

Designing Queries to Extract and Transform Data

Lab Time: 40 minutes

Lab Folder: C:\Student\Modules\02_Queries\Lab\

Lab Overview: In this lab you will begin by creating a new Power BI Desktop project and saving it as a PBIX file. Next, you will learn to work with the Power Query features of Power BI Desktop to extract data from a SQL Azure database and to transform the data as it is loaded into the data model. This is the first lab in a sequence of labs that continue with the same PBIX file. In other words, the labs that follow will build upon the work you do in this lab.

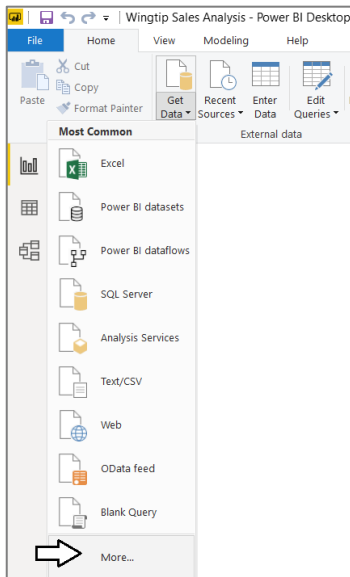
Exercise 1: Importing Data from a SQL Azure Database

In this exercise you will create and save a new Power BI Desktop project. After that, you will connect to a SQL Azure database and import data into Power BI Desktop using its Power Query features.

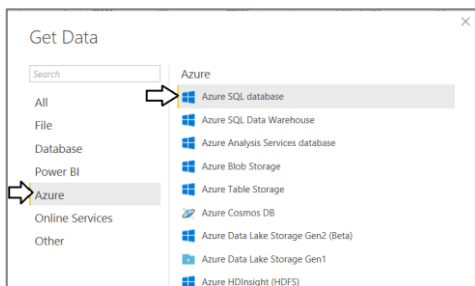
1. Launch Power BI Desktop to start a new project.
2. Save the new project as **Wingtip Sales Analysis.pbix** using the following path.

C:\Student\Projects\Wingtip Sales Analysis.pbix

3. Drop down the **Get Data** menu button on the ribbon and click **More....**

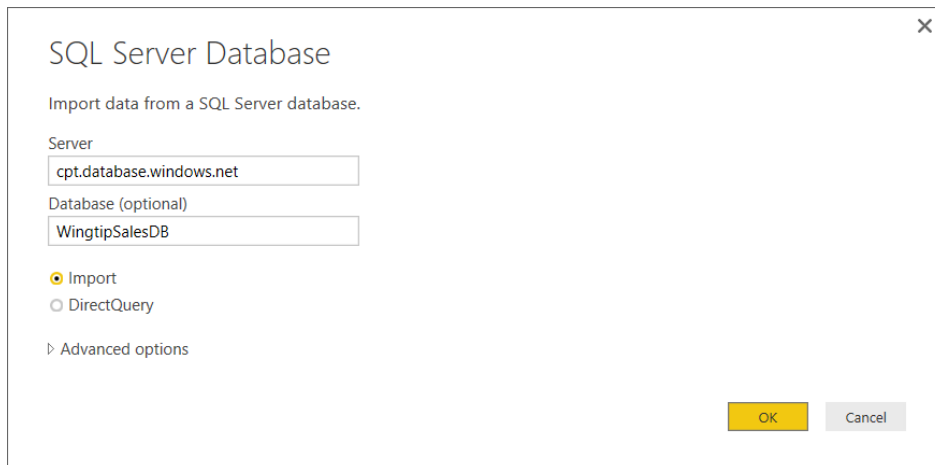


4. On the **Get Data** dialog, select **Azure** in the list on the left. Next, select **Microsoft SQL Azure Database** on the right and then click the **Connect** button.



5. When you are prompted with the **SQL Server Database** dialog, complete the following tasks.
 - a) Enter a **Server** value of **cpt.database.windows.net**
 - b) Enter a **Database** value of **WingtipSalesDB**

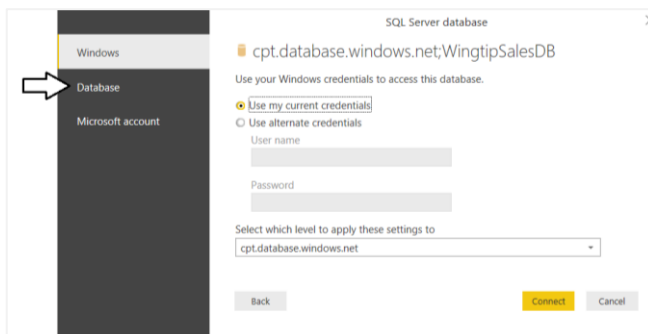
- c) Leave the option button with the default setting of **Import** and not **DirectQuery**.
- d) Click the **OK** button to continue.



The dialog box is titled "SQL Server Database" and contains the following elements:

- Text: "Import data from a SQL Server database."
- Server: A text box containing "cpt.database.windows.net".
- Database (optional): A text box containing "WingtipSalesDB".
- Radio buttons: "Import" (selected) and "DirectQuery".
- Link: "Advanced options" with a right-pointing triangle.
- Buttons: "OK" (yellow) and "Cancel" (grey).

6. 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.

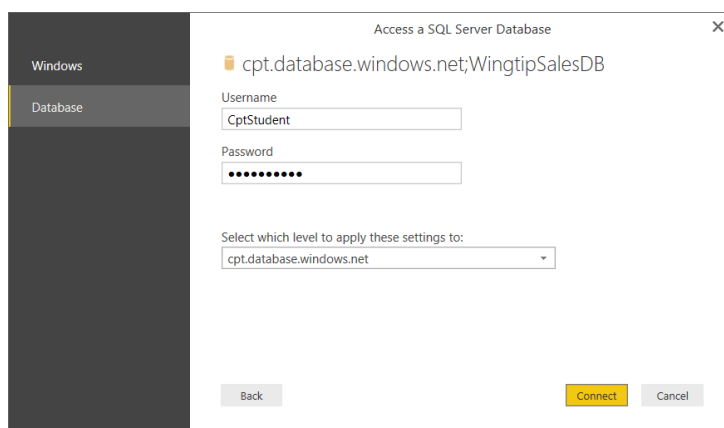


The dialog box is titled "Access a SQL Server Database" and contains the following elements:

- Left sidebar: "Windows", "Database" (selected, with a white arrow pointing to it), and "Microsoft account".
- Right pane: Shows "cpt.database.windows.net;WingtipSalesDB" and "Use your Windows credentials to access this database." with radio buttons for "Use my current credentials" (selected) and "Use alternate credentials". Below are fields for "User name" and "Password".
- Dropdown: "Select which level to apply these settings to" with "cpt.database.windows.net" selected.
- Buttons: "Back", "Connect" (yellow), and "Cancel" (grey).

If you don't switch the selection in the dialog above from **Windows** to **Database**, things will not work correctly.

- 7. 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**
- 8. Once you have entered the credentials the **Access a SQL Server Database** dialog, click the **Connect** button to continue.

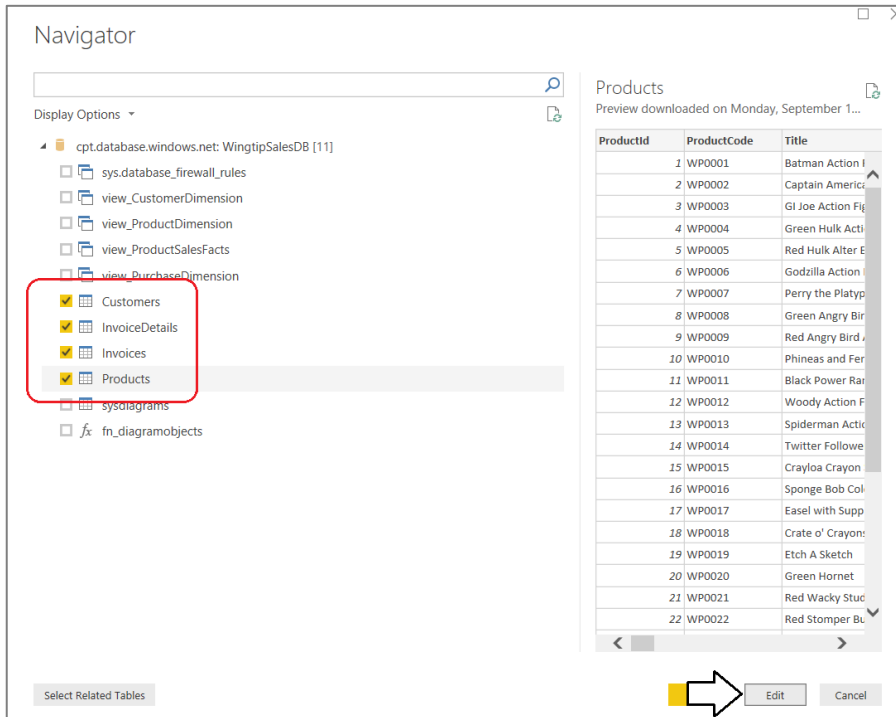


The dialog box is titled "Access a SQL Server Database" and contains the following elements:

