

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.

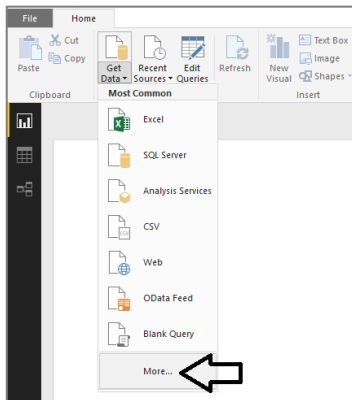
Exercise 1: Importing Data from a SQL Azure Database

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.

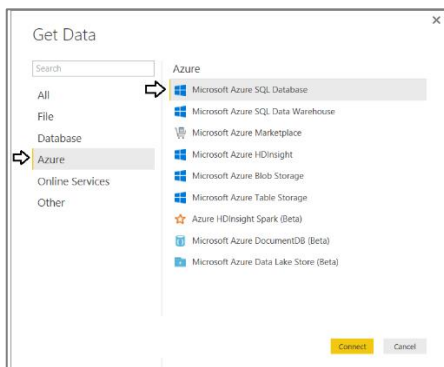
1. Launch Power BI Desktop to start a new project.
2. Save the new project as **Wingtip Sales Analysis.pbix** using the following path.

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

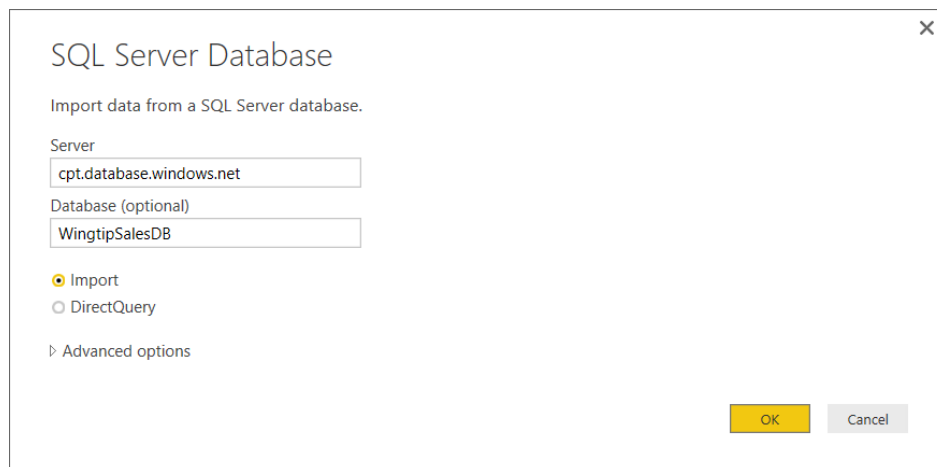
3. Drop down the **Get Data** menu button on the ribbon and click **More....**



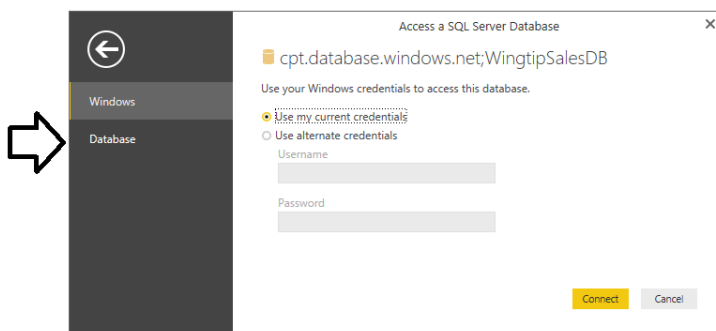
4. On the **Get Data** dialog, select **Azure** in the list on the left. Next, select **Microsoft SQL Azure Database** on the right and then click the **Connect** button.



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.

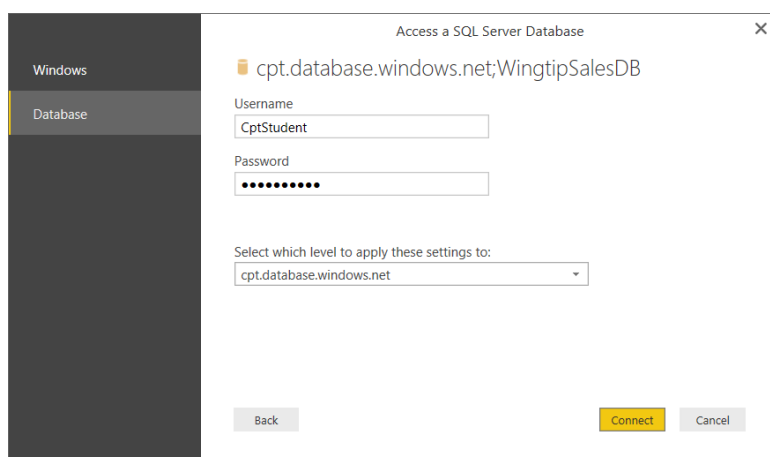


6. At this point, you will be prompted by the **Access a SQL Server Database** dialog. Click on **Database** on the left side of the dialog so that you can enter the credentials for a standard SQL account instead of using Windows authentication.

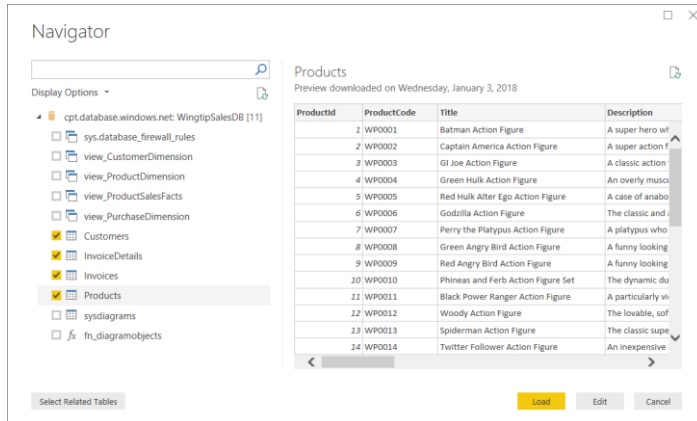


If you don't switch the selection in the dialog above from **Windows** to **Database**, things will not work correctly.

