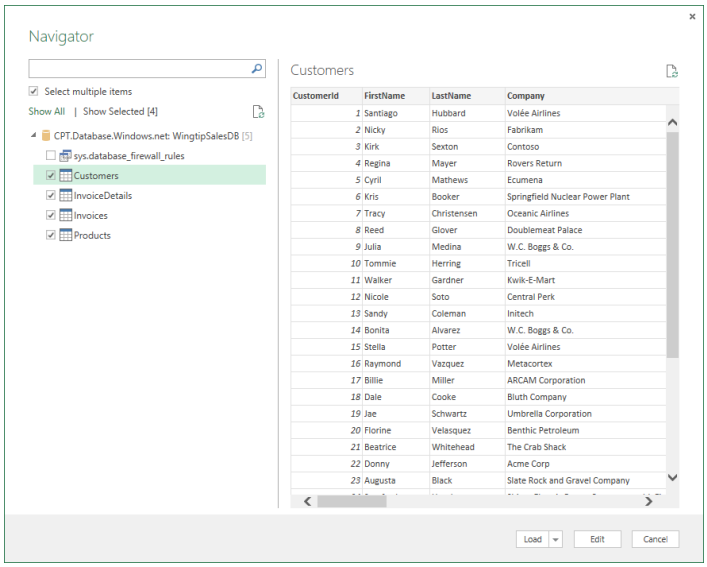
7. At this point, you will be prompted by the **Access a SQL Server Database** dialog. Click on **Database** on the left side of the dialog so that you can enter the credentials for a standard SQL account instead of using Windows authentication.



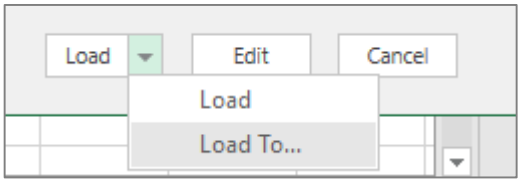
8. Enter the following credentials for a SQL user account that has been configured with read access to the database.
   a) Username: **CptStudent**
   b) Password: **pass@word1**
9. Once you have entered the credentials the **Access a SQL Server Database** dialog, click the **Connect** button to continue.



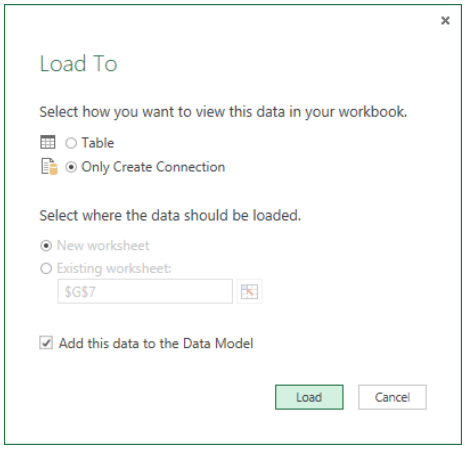
10. At this point, Excel should be able to establish a connection to the database and then prompt you with the **Navigator** dialog. The **Navigator** dialog allows you to select the tables you would like to import into your Excel workbook..
11. In the **Navigator** dialog, select the **Customers** table, the **InvoiceDetails** table, the **Invoices** table and the **Products** table as shown in the following screenshot.



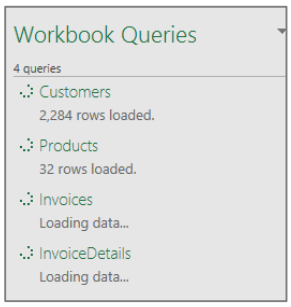
12. Look in the bottom right corner of the **Navigator** dialog and locate the **Load** drop down menu.
13. Drop down the **Load** menu and select the **Load To** command to display the **Load To** dialog.



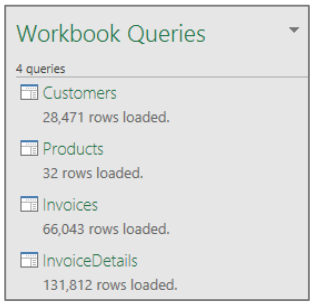
14. In the **Load To** dialog, select Only Create Connection and make sure the checkbox for the **Add this data to the Data Model** option is selected.



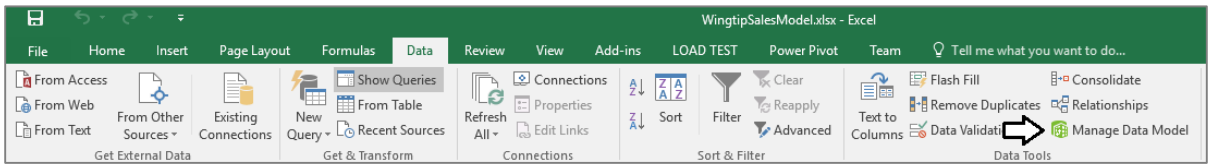
15. At this point, Excel will execute the four queries and load the data into the data model in the workbook.



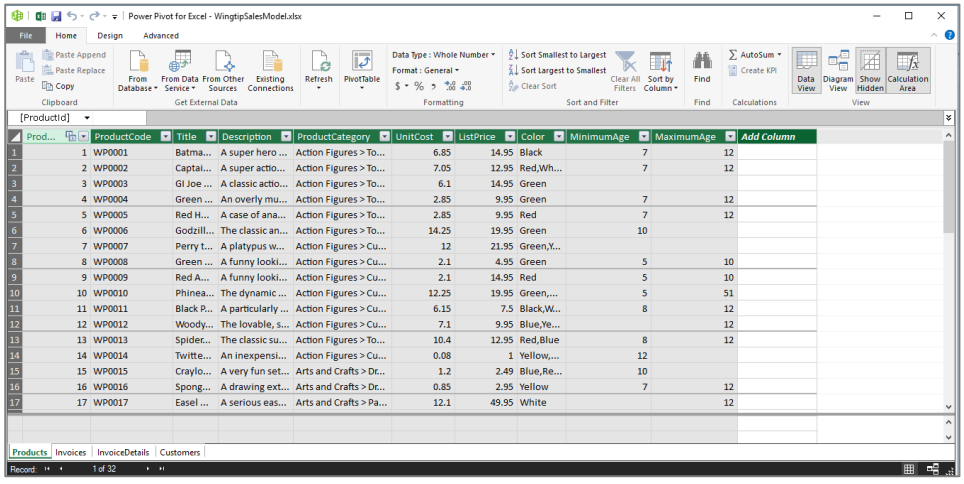
16. Once the queries have loaded their data into the data model, you can see how many rows were added to each table.



17. Inspect the tables that were added to the data model.
    a) Activate the **Data** tab in the ribbon.
    b) Click on the **Manage Data Model** button at the far right-hand side of the ribbon to display the **Power Pivot for Excel** window.