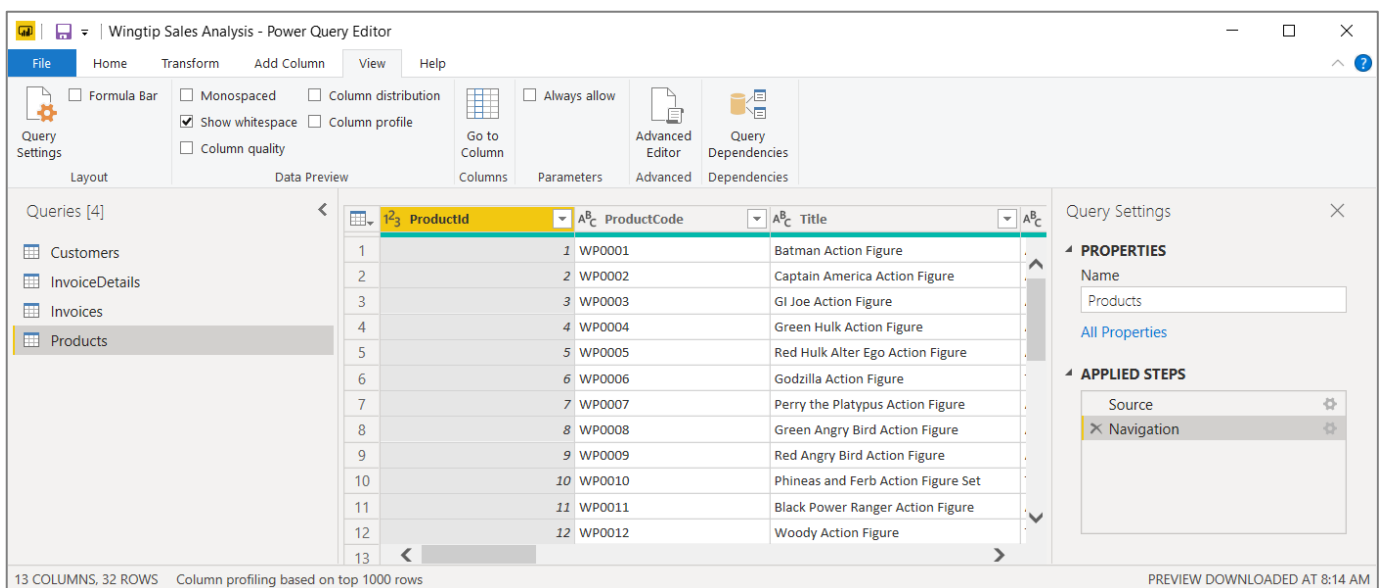
- Left sidebar: "Windows", "Database" (selected), and "Microsoft account".
- Right pane: Shows "cpt.database.windows.net;WingtipSalesDB" and "Use your Windows credentials to access this database." with radio buttons for "Use my current credentials" (selected) and "Use alternate credentials". Below are fields for "Username" (containing "CptStudent") and "Password" (containing "pass@word1").
- Dropdown: "Select which level to apply these settings to" with "cpt.database.windows.net" selected.
- Buttons: "Back", "Connect" (yellow), and "Cancel" (grey).

At this point, Power BI Desktop 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 PBIX project.

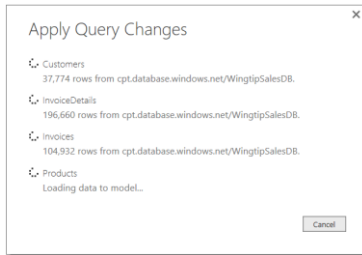
9. In the **Navigator** dialog, select the **Customers** table, the **InvoiceDetails** table, the **Invoices** table and the **Products** table as shown in the following screenshot. Once you have selected these four tables, click the **Edit** button to create a query for each of these tables and to open the **Query Editor** dialog.



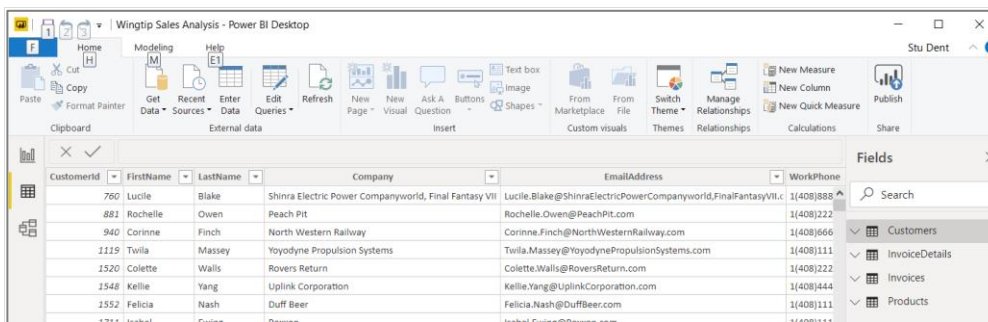
10. When you inspect the **Power Query Editor** window, you should be able to observe that Power BI Desktop has created a new query for each of the 4 tables that selected in the **Navigator** dialog. You will not begin to edit these queries until the next exercise. However, now you should take a moment to inspect each query by clicking on the query name in the **Queries** list on the left. You should be able to observe that each query is initially created to return all of the columns that are defined in the underlying tables.



11. Click the **Close and Apply** button in the **Query Editor** window to begin the process of extracting the data from the SQL Azure database. The **Apply Query Changes** dialog will be displayed with spinning icons as Power BI Desktop imports the data.



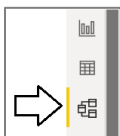
12. Once the data from all four queries has been imported into the current project, examine the **Fields** list on the right-hand side of the Power BI Desktop window. You should be able to see that four tables have been created from the four queries.



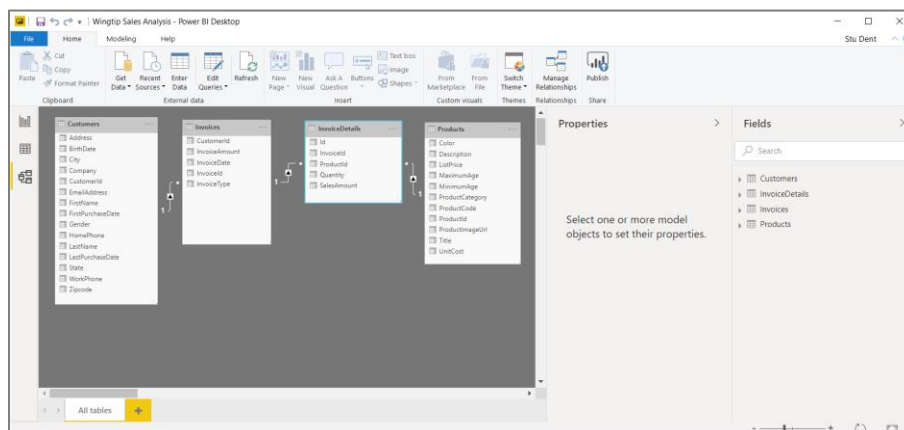
Customerid	FirstName	LastName	Company	EmailAddress	WorkPhone
760	Lucile	Blake	Shinra Electric Power Companyworld, Final Fantasy VII	Lucile.Blake@ShinraElectricPowerCompanyworld.FinalFantasyVII.c	1(408)888
881	Rochelle	Owen	Peach Pit	Rochelle.Owen@PeachPit.com	1(408)222
940	Corinne	Finch	North Western Railway	Corinne.Finch@NorthWesternRailway.com	1(408)666
1119	Twila	Massey	Yoyodyne Propulsion Systems	Twila.Massey@YoyodynePropulsionSystems.com	1(408)111
1520	Colette	Walls	Rovers Return	Colette.Walls@RoversReturn.com	1(408)222
1548	Kellie	Yang	Uplink Corporation	Kellie.Yang@UplinkCorporation.com	1(408)444
1552	Felicia	Nash	Duff Beer	Felicia.Nash@DuffBeer.com	1(408)111

13. Examine the data model for your new project in Model view.

- a) Click on the Model view button in the left navigation to move into Model view.



- b) You should be able to see each of the four tables.
- c) You should also be able to see that Power BI Desktop has automatically created relationships between the tables as they were imported into the data model.
- d) Use your mouse to move and resize the tables in relationship view so you can see all the fields in each table.

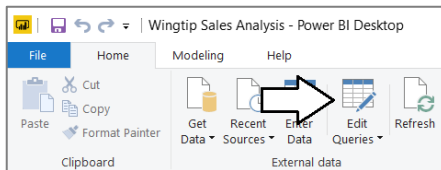


14. Save your work in the current project by clicking the **Save** button in the upper left corner of the Power BI Desktop window.