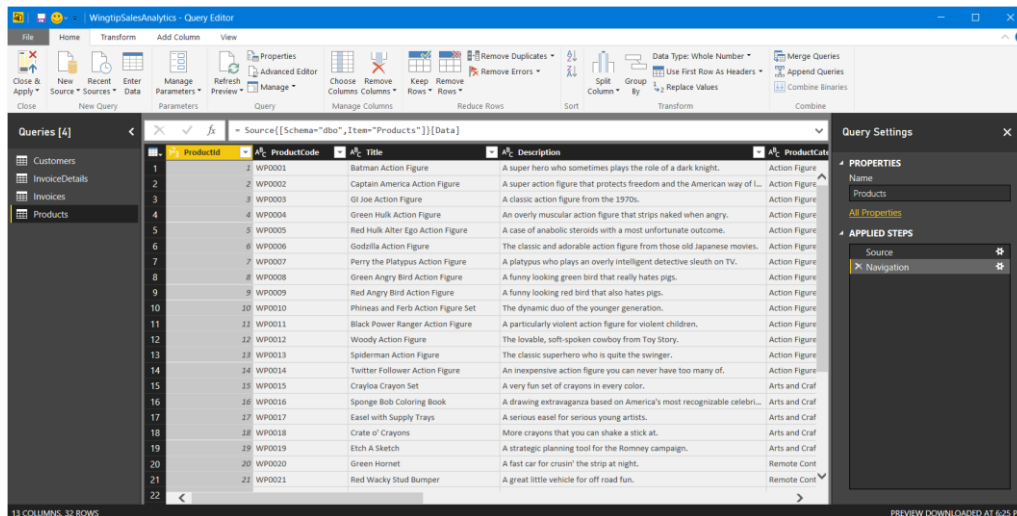
7. Enter the following credentials for a SQL user account that has been configured with read access to the database.
- a) Username: **CptStudent**
 - b) Password: **pass@word1**
8. Once you have entered the credentials the **Access a SQL Server Database** dialog, click the **Connect** button to continue.



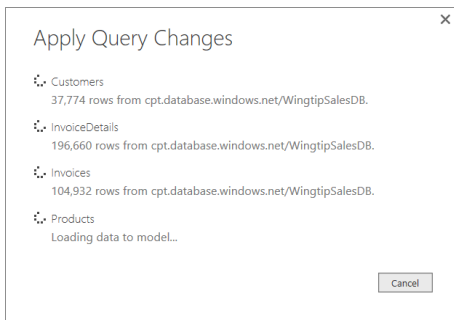
9. At this point, Power BI Desktop should be able to establish a connection to the database and then prompt you with the **Navigator** dialog. The **Navigator** dialog allows you to select the tables you would like to import into your PBIX project.
10. In the **Navigator** dialog, select the **Customers** 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.



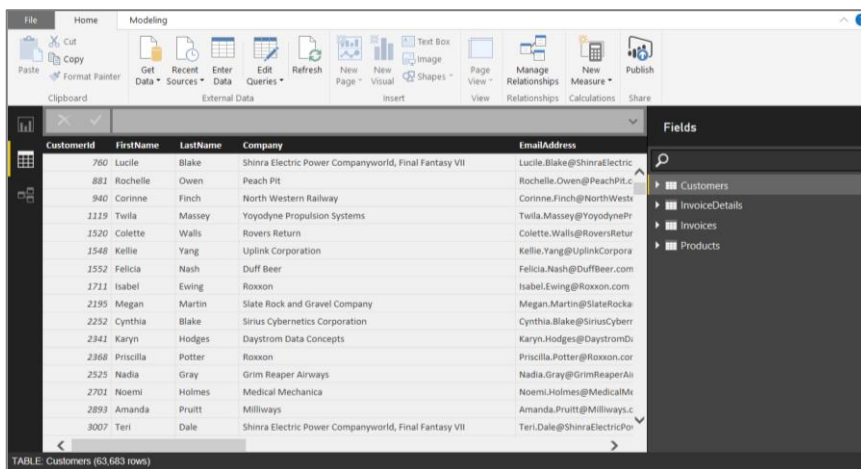
11. When you inspect the **Query Editor** window, you should now be able to observe that Power BI Desktop has created a new query for each of the four tables that you selected in the **Navigator** dialog. You will not begin to edit these queries until the next exercise. However, now you should take a 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.



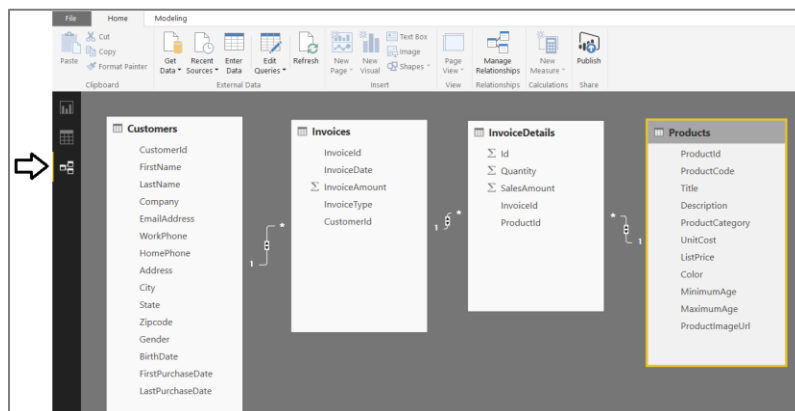
12. Click the **Close and Apply** button in the **Query Editor** window. This will close the **Query Editor** window and start the process of extracting the data from the SQL Azure database and importing it into the current project's data model using a local cache on your student workstation. The **Apply Query Changes** dialog will be displayed with spinning icons as Power BI Desktop imports that data.



13. 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.



14. Click on the bottom icon in the sidebar to navigate to relationship view.
- You should be able to see each of the four tables.
 - 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.
 - Use your mouse to move and resize the tables in relationship view so you can see all the fields in each table.

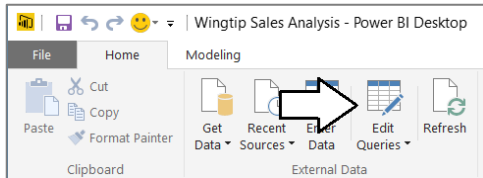


15. Save your work in the current project by clicking the Save button in the upper left corner of the Power BI Desktop window.