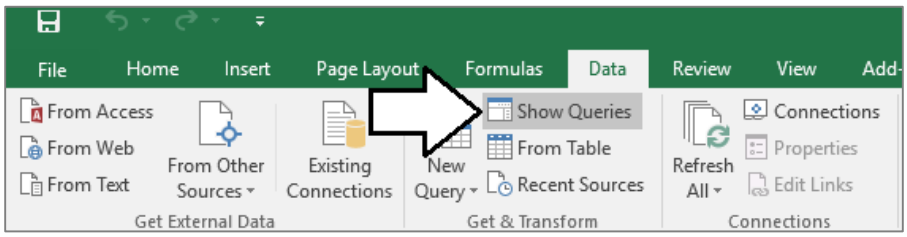


    c) When the Power Pivot windows opens, you should be able to see the four tables inside it. You can switch back and forth between the table in the data model by using the tabs at the bottom, left-hand corner of the page.

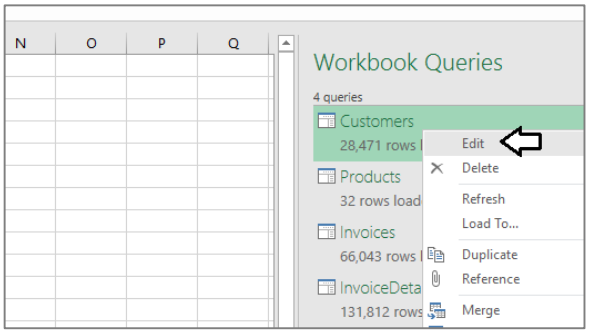


    d) Now, that you have seen the Power Pivot window, close it for now because you will need to spend more time modifying your queries and transform the imported data with Power Query before you begin to use Power Pivot to model the data .

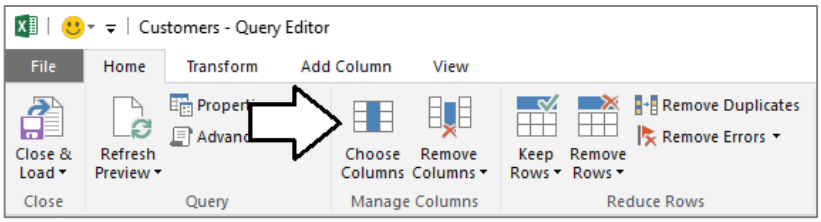
18. Make sure the **Workbook Queries** pane is showing on the right-hand side of the Excel application window.
    a) If the **Workbook Queries** pane is not showing, you can toggle it between displaying and hiding by clicking the **Show Queries** button in the **Data** tab of the ribbon.



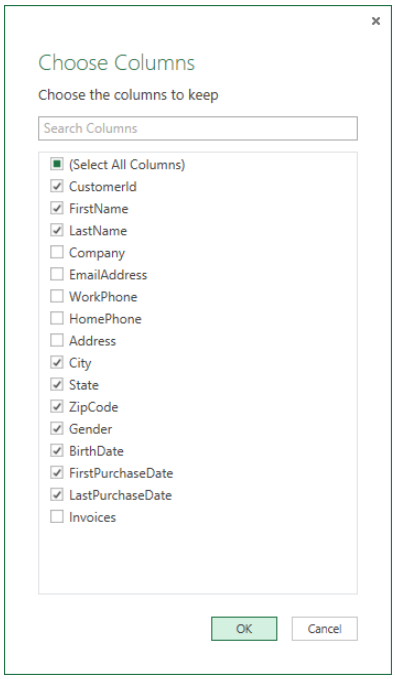
19. Right-click on the **Customers** query in the **Workbook Queries** pane and select the **Edit** command to open the query in the **Query Editor** window.



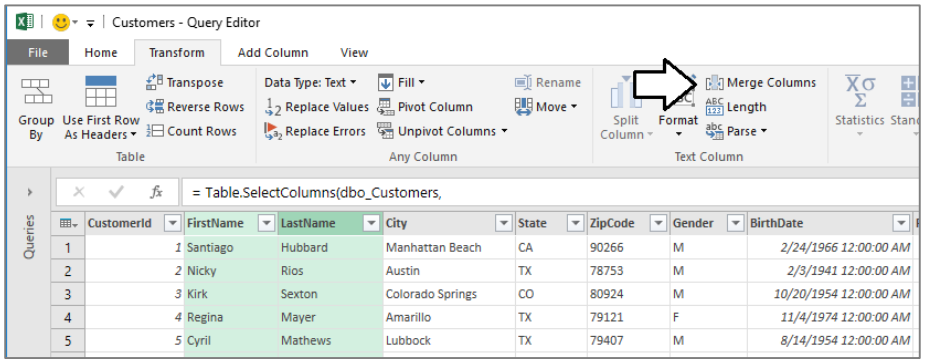
20. In the **Customers – Query Editor** window, click the **Choose Columns** button.



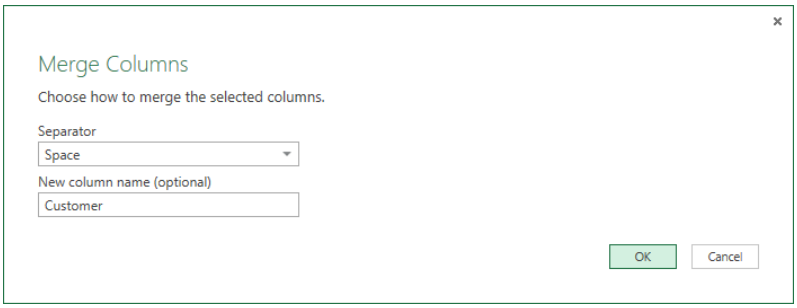
21. In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all column. Next, select the checkboxes for **CustomerId**, **FirstName**, **LastName**, **City**, **State**, **ZipCode**, **Gender**, **BirthDate**, **FirstPurchaseDate** and **LastPurchaseDate** as shown in the following screenshot. Once you have these columns selected, click the **OK** button to close the **Choose Columns** dialog and to modify the underlying query.



22. You should be able to see that the Query Editor window now only shows the columns that you selected.
23. In this step you will merge the **FirstName** column and the **LastName** column together into a single column named **Customer**.
    a) Begin by clicking on the **Transform** tab in the ribbon.
    b) Next, select the **FirstName** column by clicking on its column header.
    c) Next, hold down the **SHIFT** key and select the **LastName** column by clicking on its column header.
    d) With both the **FirstName** column and the **LastName** column selected, click the **Merge Column** button in the ribbon to display the **Merge Column** dialog.



    e) In the **Merge Column** dialog, drop down the **Separator** control and select a value of **Space**. Add a **New column name** value of **Customer** and click the **OK** button to modify the underlying query with your changes.