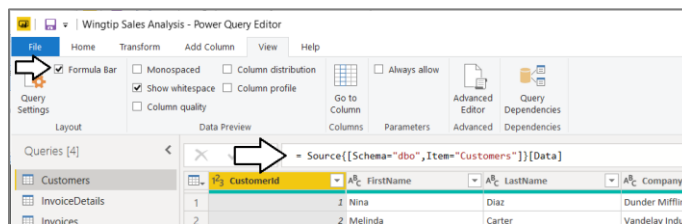
Exercise 2: Design Queries to Transform and Reshape Customer Sales Data

In the following exercise, you will use the **Query Editor** window to modify the **Customers** query to perform transforms on customer data as it is being loaded into the data model.

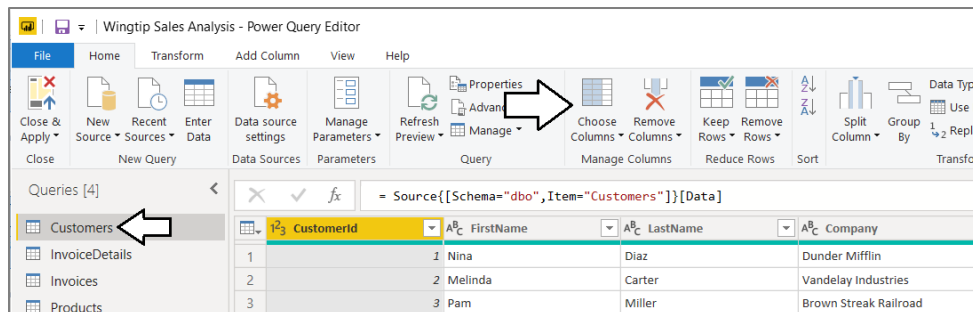
1. Make sure you have the **Wingtip Sales Analysis.pbix** project open that you started in the previous exercise.
2. Click on the **Edit Queries** button in the ribbon to display the Query Editor window.



3. Make sure the query formula bar is visible in the Query Editor window.
 - a) In the Query Editor window, navigate to the **View** tab and check the **Formula Bar** checkbox.
 - b) You should now see the query formula bar with the M code for the current step displayed just above the query results.

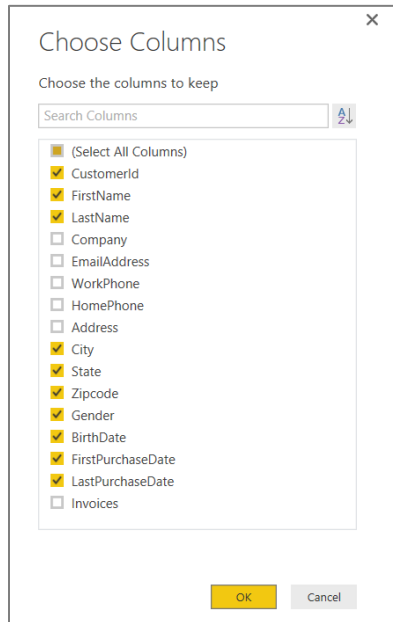


4. Select the desired set of columns from the **Customers** table.
 - a) Make sure the **Customers** query is selected in the **Queries** list on the left-hand side of the Query Editor window.
 - b) Navigate to the **Home** tab and click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.

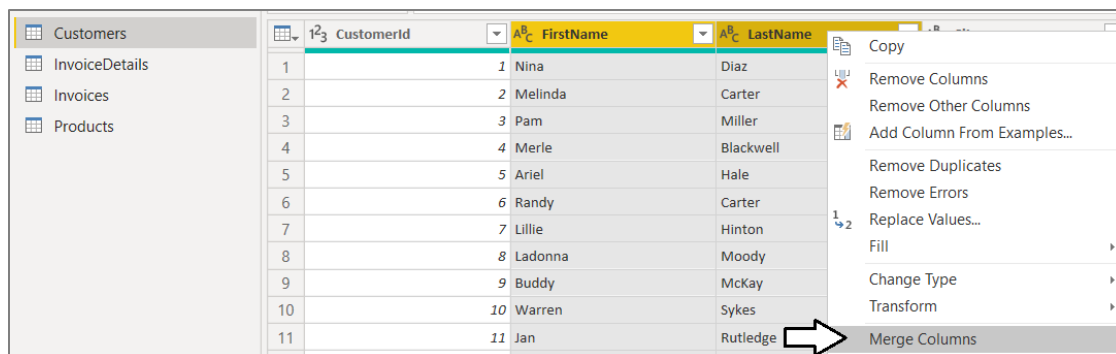


- c) In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all column. Next, select the checkboxes for the following columns.
 - i) **CustomerId**
 - ii) **FirstName**
 - iii) **LastName**
 - iv) **City**
 - v) **State**
 - vi) **Zipcode**
 - vii) **Gender**
 - viii) **BirthDate**
 - ix) **FirstPurchaseDate**
 - x) **LastPurchaseDate**

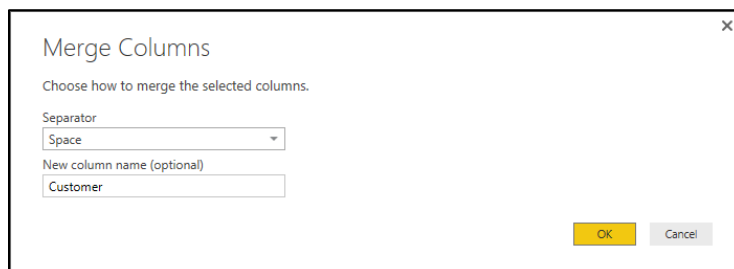
- d) Once you have the columns selected as shown in the following screenshot, click the **OK** button to close the **Choose Columns** dialog and to modify the underlying query.



- e) You should be able to see that the Query Editor window now only shows the columns that you selected.
5. In this step you will merge the **FirstName** column and the **LastName** column together into a single column named **Customer**.
- Select the **FirstName** column by clicking on its column header.
 - Next, hold down the **SHIFT** key and select the **LastName** column by clicking on its column header.
 - Right-click on the selected columns and click the **Merge Columns** menu command.



- d) 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.



- e) 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**.

	1 2 3 CustomerId	A ^B C Customer	A ^B C City	A ^B C State
1	1	Nina Diaz	Eureka	CA
2	2	Melinda Carter	Napa	CA
3	3	Pam Miller	Napa	CA
4	4	Merle Blackwell	Sacramento	CA
5	5	Ariel Hale	Sacramento	CA

6. Modify the query so that the **Gender** column returns values of **Male** and **Female** instead of **M** and **F**.
- Locate the **Gender** column in the **Customers** table.
 - Right-click the header for the **Gender** column and select the **Replace Values** command to display the **Replace Values** dialog.

Customers	A ^B C City	A ^B C State	A ^B C Zipcode	A ^B C Gender	BirthDate
7	Eureka	CA	95501	F	
8	Napa	CA	94559	F	
9	Bend	OR	97701	M	
10	Sacramento	CA	95818	M	
11	Portland	OR	97216	F	
12	Eugene	OR	97402	M	
13	Portland	OR	97220	M	
14	Sacramento	CA	95823	F	
15	Redmond	WA	98052	M	
16	Beaverton	OR	97005	M	
17	Vancouver	WA	98684	M	

- c) 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.

Replace Values

×

Replace one value with another in the selected columns.

Value To Find

F

Replace With

Female

Advanced options

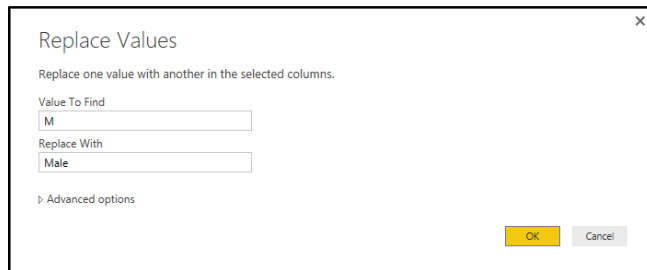
OK

Cancel

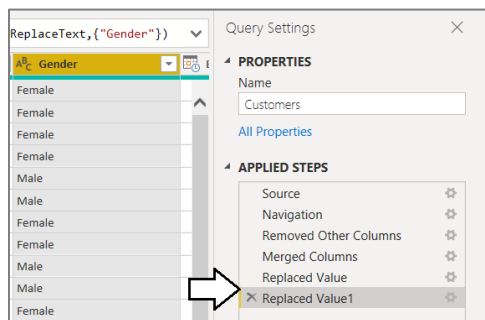
- d) You should be able to see that all values of **F** in the **Gender** column have been replaced with a value of **Female**.

A ^B C Gender
Female
Female
Female
Female
M
M
Female

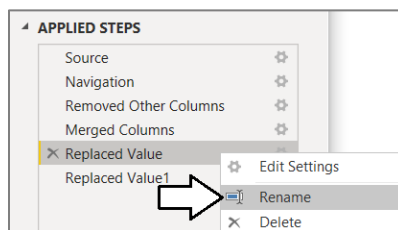
- e) Right-click the header for the **Gender** column and select the **Replace Values** command a second time.
- f) In the **Replace Value** dialog, enter a value of **M** in the **Value to Find** textbox and enter a value of **Male** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.