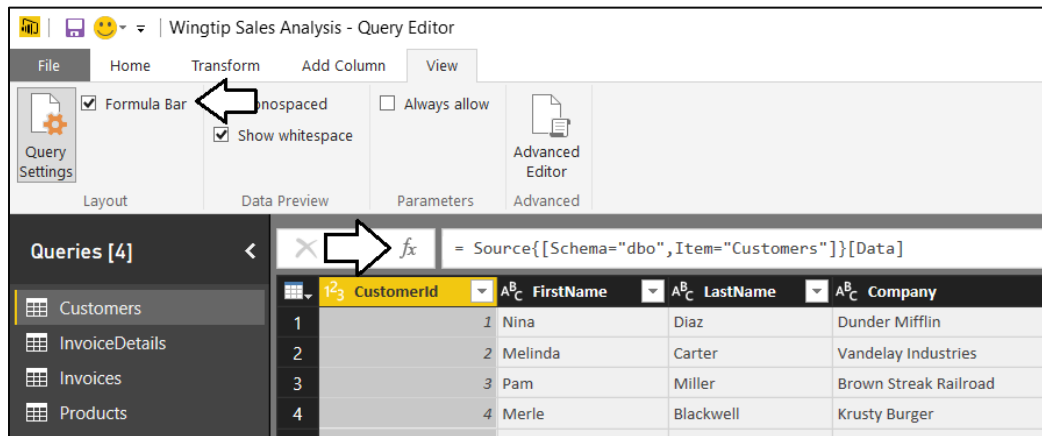
Exercise 2: Design Queries to Transform and Reshape Customer Sales Data

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.

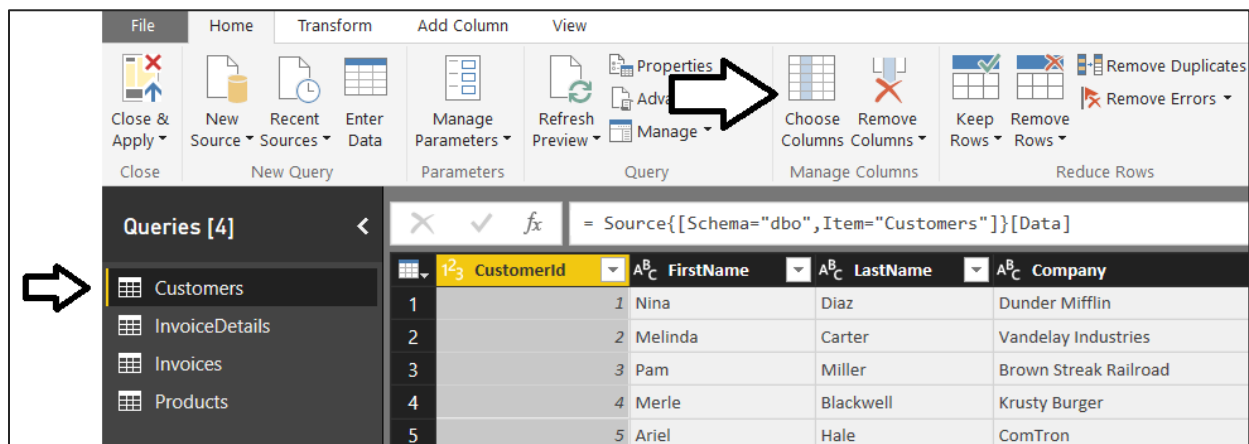
1. Make sure you have the **Wingtip Sales Analysis.pbix** project open that you started in the previous exercise.
2. Click on the **Edit Queries** button in the ribbon to display the Query Editor window.



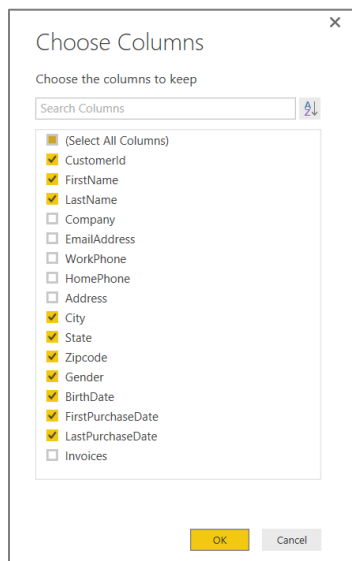
3. Make sure the query formula bar is visible in the Query Editor window.
 - a) In the Query Editor window, navigate to the **View** tab.
 - b) Check the **Formula Bar** checkbox.
 - c) You should now see the query formula bar displayed just above the query results.



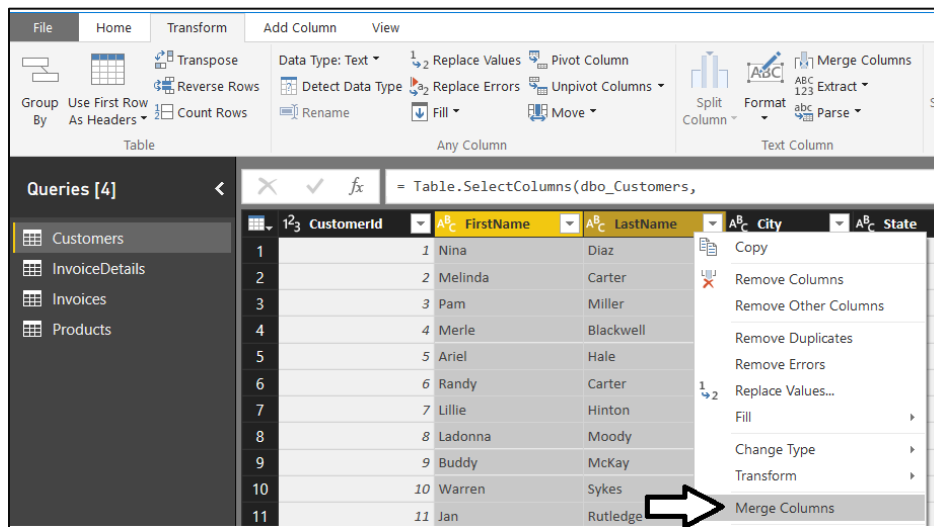
4. Select the desired set of columns from the **Customers** table.
 - a) Make sure the **Customers** query is selected in the **Queries** list on the left-hand side of the Query Editor window.
 - b) Navigate to the **Home** tab and click the **Choose Columns** button in the ribbon to display the **Choose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



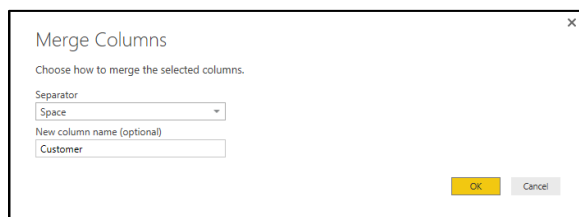
- c) In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all column. Next, select the checkboxes for the following columns.
- i) **CustomerId**
 - ii) **FirstName**
 - iii) **LastName**
 - iv) **City**
 - v) **State**
 - vi) **Zipcode**
 - 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**.
- a) Select the **FirstName** column by clicking on its column header.
 - b) Next, hold down the **SHIFT** key and select the **LastName** column by clicking on its column header.
 - c) 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**.