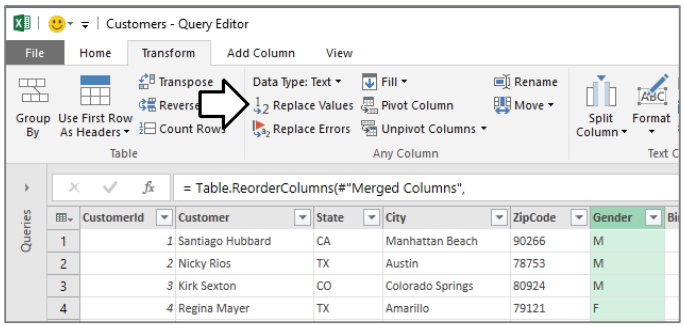
f) You should now be able to see that the **FirstName** column and the **LastName** column have been replaced with a single merged column named **Customer**.

	CustomerId	Customer	City
1	1	Santiago Hubbard	Manhattan Beach
2	2	Nicky Rios	Austin
3	3	Kirk Sexton	Colorado Springs
4	4	Regina Mayer	Amarillo
5	5	Cyril Mathews	Lubbock
6	6	Kris Booker	Tempe
7	7	Tracy Christensen	El Paso

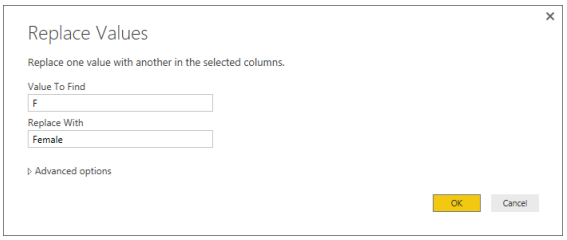
24. The **State** column is currently to the right of the **City** column. Move the **State** column so that it is repositioned to the left of the **City** column. Accomplish this by clicking on the column header for the **State** column and leaving the mouse button down. Drag the **State** column to the left of the **City** column and release the mouse button.

	CustomerId	Customer	State	City	ZipCode
1	1	Santiago Hubbard	CA	Manhattan Beach	90266
2	2	Nicky Rios	TX	Austin	78753
3	3	Kirk Sexton	CO	Colorado Springs	80924
4	4	Regina Mayer	TX	Amarillo	79121

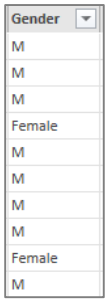
25. Modify the query so that the **Gender** column returns values of **Male** and **Female** instead of **M** and **F**.
   a) Make sure the **Transform** tab is the active tab in the ribbon.
   b) Select the **Gender** column by clicking its column header.
   c) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.



   d) In the **Replace Value** dialog, enter a value of **F** in the **Value to Find** textbox and enter a value of **Female** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.



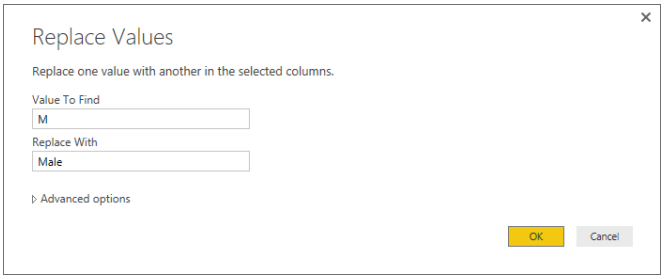
e) You should be able to see that all values of **F** in the **Gender** column have been replaced with a value of **Female**.



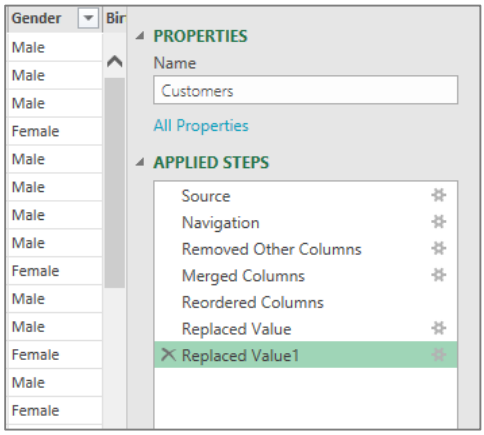
f) Make sure the **Gender** column is still selected.

g) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.

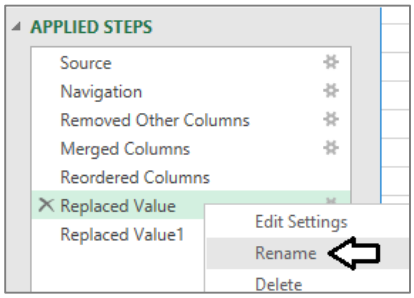
h) In the **Replace Value** dialog, enter a value of **M** in the **Value to Find** textbox and enter a value of **Male** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.



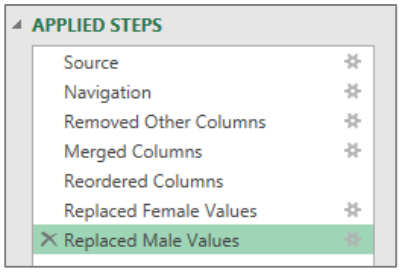
i) You should be able to confirm that all values in the **Gender** column have been replaced with a value of either **Male** or **Female**. If you inspect the **APPLIED STEPS** list in the **Query Settings** pane, you should be able to see that there are two steps at the end that have been given the generic names of **Replaced Value** and **Replaced Value 1**.



j) In order to promote higher levels of maintainability, it's often a good idea to rename steps with names such as of **Replaced Value** and **Replaced Value 1**. Rename the **Replaced Values** step by right-clicking it and clicking the **Rename** command to place the step name in edit mode. Modify the name of this step to **Replaced Female Values**.



k) Using the same technique, rename the **Replaced Value 1** step to **Replaced Male Values**.

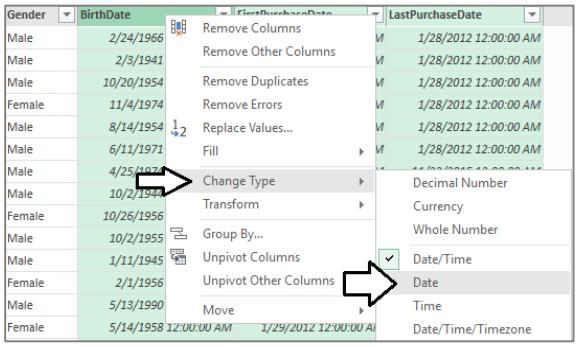


26. Change the column type of **BirthDate**, **FirstPurchasedDate** and **LastPurchasedDate** from **Date/Time** to **Date**.

a) Select the **BirthDate** column by clicking its column header.

b) Hold down the **SHIFT** key and click the column headers for **FirstPurchasedDate** and **LastPurchasedDate** so that all three columns are selected.