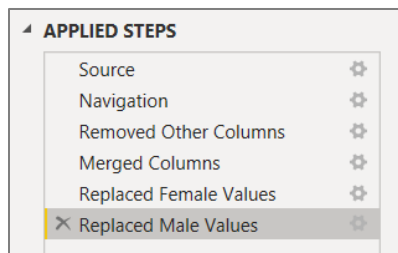
- g) You should be able to confirm that all values in the **Gender** column have been replaced with a value of either **Male** or **Female**.
- h) Inspect the **APPLIED STEPS** list in the **Query Settings** pane. You should be able to see that there are two steps at the end that have been given the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



- i) 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 **Replace Female Values**.

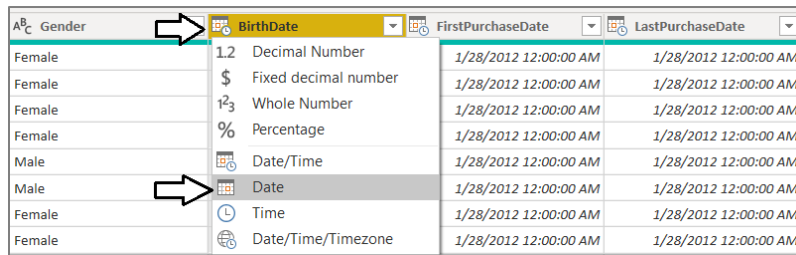


- j) Using the same technique, rename the **Replaced Value 1** step to **Replaced Male Values**.



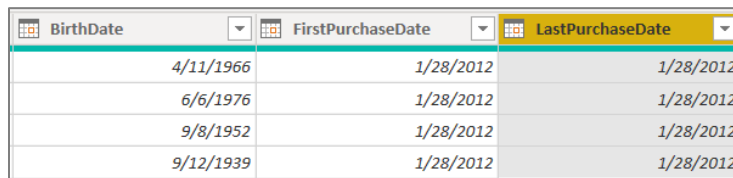
You have now learned how to rename a query step. Note that this lab exercise will not continue to ask you to change the name of every step due to time constraints. However, when you are creating queries in larger, real-world projects that involve multiple team members, it's a good practice to rename query steps to make your query logic easier for others to read, understand and extend.

7. Change the column type of **BirthDate**, **FirstPurchaseDate** and **LastPurchaseDate** from **Date/Time** to **Date**.
- a) Use the column type drop down on the left-hand side of the **BirthDate** column to configure the column using the **Date** type.



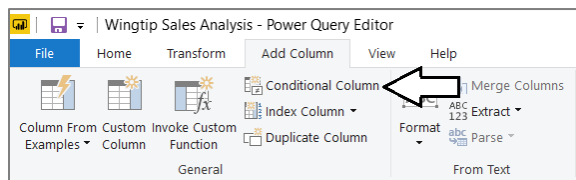
Gender	BirthDate	FirstPurchaseDate	LastPurchaseDate
Female	1.2	1/28/2012 12:00:00 AM	1/28/2012 12:00:00 AM
Female	\$	1/28/2012 12:00:00 AM	1/28/2012 12:00:00 AM
Female	123	1/28/2012 12:00:00 AM	1/28/2012 12:00:00 AM
Female	%	1/28/2012 12:00:00 AM	1/28/2012 12:00:00 AM
Male	Date/Time	1/28/2012 12:00:00 AM	1/28/2012 12:00:00 AM
Male	Date	1/28/2012 12:00:00 AM	1/28/2012 12:00:00 AM
Female	Time	1/28/2012 12:00:00 AM	1/28/2012 12:00:00 AM
Female	Date/Time/Timezone	1/28/2012 12:00:00 AM	1/28/2012 12:00:00 AM

- b) Use the column type drop down of the **FirstPurchaseDate** column to configure the column using the **Date** type.
- c) Use the column type drop down of the **LastPurchaseDate** column to configure the column using the **Date** type.
- d) You should see that the three columns now show values with a date but without a time.



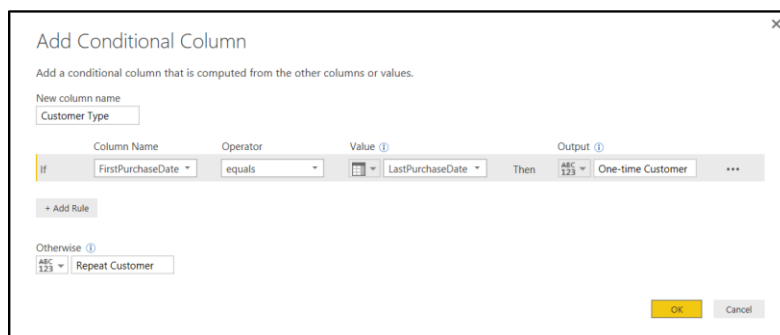
BirthDate	FirstPurchaseDate	LastPurchaseDate
4/11/1966	1/28/2012	1/28/2012
6/6/1976	1/28/2012	1/28/2012
9/8/1952	1/28/2012	1/28/2012
9/12/1939	1/28/2012	1/28/2012

8. Add a new conditional column named **Customer Type** to indicates whether the customer is a repeat customer or not.
- a) Make sure the **Customers** query is still selected as the active query.
- b) Activate the **Add Column** tab in the ribbon.
- c) Click the **Conditional Column** button in the ribbon to display the **Add Custom Column** dialog.



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 **Add Conditional Column** dialog, enter a **New column name** value of **Customer Type**.
- e) Configure a rule to return a string value of "One-time Customer" if **FirstPurchaseDate** equals **LastPurchaseDate**.
- f) For the **Otherwise** evaluation, return a string value of "Repeat Customer".
- g) When the **Add Conditional Column** dialog matches the screenshot below, click the **OK** button to add the new column.



Add Conditional Column

Add a conditional column that is computed from the other columns or values.

New column name: Customer Type

Column Name	Operator	Value	Output
If	FirstPurchaseDate	equals	LastPurchaseDate
Then			One-time Customer
Otherwise			Repeat Customer

OK Cancel

- h) 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**.

You might have to scroll down several pages of records in the **Customers** table before you begin to see repeat customers.

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

9. Now, that you have used the **FirstPurchaseDate** column and the **LastPurchaseDate** column to calculate the value of the **Customer Type** column, you can delete them because they are no longer needed.
- Select the **FirstPurchaseDate** column by clicking its column header.
 - Hold down the **SHIFT** key and click the column header for **LastPurchaseDate** so that both columns are selected.
 - Right click the one of the selected columns and click the **Remove Columns**.

BirthDate	FirstPurchaseDate	LastPurchaseDate	Customer Type
4/11/1966	1/28/2012	1/28/2012	One-time Customer
6/6/1976	1/28/2012	1/28/2012	One-time Customer
9/8/1952	1/28/2012	1/28/2012	One-time Customer
9/12/1939	1/28/2012	1/28/2012	One-time Customer

- d) You should be able to confirm that the **FirstPurchaseDate** column and the **LastPurchaseDate** columns have been removed from the query results. However, the **Customer Type** column is still there.

Gender	BirthDate	Customer Type
Female	4/11/1966	One-time Customer
Female	6/6/1976	One-time Customer
Female	9/8/1952	One-time Customer
Female	9/12/1939	One-time Customer

- e) You might notice that Type menu for the Customer Type column is not set to a specific type. When you see the type as ABC above and 123 below, that the column is being assigned the generic **Any** type.

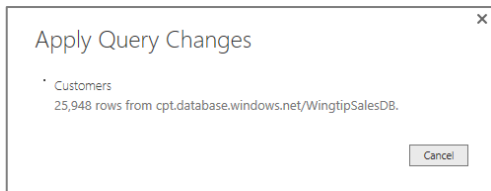
BirthDate	Customer Type
4/11/1966	1.2

- f) Drop down the Type menu for the **Customer Type** column and set its value to **Text**.