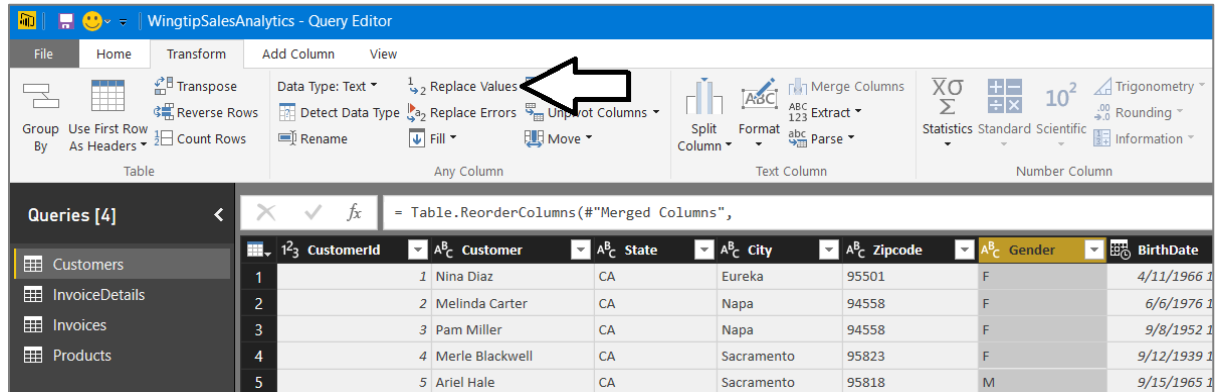
	Customerid	Customer	City	State
1	1	Nina Diaz	Eureka	CA
2	2	Melinda Carter	Napa	CA
3	3	Pam Miller	Napa	CA
4	4	Merle Blackwell	Sacramento	CA
5	5	Ariel Hale	Sacramento	CA
6	6	Randy Carter	Sacramento	CA

6. 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
1	1	Nina Diaz	CA	Eureka
2	2	Melinda Carter	CA	Napa
3	3	Pam Miller	CA	Napa
4	4	Merle Blackwell	CA	Sacramento
5	5	Ariel Hale	CA	Sacramento
6	6	Randy Carter	CA	Sacramento

7. Modify the query so that the **Gender** column returns values of **Male** and **Female** instead of **M** and **F**.

- Make sure the **Transform** tab is the active tab in the ribbon.
- Select the **Gender** column by clicking its column header.
- Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.



- 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

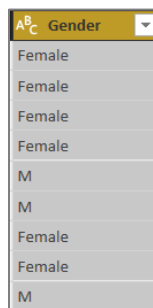
Female

Advanced options

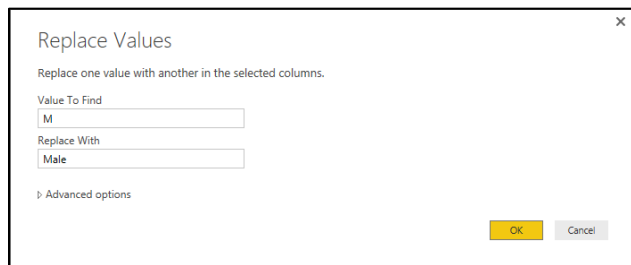
OK

Cancel

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



- Make sure the **Gender** column is still selected.
- Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.
- 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**.

Gender	BirthDate	FirstPurchase
Female	4/11/1966 12:00:00 AM	1/28/2
Female	6/6/1976 12:00:00 AM	1/28/2
Female	9/8/1952 12:00:00 AM	1/28/2
Female	9/12/1939 12:00:00 AM	1/28/2
Male	9/15/1965 12:00:00 AM	1/28/2
Male	7/14/1953 12:00:00 AM	1/28/2
Female	2/3/1992 12:00:00 AM	1/28/2
Female	4/5/1949 12:00:00 AM	1/28/2
Male	5/10/1989 12:00:00 AM	1/28/2
Male	6/17/1960 12:00:00 AM	1/28/2
Female	11/26/1981 12:00:00 AM	1/30/2
Male	3/26/1973 12:00:00 AM	1/30/2
Male	4/5/1988 12:00:00 AM	1/30/2

PROPERTIES

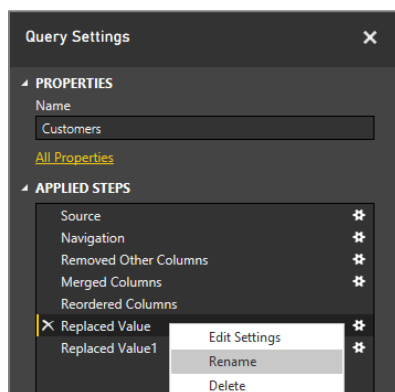
Name
Customers

[All Properties](#)

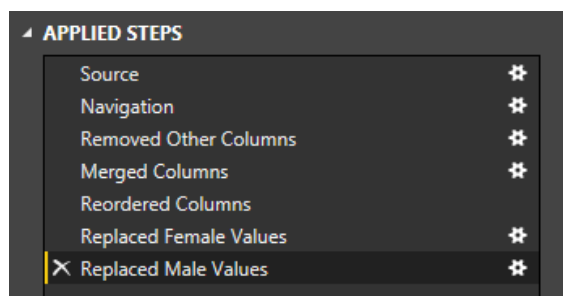
APPLIED STEPS

- Source
- Navigation
- Removed Other Columns
- Merged Columns
- Reordered Columns
- Replaced Value
- Replaced Value1

- 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**.



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.

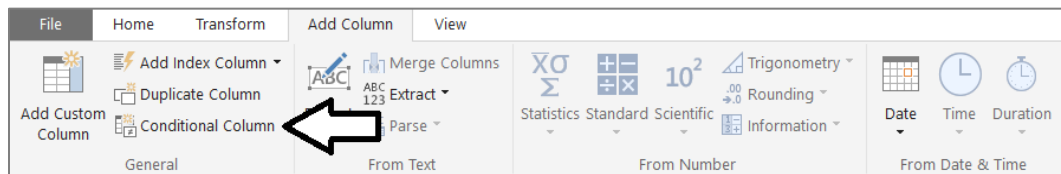
8. Change the column type of **BirthDate**, **FirstPurchaseDate** and **LastPurchaseDate** from **Date/Time** to **Date**.
 - a) Use the column type drop down on the left-hand side of the **BirthDate** column to configure the column using the **Date** type.