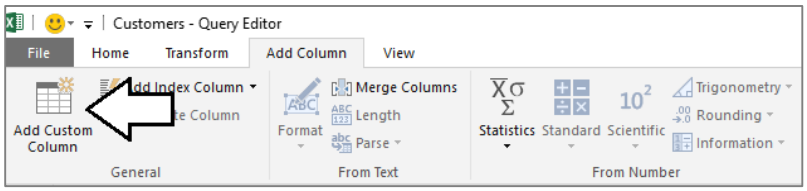
c) Right-click on any one of the selected columns and select the **Change Type > Date** command.



d) You should see that the three columns now show values with a date but without a time.

BirthDate	FirstPurchaseDate	LastPurchaseDate
2/24/1966	1/28/2012	1/28/201
2/3/1941	1/28/2012	1/28/201
10/20/1954	1/28/2012	1/28/201
11/4/1974	1/28/2012	1/28/201
8/14/1954	1/28/2012	1/28/201
6/11/1971	1/28/2012	1/28/201
4/25/1974	1/29/2012	11/22/201
10/2/1944	1/29/2012	10/2/201
10/26/1956	1/29/2012	1/29/201
10/2/1955	1/29/2012	5/6/201

27. Add a new custom column named **Customer Type** to indicates whether the customer is a repeat customer or not. You will accomplish by writing a DAX a formula which compares the **FirstPurchaseDate** column to **LastPurchaseDate** column.
    a) Begin by activating the **Add Column** tab in the ribbon.
    b) Click the **Add Custom Column** button in the ribbon to display the **Add Custom Column** dialog.



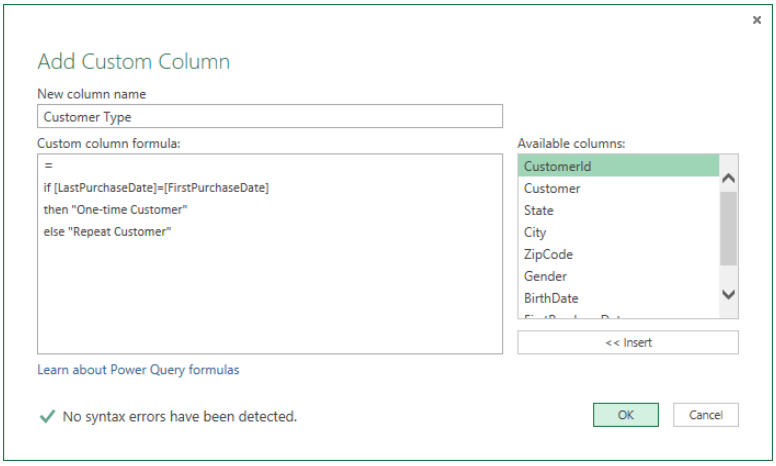
    c) In the **Add Custom Column** dialog, add a value of **Customer Type** in the **New column name** textbox.

In this particular scenario, you are working under the assumption that the customer is a repeat customer when the **FirstPurchaseDate** column and the **LastPurchaseDate** column are not equal indicating the customer has made two or more purchases.

    d) In the **Custom column formula** textbox, enter the following formula.

```
if [LastPurchaseDate]=[FirstPurchaseDate]
then "One-time Customer"
else "Repeat Customer"
```

    e) When the **Add Custom Column** dialog appears as the following screenshot, click the **OK** button to add the new column.



    f) You should be able to verify that the new **Customer Type** column has a value of **Repeat Customer** when the current customer has a **FirstPurchaseDate** column value that is not equal to the **LastPurchaseDate** column value. When these column values are equal, the **CustomerType** column has a value of **One-time Customer**.

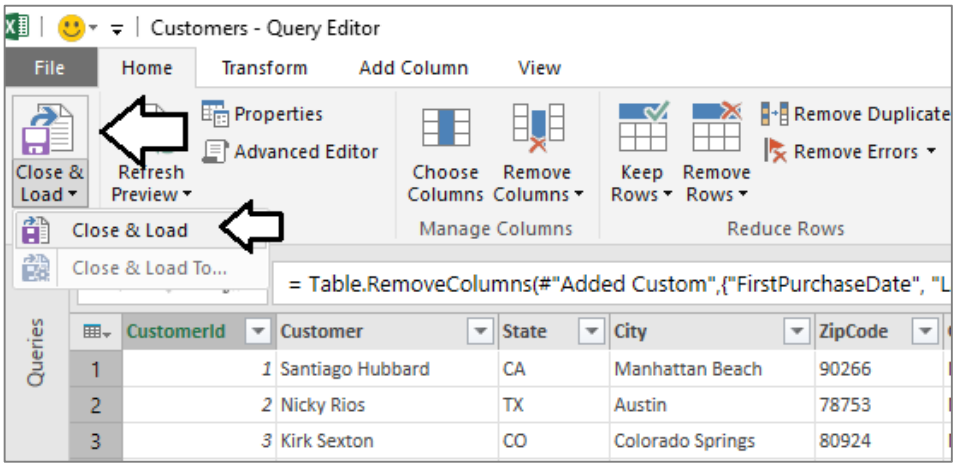
FirstPurchaseDate	LastPurchaseDate	Customer Type
1/28/2012	1/28/2012	One-time Customer
1/28/2012	1/28/2012	One-time Customer
1/28/2012	1/28/2012	One-time Customer
1/28/2012	1/28/2012	One-time Customer
1/28/2012	1/28/2012	One-time Customer
1/28/2012	1/28/2012	One-time Customer
1/29/2012	11/22/2015	Repeat Customer
1/29/2012	10/2/2015	Repeat Customer
1/29/2012	1/29/2012	One-time Customer

28. Now, that you have used the **FirstPurchaseDate** column and the **LastPurchaseDate** column to calculate the value of the **CustomerType** column, you can delete them because they are no longer needed.
    a) Activate the **Home** tab on the ribbon.
    b) Select the **FirstPurchaseDate** column by clicking its column header.
    c) Hold down the **SHIFT** key and click the column header for **LastPurchaseDate** so that both columns are selected.
    d) Click the **Remove Columns** button in the ribbon to remove both columns from the query results.
    e) You should be able to confirm that the **FirstPurchaseDate** column and the **LastPurchaseDate** columns have been removed from the query results. However, the **CustomerType** column is still there.