Gender	BirthDate	Customer Type
Female	4/11/1966	1.2
Female	6/6/1976	\$
Female	9/8/1952	123
Female	9/12/1939	%
Male	9/15/1965	Date/Time
Male	7/14/1953	Date
Female	2/3/1992	Time
Female	4/5/1949	Date/Time/Timezone
Male	5/10/1989	Duration
Male	6/11/1981	Text
Female	11/26/1981	True/False
Male	3/26/1972	

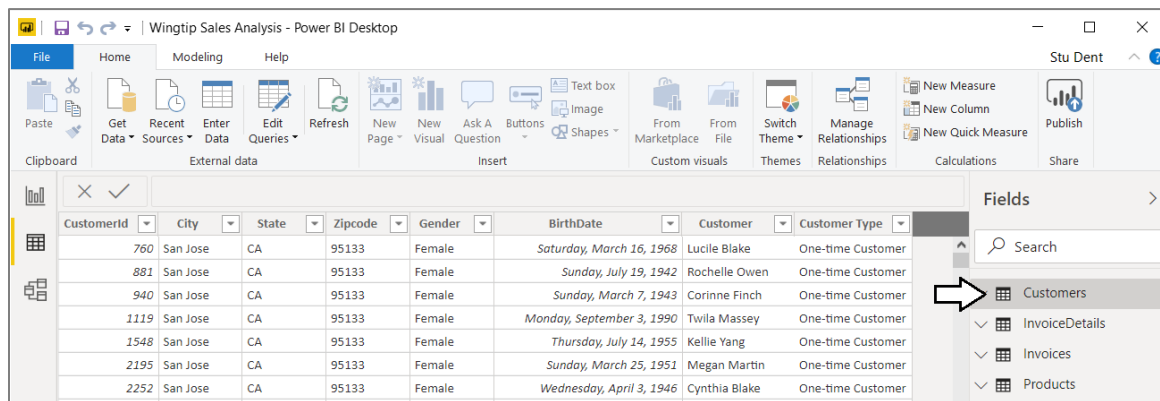
10. Execute the Customers query to see the results in the main Power BI Desktop window.

- a) Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Customers** query.



Power BI Desktop will display the **Apply Query Changes** dialog while importing the data from the SQL Azure database and transforming it as it loads the customer data into the data model.

- b) After the **Customer** query changes have been applied, you should be able to see the results in **Data** view of the main Power BI Desktop window when the **Customers** table is selected in the **Fields** list on the right.



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

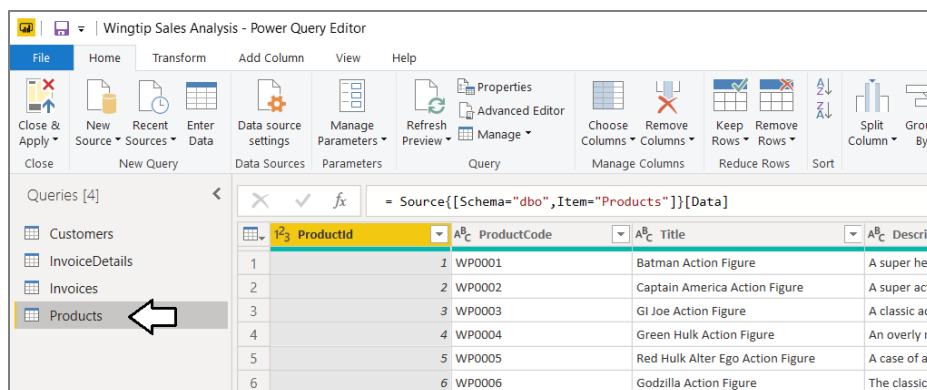
11. Save your work to **Wingtip Sales Analysis.pbix** by clicking the **Save** button in the top left corner of the Power BI Desktop window.

Exercise 3: Using Power Query to Transform and Reshape Product Data

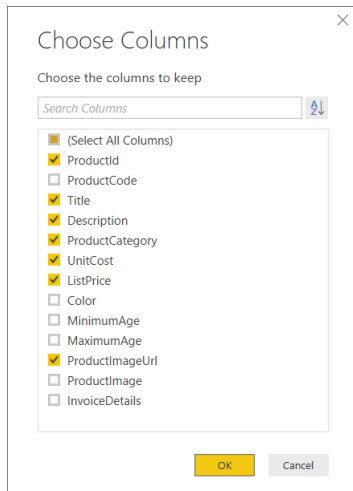
In the following exercise, you will use the **Power Query Editor** window to modify the **Products** query to perform transforms on product data as it is being loaded into the data model.

1. Open the **Products** query in edit mode.

- a) Click on the **Edit Queries** button in the **Home** tab of the ribbon to display the **Power Query Editor** window.
- b) Make sure the **Products** query is selected in the **Queries** list on the left-hand side of the **Query Editor** window.



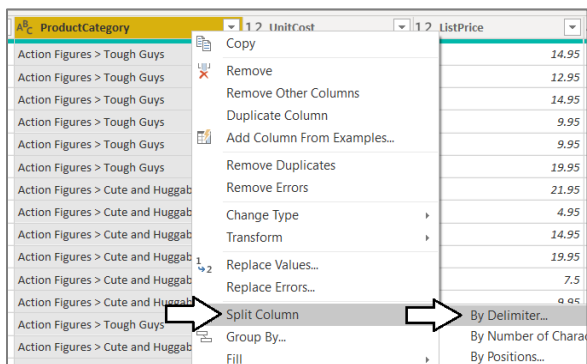
2. Remove the columns that are not required in the **Products** query results.
 - a) Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
 - b) In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns.
 - c) Select the checkboxes for **ProductId**, **Title**, **Description**, **ProductCategory**, **UnitCost**, **ListPrice** and **ProductImageUrl** as shown in the following screenshot.



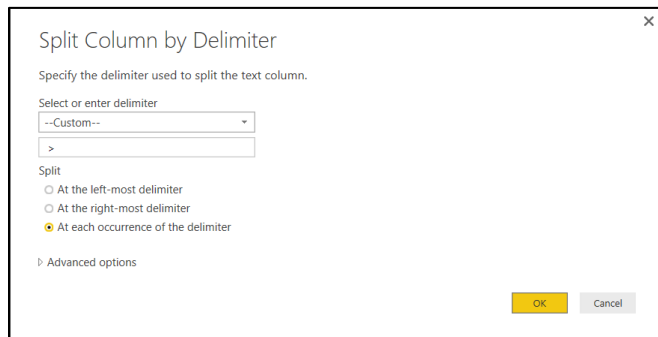
- d) Click the **OK** button to close the **Choose Columns** dialog and to modify the underlying query.
3. Rename the **Title** column to **Product**.
 - a) Right-click on the **Title** column and click **Rename**.
 - b) Update the column name to **Product**.

	1 2 3 ProductId	A ^B _C Product	A ^B _C Description
1	1	Batman Action Figure	A super hero who sometimes plays the role of a dar
2	2	Captain America Action Figure	A super action figure that protects freedom and the
3	3	GI Joe Action Figure	A classic action figure from the 1970s.
4	4	Green Hulk Action Figure	An overly muscular action figure that strips naked w
5	5	Red Hulk Alter Ego Action Figure	A case of anabolic steroids with a most unfortunate
6	6	Godzilla Action Figure	The classic and adorable action figure from those ol
7	7	Perry the Platypus Action Figure	A platypus who plays an overly intelligent detective
8	8	Green Angry Bird Action Figure	A funny looking green bird that really hates pigs.

4. Split the **ProductCategory** column up into two separate columns named **Category** and **Subcategory**.
 - a) Right-click the **ProductCategory** column and then click the **Split Column > By Delimiter** command.



- b) In the **Split Column By Column** dialog, drop down the **Select or enter delimiter** combo box and select **--Custom--**.
- c) In the **textbox** enter a three-character text value which includes a space follow by the **>** character followed by another space.
- d) When the **Split Column By Column** dialog matches the following screenshot, click the **OK** button.



- e) You should be able to confirm that Power BI Desktop has split the **ProductCategory** column into two separate columns named **ProductCategory.1** and **ProductCategory.2**.

ProductCategory.1	ProductCategory.2
Action Figures	Tough Guys
Action Figures	Tough Guys
Action Figures	Tough Guys
Action Figures	Tough Guys
Action Figures	Tough Guys
Action Figures	Tough Guys
Action Figures	Cute and Huggable
Action Figures	Cute and Huggable
Action Figures	Cute and Huggable

- f) Rename the **ProductCategory.1** column to **Category** and rename **ProductCategory.2** to **Subcategory**.

Category	Subcategory
Action Figures	Tough Guys
Action Figures	Tough Guys
Action Figures	Tough Guys
Action Figures	Tough Guys
Action Figures	Tough Guys
Action Figures	Tough Guys
Action Figures	Cute and Huggable
Action Figures	Cute and Huggable
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**.

5. Modify the column type of the **UnitCost** column and the **ListPrice** column to the **Fixed Decimal Number** type.
 - a) Use the dropdown column type menu to set the type of the **UnitCost** to **Fixed Decimal Number**.

UnitCost	ListPrice
1.2	14.95
\$	12.95
123	14.95
%	9.95

- b) Use the dropdown column type menu to set the type of the **ListPrice** to **Fixed Decimal Number**.

\$ UnitCost	\$ ListPrice
6.85	14.95
7.05	12.95
6.1	14.95
2.85	9.95
2.85	9.95

6. Apply your changes to the **Products** query.
- Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Products** query.
 - You should be able to see your changes in the **Products** table in **Data** view in the main Power BI Desktop window.