A ^B C Gender	BirthDate	FirstPurchaseDate
Female	1.2 Decimal Number	1/28/2012 12:00:00 AM
Female	\$ Fixed Decimal Number	1/28/2012 12:00:00 AM
Female	123 Whole Number	1/28/2012 12:00:00 AM
Female	Date/Time	1/28/2012 12:00:00 AM
Male	Date	1/28/2012 12:00:00 AM
Male	Time	1/28/2012 12:00:00 AM
Female	Date/Time/Timezone	1/28/2012 12:00:00 AM
Female	Duration	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.

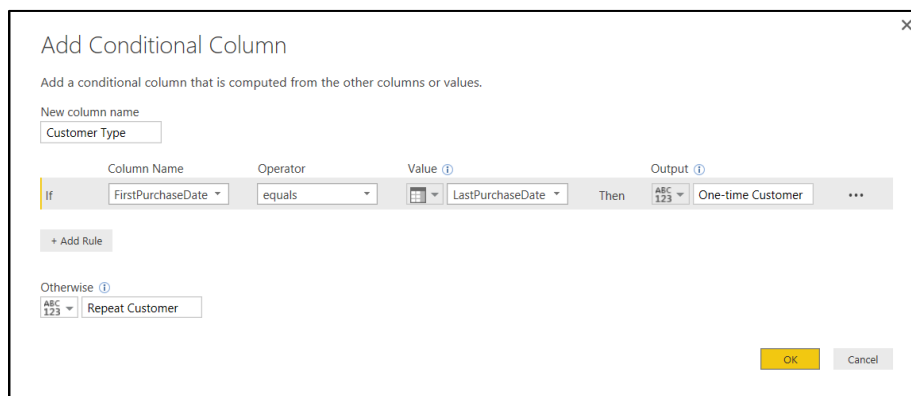
BirthDate	FirstPurchaseDate	LastPurchaseDate
4/11/1966	1/28/2012	1/28/2012
6/6/1976	1/28/2012	1/28/2012
9/8/1952	1/28/2012	1/28/2012
9/12/1939	1/28/2012	1/28/2012
9/15/1965	1/28/2012	1/28/2012
7/14/1953	1/28/2012	1/28/2012
2/3/1992	1/28/2012	1/28/2012

9. 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.



- 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
2/23/2012	11/24/2015	Repeat Customer
2/23/2012	10/14/2015	Repeat Customer
2/23/2012	4/3/2015	Repeat Customer
2/23/2012	2/23/2012	One-time Customer
2/23/2012	2/23/2012	One-time Customer
2/24/2012	2/24/2012	One-time Customer

10. 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.
 - a) Select the **FirstPurchaseDate** column by clicking its column header.
 - b) Hold down the **SHIFT** key and click the column header for **LastPurchaseDate** so that both columns are selected.
 - c) Right click the one of the selected columns and click the **Remove Columns**.

BirthDate	FirstPurchaseDate	LastPurchaseDate	Customer Type
2/16/1982	2/23/2012	11/24/	
6/2/1966	2/23/2012	10/14/	
9/3/1973	2/23/2012	4/3/	
7/19/1963	2/23/2012	2/23/	

- 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
Male	9/15/1965	One-time Customer
Male	7/14/1953	One-time Customer
Female	2/3/1992	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.

Customer Type
4/11/1966 One-time Customer
6/6/1976 One-time Customer

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

BirthDate	Customer Type
4/11/1966	1.2 Decimal Number
6/6/1976	\$ Fixed Decimal Number
9/8/1952	123 Whole Number
9/12/1939	Date/Time
9/15/1965	Date
7/14/1953	Time
2/3/1992	Date/Time/Timezone
4/5/1949	Duration
5/17/1980	Text
11/26/1981	True/False
3/26/1973	Binary

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

- 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.



- 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.

Customerid	City	State	Zipcode	Gender	BirthDate	Customer	Customer Type
760	San Jose	CA	95133	Female	Saturday, March 16, 1968	Lucile Blake	One-time Cu
881	San Jose	CA	95133	Female	Sunday, July 19, 1942	Rochelle Owen	One-time Cu
940	San Jose	CA	95133	Female	Sunday, March 7, 1943	Corinne Finch	One-time Cu
1119	San Jose	CA	95133	Female	Monday, September 3, 1990	Twila Massey	One-time Cu
1548	San Jose	CA	95133	Female	Thursday, July 14, 1955	Kellie Yang	One-time Cu
2195	San Jose	CA	95133	Female	Sunday, March 25, 1951	Megan Martin	One-time Cu
2252	San Jose	CA	95133	Female	Wednesday, April 3, 1946	Cynthia Blake	One-time Cu
2341	San Jose	CA	95133	Female	Monday, May 2, 1960	Karyn Hodges	One-time Cu

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

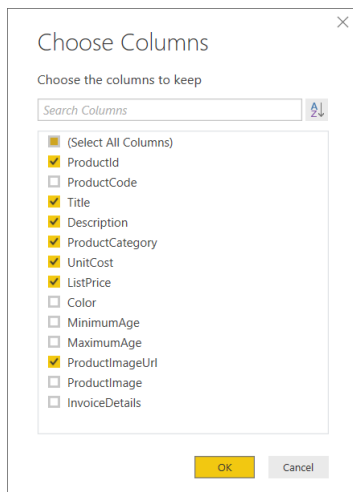
Exercise 3: Using Power Query to Transform and Reshape Product Data

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

- Make sure you have the **Wingtip Sales Analysis.pbix** project open that you started in the previous exercise.
- Click on the **Edit Queries** button in the ribbon to display the **Query Editor** window.
- Make sure the **Products** query is selected in the **Queries** list on the left-hand side of the **Query Editor** window.

Productid	ProductCode	Title	Description
1	WP0001	Batman Action Figure	A super hero who sometim
2	WP0002	Captain America Action Figure	A super action figure that p
3	WP0003	GI Joe Action Figure	A classic action figure from
4	WP0004	Green Hulk Action Figure	An overly muscular action f
5	WP0005	Red Hulk Alter Ego Action Figure	A case of anabolic steroids
6	WP0006	Godzilla Action Figure	The classic and adorable ac
7	WP0007	Perry the Platypus Action Figure	A platypus who plays an ov
8	WP0008	Green Angry Bird Action Figure	A funny looking green bird

4. 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**, **Description**, **ProductCategory**, **UnitCost**, **ListPrice** and **ProductImageUrl** as shown in the following screenshot.