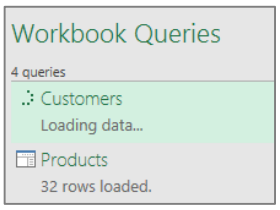
	CustomerId	Customer	State	City	ZipCode	Gender	BirthDate	Customer Type
1	1	Santiago Hubbard	CA	Manhattan Beach	90266	Male	2/24/1966	One-time Customer
2	2	Nicky Rios	TX	Austin	78753	Male	2/3/1941	One-time Customer
3	3	Kirk Sexton	CO	Colorado Springs	80924	Male	10/20/1954	One-time Customer
4	4	Regina Mayer	TX	Amarillo	79121	Female	11/4/1974	One-time Customer
5	5	Cyril Mathews	TX	Lubbock	79407	Male	8/14/1954	One-time Customer
6	6	Kris Booker	AZ	Tempe	85281	Male	6/11/1971	One-time Customer
7	7	Tracy Christensen	TX	El Paso	79936	Male	4/25/1974	Repeat Customer
8	8	Reed Glover	TX	Austin	78730	Male	10/2/1944	Repeat Customer

You are now done working with the **Customers** query.

29. Click the **Close & Load** button on the **Home** tab of the ribbon to execute the updated **Customers** query.



30. Excel will display an animated gif next to the **Customers** query while the query is executing.



31. After the query changes have been applied, you should be able to see the results of your changes in the Customers table that has been loaded into the project's data model.
32. Save the work you have done to **WingtipSalesModel.xlsx** by clicking the Save button in the upper left corner of the Excel window.
33. Right-click on the **Products** query in the **Workbook Queries** pane and select the **Edit** command to open the query in the **Query Editor** window.
34. Remove the columns that are not required in the **Products** query results.
    a) Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
    b) In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkboxes for **ProductId**, **Title**, **ProductCategory**, **UnitCost** and **ListPrice**. Once you have these columns selected, click the **OK** button to close the dialog and to modify the underlying query.
35. Rename the **Title** column to **Product**.
    a) Right-click on the **Title** column and click **Rename**.
    b) Modify the column name to **Product**.
36. Split the **ProductCategory** column up into two separate columns named **Category** and **Subcategory**.
    a) Activate the **Transform** tab on the ribbon.
    b) Select the **ProductCategory** by clicking its column header.
    c) Drop down the **Split Column** menu button and click by **By Delimiter** to display the **Split Column By Column** dialog.
    d) In the **Split Column By Column** dialog, drop down the **Select or enter delimiter** combo box and select **--Custom--**.
    e) In the textbox below the combo box, enter a three-character text value which includes a space follow by the **>** character followed by another space.
    f) When the **Split Column By Column** dialog appears as the one in the following screenshot, click the **OK** button to add the step to the **Products** query.
    g) You should be able to confirm that Excel has split the **ProductCategory** column into two separate columns named **ProductCategory.1** and **ProductCategory.2**.

ProductId	Product	ProductCategory.1	ProductCategory.2	UnitCost	ListPrice
1	Batman Action Figure	Action Figures	Tough Guys	6.85	14.95
2	Captain America Action Figure	Action Figures	Tough Guys	7.05	12.95
3	GI Joe Action Figure	Action Figures	Tough Guys	6.1	14.95
4	Green Hulk Action Figure	Action Figures	Tough Guys	2.85	9.95
5	Red Hulk Alter Ego Action Figure	Action Figures	Tough Guys	2.85	9.95
6	Godzilla Action Figure	Action Figures	Tough Guys	14.25	19.95

    h) Rename the **ProductCategory.1** column to **Category.** Also rename the **ProductCategory.2** column to **Subcategory** as shown in the following screenshot.

	ProductId	Product	Category	Subcategory
1	1	Batman Action Figure	Action Figures	Tough Guys
2	2	Captain America Action Figure	Action Figures	Tough Guys
3	3	GI Joe Action Figure	Action Figures	Tough Guys
4	4	Green Hulk Action Figure	Action Figures	Tough Guys
5	5	Red Hulk Alter Ego Action Figure	Action Figures	Tough Guys
6	6	Godzilla Action Figure	Action Figures	Tough Guys
7	7	Perry the Platypus Action Figure	Action Figures	Cute and Huggable
8	8	Green Angry Bird Action Figure	Action Figures	Cute and Huggable

When you have query columns based on numeric currency values, it is best to change their column type to **Fixed Decimal Number**.

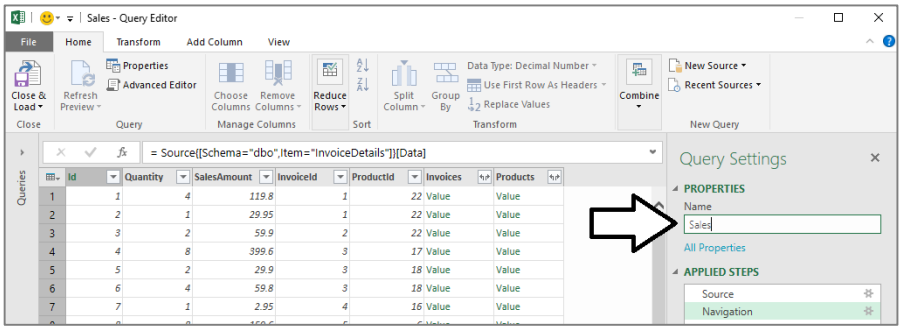
37. Modify the column type of the **UnitCost** column and the **ListPrice** column to the **Currency** type.
    a) Select the **UnitCost** column by clicking its column header.
    b) Hold down the **SHIFT** key and click the ListPrice column so that both columns are selected.
    c) Right-click either one of the selected columns and click the **Change Type > Currency** menu command.

You are now done working with the **Products** query.

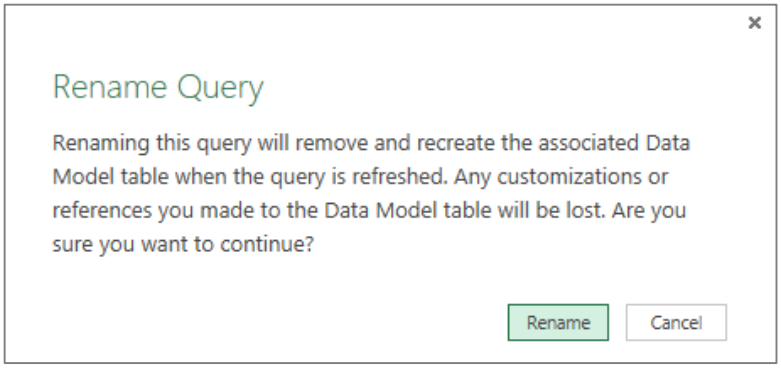
38. Click the **Close & Load** button on the **Home** tab of the ribbon to execute the updated **Products** query.
39. Save your work by clicking the Save button in the upper left corner of the Excel window.
40. Right-click on the **InvoiceDetails** query in the **Workbook Queries** pane and select the **Edit** command to open the query in the **Query Editor** window.