ProductId	Product	Description	UnitCost	ListPrice	http://classr
1	Batman Action Figure	A super hero who sometimes plays the role of a dark knight.	6.85	14.95	http://classr
2	Captain America Action Figure	A super action figure that protects freedom and the American way.	7.05	12.95	http://classr
3	GI Joe Action Figure	A classic action figure from the 1970s.	6.1	14.95	http://classr
4	Green Hulk Action Figure	An overly muscular action figure that strips naked when angry.	2.85	9.95	http://classr
5	Red Hulk Alter Ego Action Figure	A case of anabolic steroids with a most unfortunate outcome.	2.85	9.95	http://classr
6	Godzilla Action Figure	The classic and adorable action figure from those old Japanese mo-	14.25	19.95	http://classr
7	Perry the Platypus Action Figure	A platypus who plays an overly intelligent detective sleuth on TV.	12	21.95	http://classr
8	Green Angry Bird Action Figure	A funny looking green bird that really hates pigs.	2.1	4.95	http://classr
9	Red Angry Bird Action Figure	A funny looking red bird that also hates pigs.	2.1	14.95	http://classr
10	Phineas and Ferb Action Figure Set	The dynamic duo of the younger generation.	12.25	19.95	http://classr
11	Black Power Ranger Action Figure	A particularly violent action figure for violent children.	6.15	7.5	http://classr
12	Woody Action Figure	The lovable, soft-spoken cowboy from Toy Story.	7.1	9.95	http://classr
13	Spiderman Action Figure	The classic superhero who is quite the swinger.	10.4	12.95	http://classr
14	Twitter Follower Action Figure	An inexpensive action figure you can never have too many of.	0.08	1	http://classr
15	Crayola Crayon Set	A very fun set of crayons in every color.	1.2	2.49	http://classr
16	Sponge Bob Coloring Book	A drawing extravaganza based on America's most recognizable cel-	0.85	2.95	http://classr

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

7. Save your work by clicking the **Save** button in the upper left corner of the Power BI Desktop window.

Exercise 4: Using Power Query to Transform and Reshape Product Sales Data

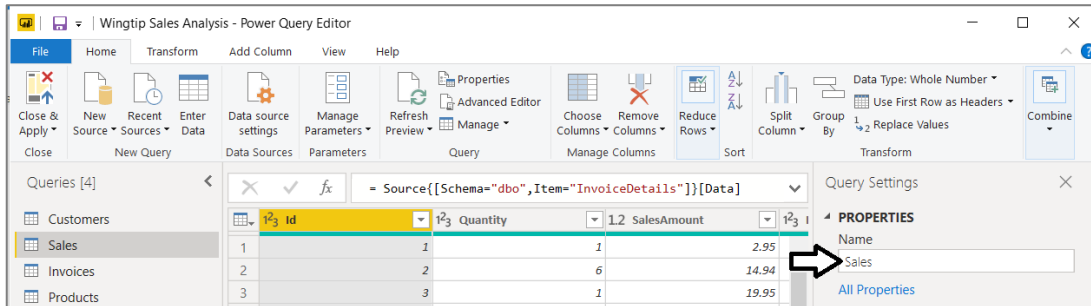
In the following exercise, you will use Power Query to modify the **InvoiceDetails** query to transform product sales data into a table named **Sales** that will play the role of the primary fact table in the data model for **Wingtip Sales Analysis.pbix**.

- Open the **InvoiceDetails** query in edit mode.
 - Click on the **Edit Queries** button in the **Home** tab of the ribbon to display the **Power Query Editor** window.
 - Make sure the **InvoiceDetails** query is selected in the **Queries** list on the left-hand side of the **Query Editor** window.

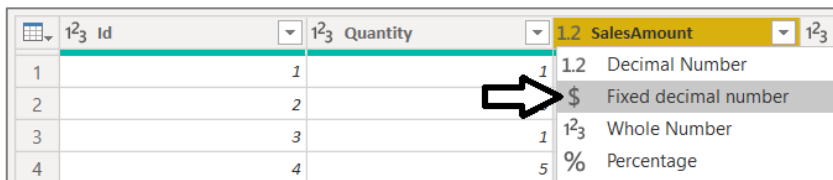
Id	Quantity	SalesAmount	InvoiceId	ProductId
1	1	2.95	1	16
2	2	14.94	2	15
3	3	19.95	3	6
4	4	249.75	3	17
5	5	2.95	4	16

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 in the data model. Try to use query/table names that are more intuitive and easier to understand. In the next step you will rename the **InvoiceDetails** query to **Sales** to indicate that it is a fact table with all the product sales data.

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

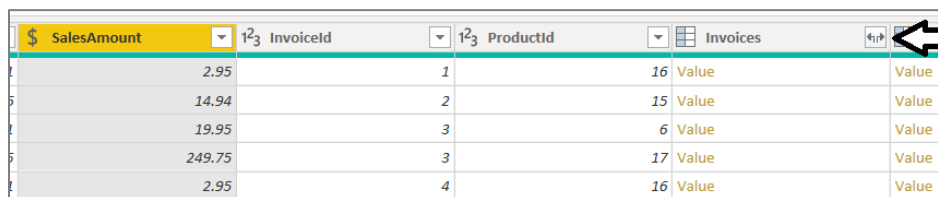


3. Modify the column type of the **SalesAmount** column to the **Fixed Decimal Number** type.

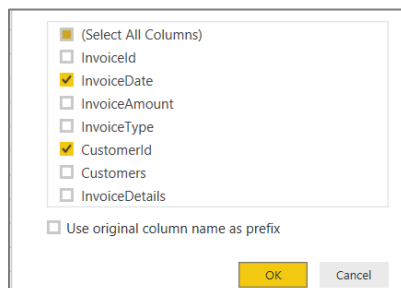


In order for the **Sales** table to play the role of a fact table, the **Sales** query must merge in additional columns from the **Invoices** table. You must pull in the **CustomerId** column so that you can create a direct relationship between the **Sales** table and the **Customers** table. You will also pull in the **InvoiceDate** so that you can analyze data in the **Sales** table over time.

4. 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.
 - c) Select the checkboxes for the **InvoiceDate** column and the **CustomerId** column.
 - d) Make sure to uncheck the checkbox with the caption **Use original column name as prefix**.
 - e) Click the **OK** button to close the dialog and to modify the underlying query.



- f) You should see that the **InvoiceDate** column and the **CustomerId** column have now been added to the **Sales** query results.

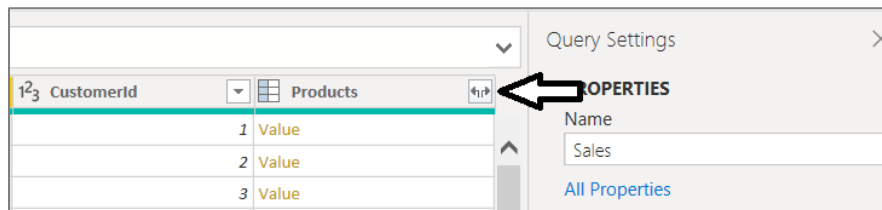
Productid	InvoiceDate	CustomerId	Products
16	1/28/2012 12:00:00 AM	1	Value
15	1/28/2012 12:00:00 AM	2	Value
6	1/28/2012 12:00:00 AM	3	Value
17	1/28/2012 12:00:00 AM	3	Value
16	1/28/2012 12:00:00 AM	4	Value

- g) Change the column type of the **InvoiceDate** to the **Date** type.

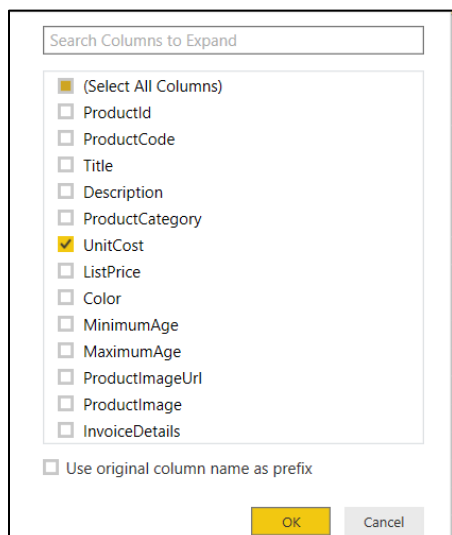
Productid	InvoiceDate	CustomerId
16	1.2 Decimal Number	
15	\$ Fixed decimal number	
6	1.3 Whole Number	
17	% Percentage	
16	Date/Time	
15	Date	
	Time	

5. 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.
- d) Modify the column type of the **SalesAmount** column to the **Fixed Decimal Number** type.

InvoiceDate	CustomerId	UnitCost
1/28/2012	1	0.85
1/28/2012	4	0.85
1/28/2012	6	0.85
1/28/2012	9	0.85
1/28/2012	10	0.85

6. Add a new custom column named **ProductCost** to calculate the product of the **Quantity** field multiplied by the **UnitCost** field.
 - a) Activate the **Add Column** tab in the ribbon.
 - b) Click the **Add Custom Column** button in the ribbon to display the **Add Custom Column** dialog.

SalesAmount	InvoiceId	ProductId	InvoiceDate
1	1	2.95	16
2	1	2.95	16
3	9	26.55	16
4	5	14.75	16

- 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.