- c) Click the **OK** button to close the **Choose Columns** dialog and to modify the underlying query.
5. 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 ProductId	A ^B _C Product
1		1 Batman Action Figure
2		2 Captain America Action Figure
3		3 GI Joe Action Figure
4		4 Green Hulk Action Figure
5		5 Red Hulk Alter Ego Action Figure
6		6 Godzilla Action Figure
7		7 Perry the Platypus Action Figure
8		8 Green Angry Bird Action Figure

6. 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.

A ^B C Description	A ^B C ProductCategory	1.2 UnitCost	1.2 ListPrice	A ^B C ProductImageUri
A super hero who sometimes plays t...	Action Figures > Tough Gi		14.95	http://classresources.
A super action figure that protects fr...	Action Figures > Tough Gi		12.95	http://classresources.
A classic action figure from the 1970s.	Action Figures > Tough Gi		14.95	http://classresources.
An overly muscular action figure that...	Action Figures > Tough Gi		9.95	http://classresources.
A case of anabolic steroids with a mo...	Action Figures > Tough Gi		9.95	http://classresources.
The classic and adorable action figur...	Action Figures > Tough Gi		19.95	http://classresources.
A platypus who plays an overly intelli...	Action Figures > Cute and		21.95	http://classresources.
A funny looking green bird that reall...	Action Figures > Cute and		4.95	http://classresources.
A funny looking red bird that also ha...	Action Figures > Cute and		14.95	http://classresources.
The dynamic duo of the younger gen...	Action Figures > Cute and		19.95	http://classresources.
A particularly violent action figure fo...	Action Figures > Cute and		7.5	http://classresources.
The lovable, soft-spoken cowboy fro...	Action Figures > Cute and		9.95	http://classresources.
The classic superhero who is quite th...	Action Figures > T			
An inexpensive action figure you can...	Action Figures > Cute and			

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

Split Column by Delimiter

Specify the delimiter used to split the text column.

Select or enter delimiter

--Custom--

>

Split

☐ At the left-most delimiter
 ☐ At the right-most delimiter
 ☒ At each occurrence of the delimiter

Advanced options

OK

Cancel

- 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**.

A ^B C Description	A ^B C ProductCategory.1	A ^B C ProductCategory.2
A super hero who sometimes plays t...	Action Figures	Tough Guys
A super action figure that protects fr...	Action Figures	Tough Guys
A classic action figure from the 1970s.	Action Figures	Tough Guys
An overly muscular action figure that...	Action Figures	Tough Guys
A case of anabolic steroids with a mo...	Action Figures	Tough Guys
The classic and adorable action figur...	Action Figures	Tough Guys
A platypus who plays an overly intelli...	Action Figures	Cute and Huggable
A funny looking green bird that reall...	Action Figures	Cute and Huggable
A funny looking red bird that also ha...	Action Figures	Cute and Huggable
The dynamic duo of the younger gen...	Action Figures	Cute and Huggable

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

A ^B C Category	A ^B C 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

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

7. 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**.

1.2 UnitCost	1.2 ListPrice
1.2 Decimal Number	14.95
\$ Fixed Decimal Number	12.95
123 Whole Number	14.95
Date/Time	9.95
Date	9.95
Time	19.95

- 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

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

8. Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Products** query. Power BI Desktop will display the **Apply Query Changes** dialog while importing the data and transforming it to load it into the data model.
9. After the query changes have been applied, you should be able to see the results of your changes in the **Products** table that has been loaded into the project's data model.

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

You might have noticed that the columns that are displayed in the **Products** table do not appear in the same order as they appear in the **Products** query. Power BI Desktop currently provides no ability to control and change the order of columns or fields as they are defined in a table.

10. 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

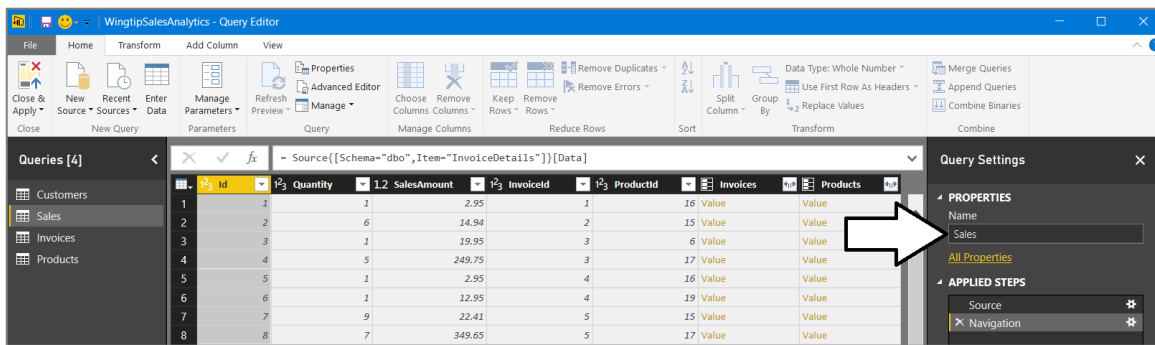
In the following exercise, you will use the Query Editor to modify the **InvoiceDetails** query to transform data related to sales data before it is loaded into the data model.

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 **InvoiceDetails** query is selected in the **Queries** list on the left-hand side of the **Query Editor** window.

Id	Quantity	SalesAmount	InvoiceId	ProductId	Invoices
1	2	2.95	1	16	Value
2	6	14.94	2	15	Value
3	1	19.95	3	6	Value
4	5	249.75	3	17	Value
5	1	2.95	4	16	Value
6	1	12.95	4	19	Value
7	9	22.41	5	15	Value
8	7	349.65	5	17	Value
9	7	90.65	6	19	Value

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.

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



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

	Id	InvoiceId	Quantity	SalesAmount
1	1	1	1	2.95
2	2	2	6	14.94
3	3	3	1	19.95
4	4	3	5	249.75
5	5	4	1	2.95
6	6	4	1	12.95

6. 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
1	1	1	16	1
2	2	2	15	6
3	3	3	6	1
4	4	3	17	5
5	5	4	16	1
6	6	4	19	1

Keep in mind that moving columns in this fashion only affects the Query Editor window. It will not have any effect on how the columns are ordered when you view the table created by this query in the main Power BI Desktop window.