One important point of flexibility in the import process is that you can change the name of a query, and therefore the name of the resulting table to make the data model more intuitive and easier to understand. The **InvoiceDetails** query is returning data that will be used to calculate sales results at the most granular level. Therefore, the data model will be easier to understand if you change the name the name of the **InvoiceDetails** query to **Sales**. The will result in renaming the resulting table to the **Sales** table.

41. Update the name of the **InvoiceDetails** query to **Sales** by replacing the text in the **Name** textbox in the **Query Settings** pane.



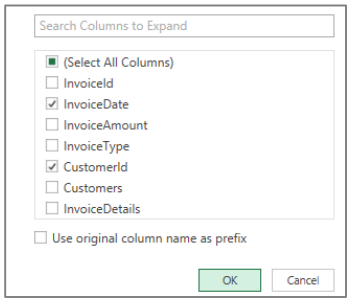
42. When prompted to rename the query, read the warning and then click the **Rename** button.



43. Using the mouse, drag and drop the column header for the **InvoiceId** column to move it to the immediate right of the **Id** column.
44. Using the mouse, drag and drop the column header for the **ProductId** column to move it to the right of the **InvoiceId** column.

Id	InvoiceId	ProductId	Quantity	SalesAmount	Invoices	Products
1	1	22	4	119.8	Value	Value
2	1	22	1	29.95	Value	Value
3	2	22	2	59.9	Value	Value
4	3	17	8	399.6	Value	Value
5	3	18	2	29.9	Value	Value
6	3	18	4	59.8	Value	Value
7	4	16	1	2.95	Value	Value

45. Modify the column type of the **SalesAmount** column to the **Currency** type.
    a) Select the **SalesAmoungt** column by clicking its column header.
    b) Right-click the selected column and click the **Change Type > Currency** menu command.
46. Expand the **Invoices** column to add the **InvoiceDate** column and the **CustomerId** column to the **Sales** query.
    a) Click the Expand button inside the column header of the **Invoices** column to display the **Columns to Expand** dialog.
    b) In the **Columns to Expand** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkboxes for the **InvoiceDate** column and the **CustomerId** column. Also make sure to uncheck the checkbox with the caption **Use original column name as prefix**. Once the **Columns to Expand** dialog looks like the one shown in the following screenshot, click the **OK** button to close the dialog and to modify the underlying query.



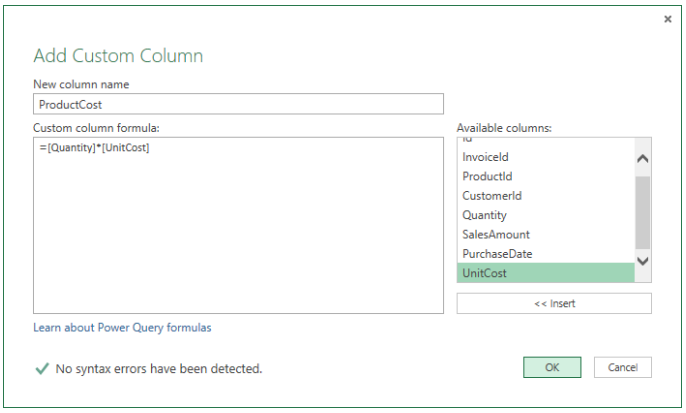
    c) You should see that the **InvoiceDate** column and the **CustomerId** column have now been added to the **Sales** query results.
47. Use the mouse to drag and drop the **CustomerId** column to move it to the right of the **InvoiceId** column.
48. Use the mouse to drag and drop the **InvoiceDate** column to move it to the right of the **ProductId** column.
49. Change the column type of the **InvoiceDate** to the **Date** type by right-clicking its column header and selecting the **Change Type > Date** command. When you are done, the data in the column should show a date value without a time.
50. Change the name of the **InvoiceDate** column to **PurchaseDate**.
51. Expand the **Products** column to add the **UnitCost** column to the **Sales** query.
    a) Click the Expand button inside the column header of the **Products** column to display the **Columns to Expand** dialog.
    b) In the **Columns to Expand** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkbox for the **UnitCost** column. Also make sure to uncheck the checkbox with the caption **Use original column name as prefix**. Once the **Columns to Expand** dialog looks like the one shown in the following screenshot, click the **OK** button to close the dialog and to modify the underlying query.
    c) You should see that the **UnitCost** column has now been added to the **Sales** query results.

	Id	InvoiceId	ProductId	CustomerId	Quantity	SalesAmount	PurchaseDate	UnitCost
1	1	1	22	1	4	119.8	1/28/2012	21.5
2	2	1	22	1	1	29.95	1/28/2012	21.5
3	3	2	22	2	2	59.9	1/28/2012	21.5
4	22	14	22	14	1	29.95	1/29/2012	21.5
5	10	6	1	6	4	59.8	1/28/2012	6.85
6	18	12	1	12	3	44.85	1/29/2012	6.85
7	27	17	1	17	4	59.8	1/29/2012	6.85
8	11	7	3	7	4	59.8	1/29/2012	6.1

52. Add a new custom column named **ProductCost** to calculate the product of the **Quantity** field multiplied by the **UnitCost** field.
    a) Begin by activating the **Add Column** tab in the ribbon.
    b) Click the **Add Custom Column** button in the ribbon to display the **Add Custom Column** dialog.
    c) In the **Add Custom Column** dialog, add a value of **ProductCost** in the **New column name** textbox.
    d) In the **Custom column formula** textbox, enter the following formula.

```
[Quantity] * [UnitCost]
```

    e) When the **Add Custom Column** dialog appears as the following screenshot, click the **OK** button to add the new column.



    f) You should be able to verify that the new **ProductCost** column has a value calculated by multiplying the value of the **Quantity** column together with the value of the **UnitCost** column.

Once the **UnitCost** column has been used to calculate the value for the **ProductCost** value, this column is no longer needed and can be removed from the results of the **Sales** query.

53. Remove the **UnitCost** column by selecting its column header and pressing the DELETE key on the keyboard.
54. Modify the column type of the **SalesAmount** column and the **ProductCost** column to the **Currency** type.
    a) Select the **SalesAmount** column by clicking its column header.
    b) Hold down the **SHIFT** key and click the **ProductCost** column so that both columns are selected.
    c) Right-click either one of the selected columns and click the **Change Type > Currency** menu command.
55. You are now done working with the **Sales** query and its output should match the following screenshot.