UnitCost	ProductCost
0.85	0.85
0.85	0.85
0.85	7.65
0.85	4.25
0.85	6.8

- g) Modify the column type of the **SalesAmount** column to the **Fixed Decimal Number** type.

\$ UnitCost	\$ ProductCost
0.85	0.85
0.85	0.85
0.85	7.65
0.85	4.25

Once the **UnitCost** column has been used to calculate the **ProductCost** value, the column is no longer needed and can be removed.

7. Remove the **UnitCost** column by right-clicking its column header and selecting the **Remove** command.

123 CustomerId	\$ UnitCost	\$ ProductCost
1		
4		
6		
9		
10		

8. Now you should see the **ProductCost** column in the query output, but not the **UnitCost** column.

InvoiceDate	123 CustomerId	\$ ProductCost
1/28/2012	1	0.85
1/28/2012	4	0.85
1/28/2012	6	7.65
1/28/2012	9	4.25

9. Apply your changes to the **Sales** query.
- Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Sales** query.
 - You should be able to see your changes in the **Sales** table in **Data** view in the main Power BI Desktop window.

Wingtip Sales Analysis - Power BI Desktop

Id	Quantity	SalesAmount	InvoiceId	ProductId	InvoiceDate	CustomerId	ProductCost
95	9	179.55	46	6	Saturday, February 4, 2012	46	\$128.25
96	9	179.55	47	6	Saturday, February 4, 2012	47	\$128.25
307	9	179.55	155	6	Thursday, February 23, 2012	142	\$128.25
313	9	179.55	157	6	Thursday, February 23, 2012	114	\$128.25
357	9	179.55	180	6	Saturday, February 25, 2012	116	\$128.25
601	9	179.55	296	6	Saturday, March 10, 2012	240	\$128.25
617	9	179.55	306	6	Saturday, March 10, 2012	130	\$128.25
761	9	179.55	378	6	Monday, March 19, 2012	297	\$128.25
861	9	179.55	427	6	Saturday, March 24, 2012	325	\$128.25
863	9	179.55	428	6	Saturday, March 24, 2012	326	\$128.25
864	9	179.55	429	6	Saturday, March 24, 2012	327	\$128.25
924	9	179.55	460	6	Wednesday, March 28, 2012	344	\$128.25
945	9	179.55	468	6	Wednesday, March 28, 2012	187	\$128.25
1146	9	179.55	572	6	Monday, April 9, 2012	278	\$128.25
1201	9	179.55	596	6	Wednesday, April 11, 2012	26	\$128.25
1316	9	179.55	654	6	Wednesday, April 18, 2012	474	\$128.25

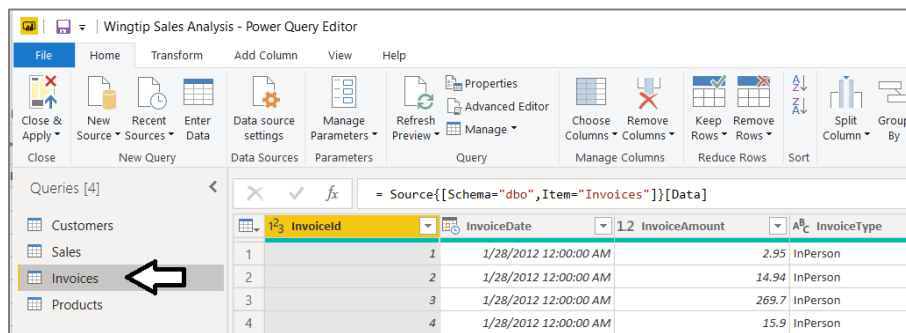
You are now done working with the **Sales** query.

10. Save your work by clicking the **Save** button in the upper left corner of the Power BI Desktop window.

Exercise 5: Using Power Query to Transform and Reshape Invoice Data

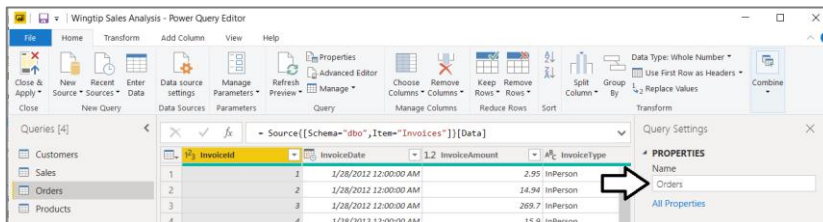
In the following exercise, you will use the **Query Editor** window to modify the **Invoices** query to transform invoice data as it is being loaded into the data model.

1. Open the **Invoices** query in edit mode.
 - a) Click on the **Edit Queries** button in the **Home** tab of the ribbon to display the **Power Query Editor** window.
 - b) Make sure the **Invoices** query is selected in the **Queries** list on the left-hand side of the **Query Editor** window.

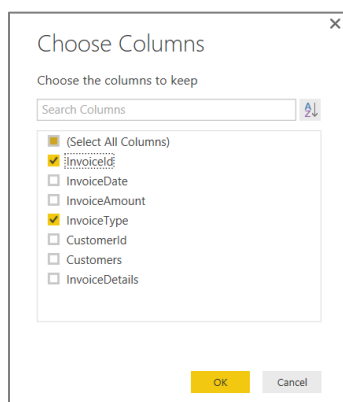


In the last exercise you changed the name of the **InvoiceDetails** to **Sales** to make the data model easier to understand. In this exercise you will change the name of the **Invoices** query to **Order** for the same reason.

2. Update the name of the **Invoices** query to **Orders** by replacing the text in the **Name** textbox in the **Query Settings** pane.

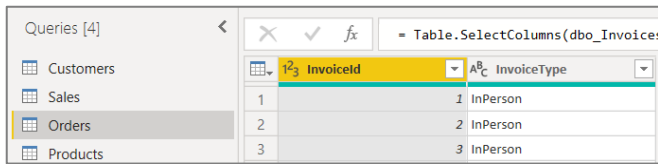


3. Remove the unneeded columns from the **Orders** query.
 - a) Make sure the **Orders** query is selected in the **Queries** list
 - b) Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
 - c) In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns.
 - d) Select the checkboxes for **InvoiceId** and **InvoiceType** as shown in the following screenshot.



- e) Once you have the columns selected, click **OK** to close the **Choose Columns** dialog and to modify the underlying query.

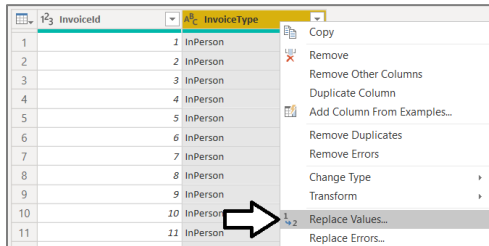
4. You should be able to see that the **Orders** query now only shows the columns that you selected.



	InvoiceId	InvoiceType
1	1	InPerson
2	2	InPerson
3	3	InPerson

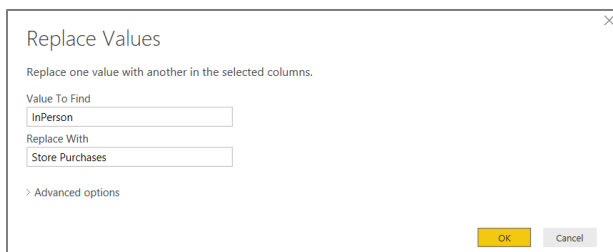
5. Modify the query so that the **InvoiceType** column returns values that are more human-readable.

- a) Right-click the header for the **InvoiceType** column and select **Replace Values....**



	InvoiceId	InvoiceType
1	1	InPerson
2	2	InPerson
3	3	InPerson
4	4	InPerson
5	5	InPerson
6	6	InPerson
7	7	InPerson
8	8	InPerson
9	9	InPerson
10	10	InPerson
11	11	InPerson

- b) In the **Replace Values** dialog, enter a value of **InPerson** in the **Value to Find** textbox.
c) Enter a value of **Store Purchases** in the **Replace With** textbox and click **OK** to add your changes to the underlying query.



Replace Values

Replace one value with another in the selected columns.

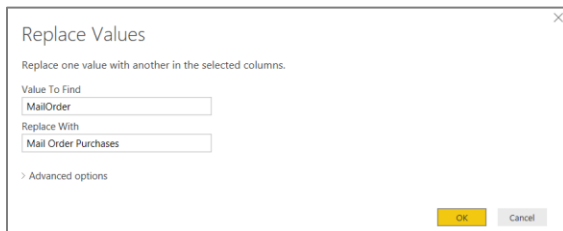
Value To Find
InPerson

Replace With
Store Purchases

> Advanced options

OK Cancel

- d) Add a second **Replace Values** step to replace **MailOrder** with a value of **Mail Order Purchases**.



Replace Values

Replace one value with another in the selected columns.