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

Quantity	SalesAmount
1	1.2 Decimal Number
6	\$ Fixed Decimal Number
1	1.3 Whole Number
5	Date/Time
1	Date
1	Time
9	Date/Time/Timezone

8. 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.

Quantity	SalesAmount	Invoices
1	2.95	Value
6	14.94	Value

- 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.

\$ SalesAmount	InvoiceDate	CustomerId	Products
2.95	1/28/2012 12:00:00 AM	1	Value
14.94	1/28/2012 12:00:00 AM	2	Value
19.95	1/28/2012 12:00:00 AM	3	Value
249.75	1/28/2012 12:00:00 AM	3	Value
2.95	1/28/2012 12:00:00 AM	4	Value

9. Use the mouse to drag and drop the **CustomerId** column to move it to the right of the **InvoiceId** column.

Id	InvoiceId	CustomerId	ProductId
1	1	1	16
2	2	2	15
3	3	3	6
4	4	3	17
5	5	4	16

10. Use the mouse to drag and drop the **InvoiceDate** column to move it to the right of the **ProductId** column.

Id	InvoiceId	CustomerId	ProductId	InvoiceDate	Quantity
1	1	1	16	1/28/2012 12:00:00 AM	1
2	2	2	15	1/28/2012 12:00:00 AM	6
3	3	3	6	1/28/2012 12:00:00 AM	1
4	4	3	17	1/28/2012 12:00:00 AM	5
5	5	4	16	1/28/2012 12:00:00 AM	1
6	6	4	19	1/28/2012 12:00:00 AM	1

11. Change the column type of the **InvoiceDate** to the **Date** type.

	Id	InvoiceId	CustomerId	ProductId	InvoiceDate	Quantity
1	1	1	1	16	1.2	Decimal Number
2	2	2	2	15	\$	Fixed Decimal Number
3	3	3	3	6	123	Whole Number
4	4	3	3	17		Date/Time
5	5	4	4	16		Date
6	6	4	4	19		Time
7	7	5	5	15		Date/Time/Timezone

12. Change the name of the **InvoiceDate** column to **PurchaseDate**.

	Id	InvoiceId	CustomerId	ProductId	PurchaseDate	Quantity
1	1	1	1	16	1/28/2012	1
2	2	2	2	15	1/28/2012	6
3	3	3	3	6	1/28/2012	1
4	4	3	3	17	1/28/2012	5
5	5	4	4	16	1/28/2012	1
6	6	4	4	19	1/28/2012	1

13. 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.

\$ SalesAmount	Products
2.95	Value
14.94	Value
19.95	Value

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.

Search Columns to Expand

☒ (Select All Columns)
 ☐ ProductId
 ☐ ProductCode
 ☐ Title
 ☐ Description
 ☐ ProductCategory
 ☒ UnitCost
 ☐ ListPrice
 ☐ Color
 ☐ MinimumAge
 ☐ MaximumAge
 ☐ ProductImageUrl
 ☐ ProductImage
 ☐ InvoiceDetails

☐ Use original column name as prefix

OK Cancel

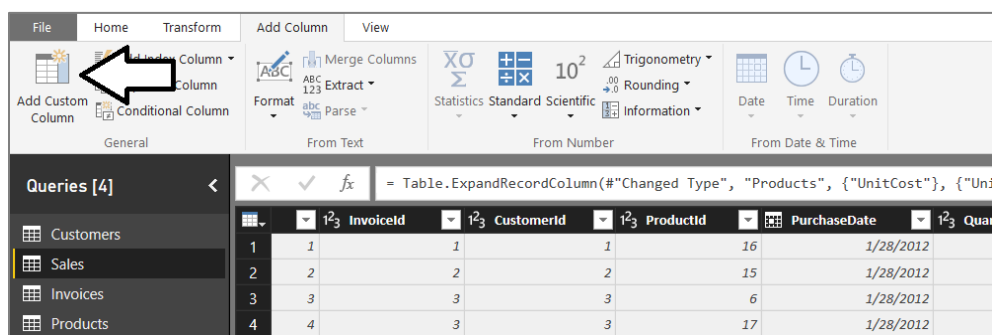
c) You should see that the **UnitCost** column has now been added to the **Sales** query results.

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

¹²³ Quantity	\$ SalesAmount	\$ UnitCost
1	2.95	0.85
1	2.95	0.85
9	26.55	0.85
5	14.75	0.85
8	23.6	0.85

14. Add a new custom column named **ProductCost** to calculate the product of the **Quantity** field multiplied by the **UnitCost** field.

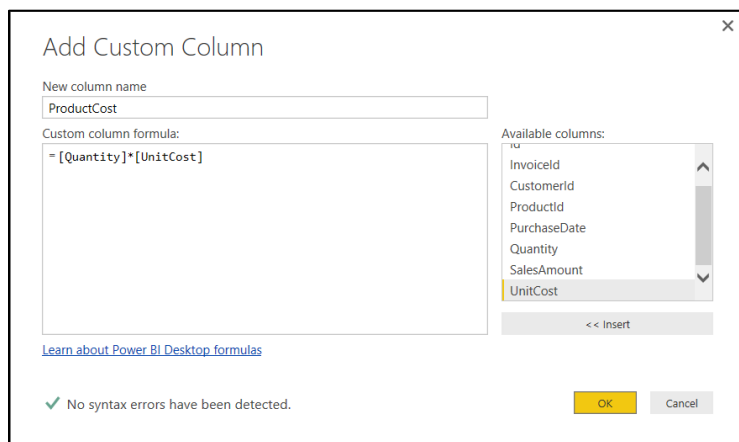
- a) Activate the **Add Column** tab in the ribbon.
b) Click the **Add Custom Column** button in the ribbon to display the **Add Custom Column** dialog.



- 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.

\$ SalesAmount	\$ UnitCost	ABC 123 ProductCost
2.95	0.85	0.85
2.95	0.85	0.85
26.55	0.85	7.65

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

\$ SalesAmount	\$ UnitCost	\$ ProductCost
2.95	0.85	0.85
2.95	0.85	0.85
26.55	0.85	7.65
14.75	0.85	4.25

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.

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

\$ SalesAmount	\$ UnitCost	\$ ProductCost
2.95		
2.95		
26.55		
14.75		

16. Now you should see the **SalesAmount** and **ProductCost** columns in the query output, but not the **UnitCost** column.