	Id	InvoiceId	ProductId	CustomerId	Quantity	SalesAmount	PurchaseDate	ProductCost
1	1	1	22	1	4	119.8	1/28/2012	86
2	2	1	22	1	1	29.95	1/28/2012	21.5
3	3	2	22	2	2	59.9	1/28/2012	43
4	22	14	22	14	1	29.95	1/29/2012	21.5
5	10	6	1	6	4	59.8	1/28/2012	27.4
6	18	12	1	12	3	44.85	1/29/2012	20.55

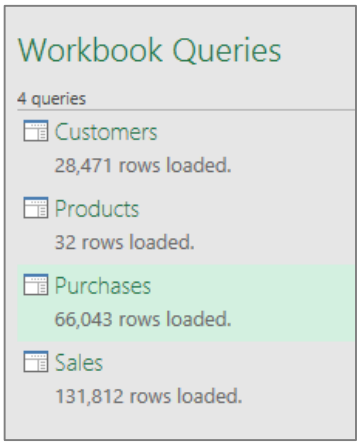
56. Click the **Close & Load** button on the **Home** tab of the ribbon to execute the updated **Sales** query. Excel will display the **Apply Query Changes** dialog while importing the data and transforming it to load it into the data model.
57. After the query changes have been applied, you should be able to see the results of your changes in the **Sales** table that has been loaded into the project's data model.
58. Save your work by clicking the Save button in the upper left corner of the Excel window.
59. Right-click on the **Invoices** query in the **Workbook Queries** pane and select the **Edit** command to open the query in the **Query Editor** window.

In an earlier step you changed the name of the **InvoiceDetails** query to **Sales** to make the data model easier to understand. In this exercise you will change the name of the **Invoices** query to **Purchases** for the same reason.

60. Update the name of the **Invoices** query to **Purchases** by replacing the text in the **Name** textbox in the **Query Settings** pane.
61. Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
62. In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkboxes for **InvoiceId** and **InvoiceType** as shown in the following screenshot. Once you have these columns selected, click the **OK** button to close the **Choose Columns** dialog and to modify the underlying query.
63. You should be able to see that the Query Editor window now only shows the columns that you selected.
64. Modify the query so that the **InvoiceType** column returns values that are more human readable.
    a) Make sure the **Transform** tab is the active tab in the ribbon.
    b) Select the **InvoiceType** column by clicking its column header.
    c) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.
    d) In the **Replace Value** dialog, enter a value of **InPerson** in the **Value to Find** textbox and enter a value of **Store Purchase** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.
    e) Make sure the **InvoiceType** column is still selected.
    f) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.
    g) In the **Replace Value** dialog, enter a value of **MailOrder** in the **Value to Find** textbox and enter a value of **Mail Order Purchase** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.
    h) Make sure the **InvoiceType** column is still selected.
    i) Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.
    j) In the **Replace Value** dialog, enter a value of **Online** in the **Value to Find** textbox and enter a value of **Online Purchase** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.
    k) If you scroll down and look at all the rows within the **Purchases** table, you should be able to see that each row has a **InvoiceType** column value of either **Store Purchase**, **Mail Order Purchase** or **Online Purchase**.
65. Change the name of the **InvoiceType** column to **Purchase Type**.

	InvoiceId	Purchase Type
1	1	Store Purchase
2	2	Store Purchase
3	3	Store Purchase
4	4	Store Purchase
5	5	Store Purchase
6	6	Store Purchase

66. You are now done working with the **Purchase** query.



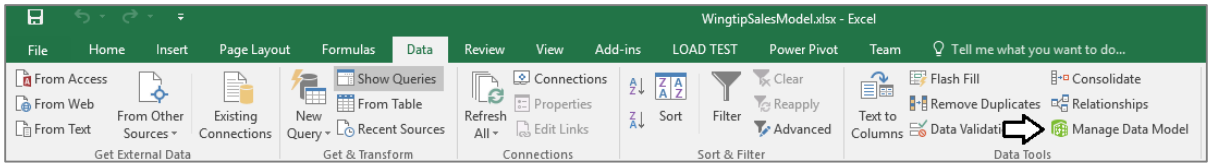
67. Save your work by clicking the Save button in the upper left corner of the Excel window.

You have now finished your work with Power Query and ready to begin working with Power Pivot.

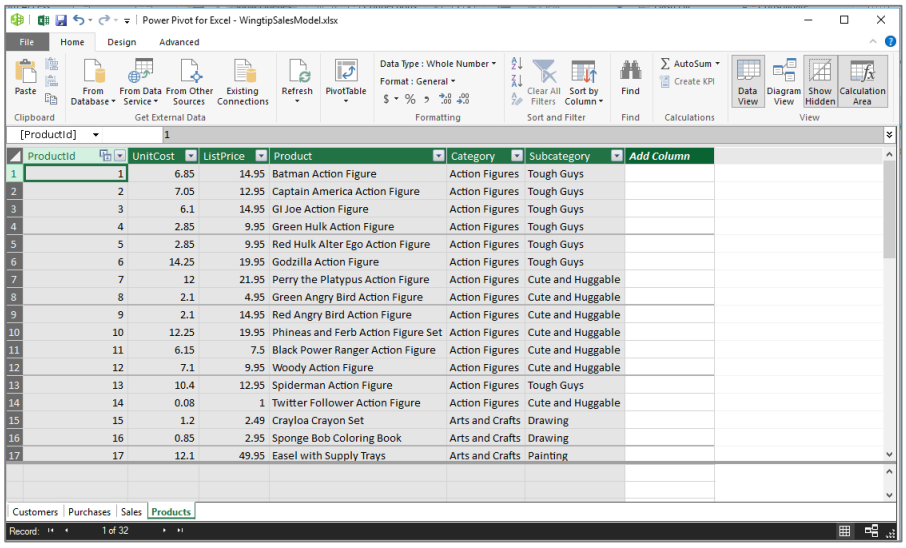
Exercise 2: Using the Power Pivot Features of Excel 2016

In this exercise you will model the data you just imported using Power Pivot.

1. Open the Power Pivot windows to view and edit the data model.
   a) Activate the **Data** tab in the ribbon.
   b) Click on the **Manage Data Model** button at the far right-hand side of the ribbon to display the **Power Pivot for Excel** window.

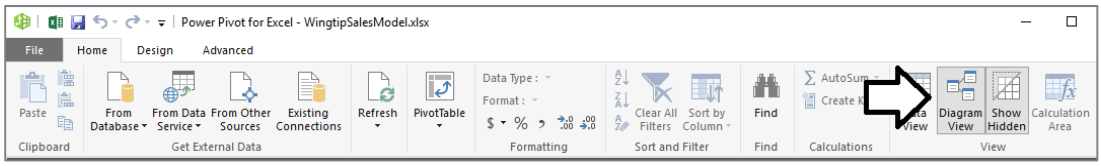


   c) Examine the tables and verify that they have the Power Query transformations you applied in the previous exercise.