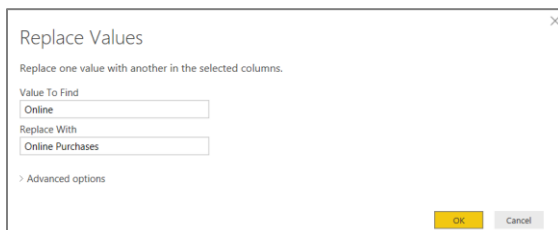
Value To Find
MailOrder

Replace With
Mail Order Purchases

> Advanced options

OK Cancel

- e) Add a third **Replace Values** step to replace **Online** with a value of **Online Purchases**.



Replace Values

Replace one value with another in the selected columns.

Value To Find
Online

Replace With
Online Purchases

> Advanced options

OK Cancel

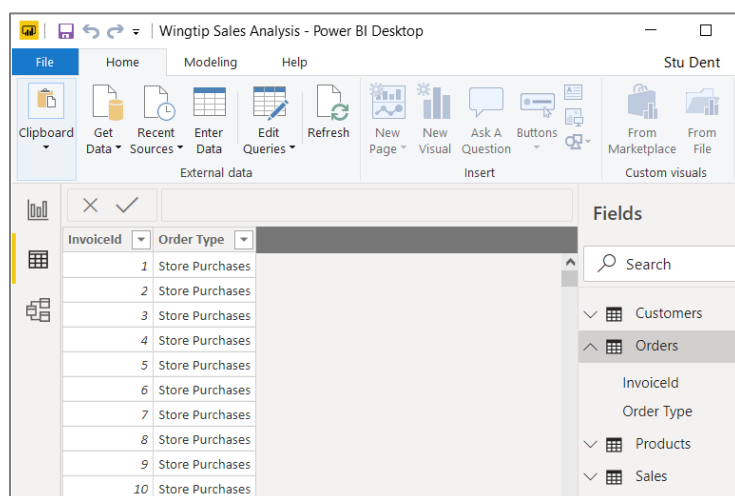
If you scroll down and look at all the rows within the **Orders** query results, you should be able to see that each row has a **InvoiceType** column value of either **Store Purchases**, **Mail Order Purchases** or **Online Purchases**.

6. Change the name of the **InvoiceType** column to **Order Type**.

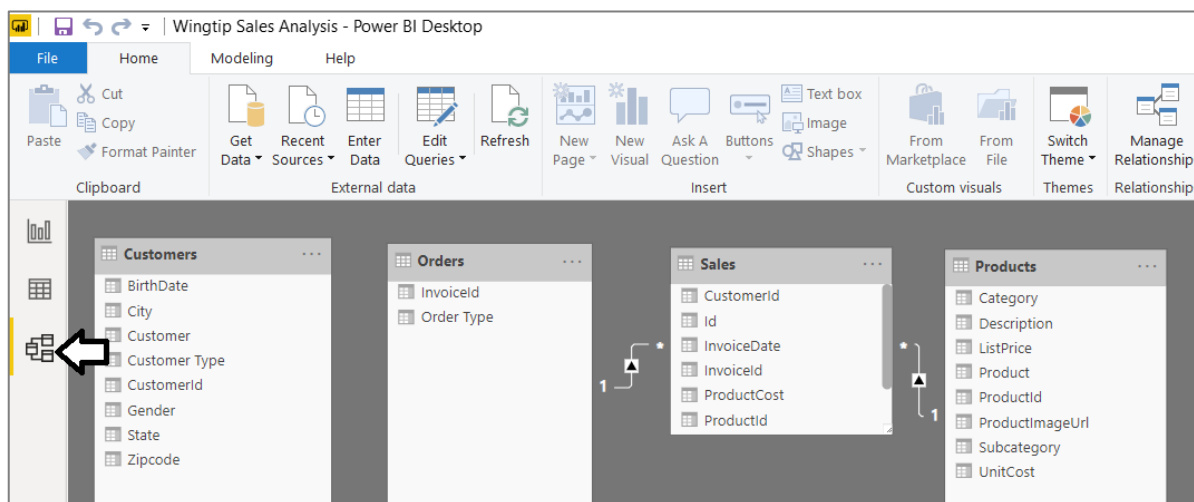
1 ²	InvoiceId	A ^B C	Order Type
1		1	Store Purchases
2		2	Store Purchases
3		3	Store Purchases
4		4	Store Purchases
5		5	Store Purchases

7. Apply your changes to the **Order** query.

- Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Orders** query.
- You should be able to see your changes in the **Orders** table in **Data** view in the main Power BI Desktop window.

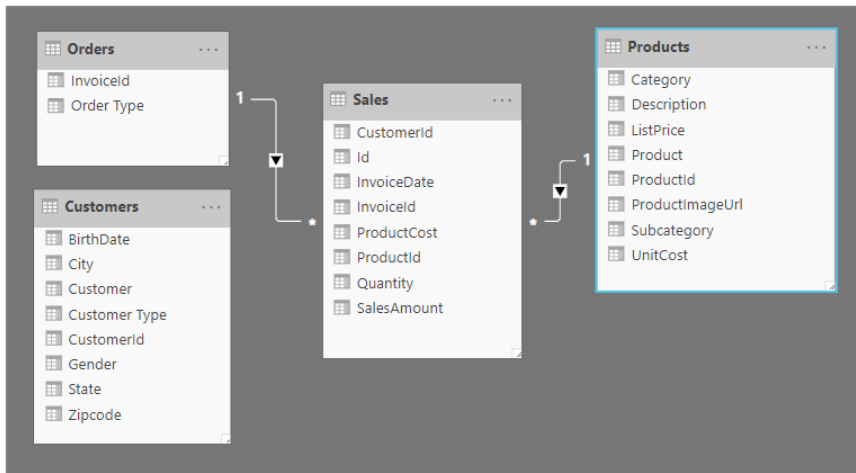


8. Navigate to Model view to see how your updates to the queries in this project have affected the relationships between the tables.



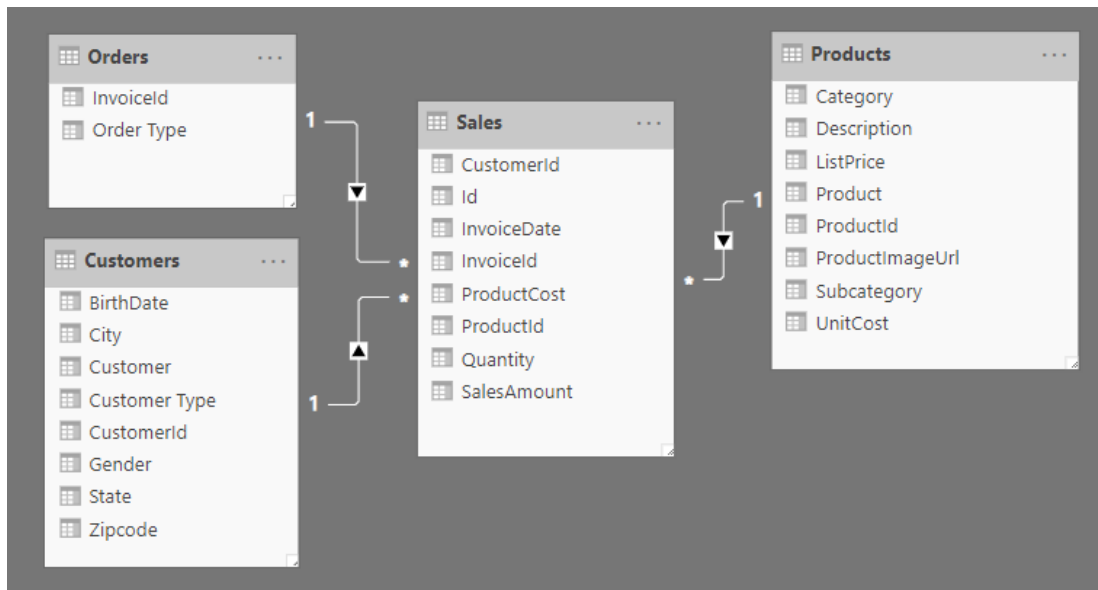
You should be able to see that there is no longer a relationship between the **Customers** table and the **Orders** table. This is due to the fact that you removed the **CustomerId** column from the **Orders** table.

9. Use your mouse to move and resize the four tables in the data model to match the following screenshot.



10. Create a relationship between the **Customers** table and the **Sales** table.

- a) Click on the **CustomerId** column in the **Customers** table and then drag-and-drop it on top of the **CustomerId** column in the **Sales** table to create a new relationship.



- b) You should see a solid line between the **Customers** table and the **Sales** tables indicating a 1-to-many relationship.

11. Save your work by clicking the **Save** button in the upper left corner of the Power BI Desktop window.

You have now completed all the work for this lab. While you haven't yet begun the data modeling phase yet, you have refactored the database tables from **WingtipSalesDB** into a **star schema** which is a best practice in query design for data analysis projects.

NOTE: You will continue to work on this PBIX file named **Wingtip Sales Analysis.pbix** in the next lab and the ones that follow.