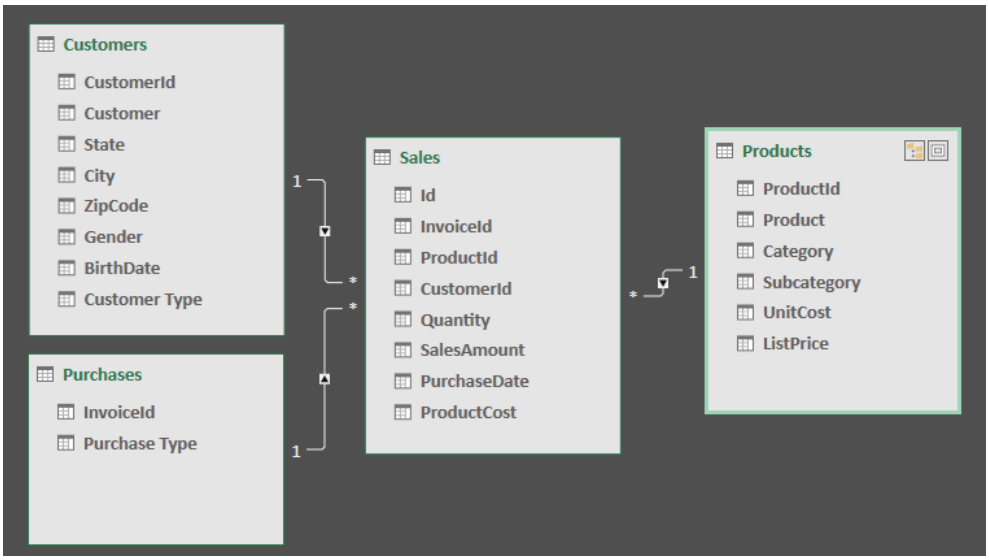


ProductId	UnitCost	ListPrice	Product	Category	Subcategory	Add Column
1	6.85	14.95	Batman Action Figure	Action Figures	Tough Guys	
2	7.05	12.95	Captain America Action Figure	Action Figures	Tough Guys	
3	6.1	14.95	GI Joe Action Figure	Action Figures	Tough Guys	
4	2.85	9.95	Green Hulk Action Figure	Action Figures	Tough Guys	
5	2.85	9.95	Red Hulk Alter Ego Action Figure	Action Figures	Tough Guys	
6	14.25	19.95	Godzilla Action Figure	Action Figures	Tough Guys	
7	12	21.95	Perry the Platypus Action Figure	Action Figures	Cute and Huggable	
8	2.1	4.95	Green Angry Bird Action Figure	Action Figures	Cute and Huggable	
9	2.1	14.95	Red Angry Bird Action Figure	Action Figures	Cute and Huggable	
10	12.25	19.95	Phineas and Ferb Action Figure Set	Action Figures	Cute and Huggable	
11	6.15	7.5	Black Power Ranger Action Figure	Action Figures	Cute and Huggable	
12	7.1	9.95	Woody Action Figure	Action Figures	Cute and Huggable	
13	10.4	12.95	Spiderman Action Figure	Action Figures	Tough Guys	
14	0.08	1	Twitter Follower Action Figure	Action Figures	Cute and Huggable	
15	1.2	2.49	Crayloa Crayon Set	Arts and Crafts	Drawing	
16	0.85	2.95	Sponge Bob Coloring Book	Arts and Crafts	Drawing	
17	12.1	49.95	Easel with Supply Trays	Arts and Crafts	Painting	

2. Modify the relations in the data model.
   a) Navigate to diagram view to see the relationships between the tables.



   b) Reorganize the tables so the view looks cleaner.



3. Modify the formatting of the **BirthDate** column in the **Customers** table.
   a) In the Power BI Desktop windows, navigate back to data view.
   b) In the **Fields** list on the right, select the **Customers** table to display its rows and columns.
   c) Select the **BirthDate** column by clicking on its column header.
   d) Modify the formatting of the **BirthDate** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Date Time > 3/14/2001 (M/d/yyyy)**.
   e) The **BirthDate** column should now reflect the change in formatting.
4. Modify the formatting of columns in the **Products** table.
   a) In the **Fields** list on the right, select the **Products** table to display its rows and columns.
   b) Select the **UnitCost** column by clicking on its column header.
   c) Use the **Format** menu button in the ribbon to update the format setting to **Currency > English (United States)**.
   d) Changing the format setting of the **ListPrice** column to **Currency > English (United States)** so it matches **UnitCost**.
5. Modify the formatting of columns in the **Sales** table.
   a) In the **Fields** list on the right, select the **Sales** table to display its rows and columns.
   b) Select the **Quantity** column by clicking on its column header.
   c) Modify the **Quantity** column by clicking to select the comma button on the ribbon to add a comma separator.
   d) Select the **SalesAmount** column by clicking on its column header.
   e) Modify the formatting of the **SalesAmount** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Currency > English (United States)**.
   f) Select the **PurchaseDate** column by clicking on its column header.

g) Modify the formatting of the **Purchase Date** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Date Time > 3/14/2001 (M/d/yyyy)**.

h) Select the **ProductCost** column by clicking on its column header.

i) Modify the formatting of the **ProductCost** column by dropping down the **Format** button menu in the ribbon and selecting a format setting of **Currency > English (United States)**.

6. Add a calculated column to the **Sales** table named **SalesProfit** to determine profit by calculating the difference between **SalesAmount** and **ProductCost**.

a) Navigate to data view.

b) Select the **Sales** table in the **Fields** list.

c) Create a new calculated column by clicking the **New Column** button in the ribbon.

d) Enter to following DAX expression into the formula bar to create the calculated column named **SalesProfit**.

```
SalesProfit = Sales[SalesAmount]-Sales[ProductCost]
```

e) Press the **ENTER** key to add the calculated column to the table. You should be able to see a **SalesProfit** value for each row in the **Sales** table.

f) Configure the column's formatting by using the **Format** menu on the ribbon to select **Currency > English (United States)**.

7. Add a calculated column to the **Sales** table named **PurchaseYear** to indicate the calendar year of each purchase.

a) Navigate to data view.

b) Select the **Sales** table in the **Fields** list.

c) Create a new calculated column by clicking the **New Column** button in the ribbon.

d) Enter to following DAX expression into the formula bar to create the calculated column named **PurchaseYear**.

```
PurchaseYear = YEAR(Sales[PurchaseDate])
```

e) Press the **ENTER** key to add the calculated column to the table. You should be able to see a calendar year value (e.g. 2012) for each row in the **Sales** table

f) Once the **PurchaseYear** column has been created, use the **Data Type** dropdown menu on the ribbon to change the column type from **Whole Number** to **Text**.

8. At this point, you are on your own to continue extending the data model for the workbook file to make it identical to the data model you created earlier with Power BI Desktop.