123 Quantity	\$ SalesAmount	\$ ProductCost
1	2.95	0.85
1	2.95	0.85
9	26.55	7.65
5	14.75	4.25
8	23.6	6.8
9	26.55	7.65

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

17. Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Sales** query. Power BI Desktop will display the **Apply Query Changes** dialog while importing the data and transforming it to load it into the data model.
18. 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.

Id	Quantity	SalesAmount	InvoiceId	ProductId	CustomerId	PurchaseDate	ProductCost
95	9	179.55	46	6	46	Saturday, February 4, 2012	\$128.25
96	9	179.55	47	6	47	Saturday, February 4, 2012	\$128.25
307	9	179.55	155	6	142	Thursday, February 23, 2012	\$128.25
313	9	179.55	157	6	114	Thursday, February 23, 2012	\$128.25
357	9	179.55	180	6	116	Saturday, February 25, 2012	\$128.25
601	9	179.55	296	6	240	Saturday, March 10, 2012	\$128.25
617	9	179.55	306	6	130	Saturday, March 10, 2012	\$128.25
761	9	179.55	378	6	297	Monday, March 19, 2012	\$128.25
861	9	179.55	427	6	325	Saturday, March 24, 2012	\$128.25
863	9	179.55	428	6	326	Saturday, March 24, 2012	\$128.25
864	9	179.55	429	6	327	Saturday, March 24, 2012	\$128.25

Unfortunately, the current version of Power BI Desktop does not retain the order of the columns that you configured in the **Sales** query.

19. 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

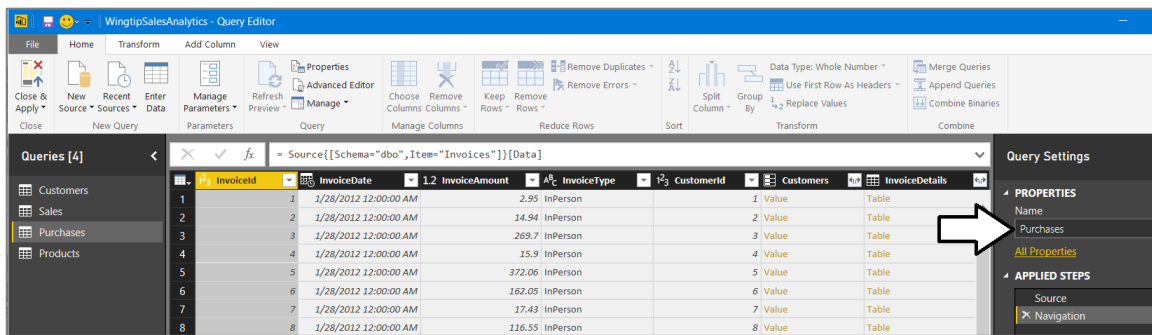
In the following exercise, you will use the **Query Editor** window to modify the **Invoices** query to transform invoice data as it is being loaded into the data model.

1. 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 **Invoices** query is selected in the **Queries** list on the left-hand side of the Query Editor window.

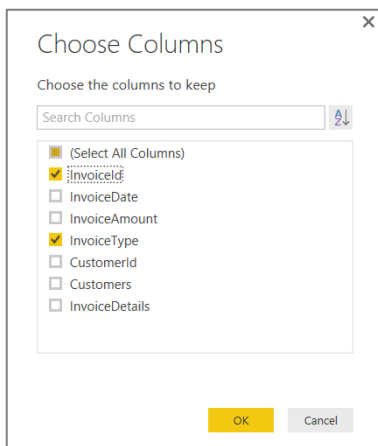
InvoiceId	InvoiceDate	InvoiceAmount	InvoiceType	CustomerId	Customers	InvoiceDetails
1	1/28/2012 12:00:00 AM	2.95	InPerson	1	Value	Table
2	1/28/2012 12:00:00 AM	14.94	InPerson	2	Value	Table
3	1/28/2012 12:00:00 AM	269.7	InPerson	3	Value	Table
4	1/28/2012 12:00:00 AM	15.9	InPerson	4	Value	Table
5	1/28/2012 12:00:00 AM	372.06	InPerson	5	Value	Table
6	1/28/2012 12:00:00 AM	162.05	InPerson	6	Value	Table
7	1/28/2012 12:00:00 AM	17.43	InPerson	7	Value	Table
8	1/28/2012 12:00:00 AM	116.55	InPerson	8	Value	Table

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 **Purchases** for the same reason.

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



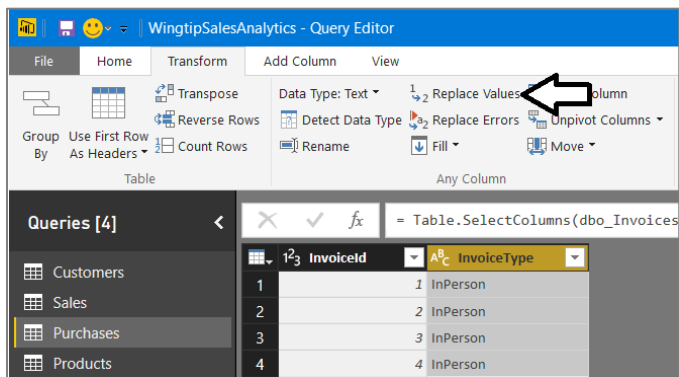
5. Make sure the **Purchases** query is selected in the **Queries** list on the left-hand side of the Query Editor window. Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
6. In the **Choose Columns** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkboxes for **InvoiceId** and **InvoiceType** as shown in the following screenshot.



7. Once you have the columns selected, click **OK** to close the **Choose Columns** dialog and to modify the underlying query.
8. You should be able to see that the Query Editor window now only shows the columns that you selected.

	1 ² InvoiceId	A ^B 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

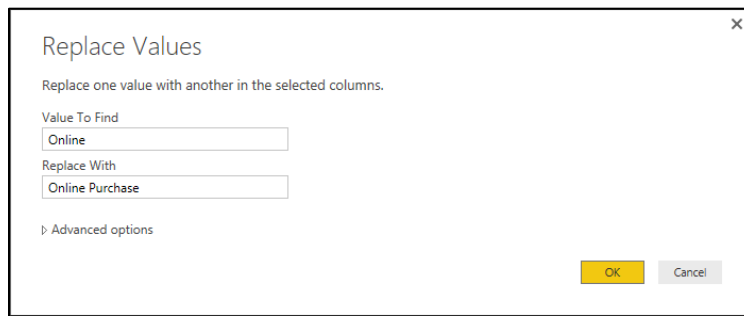
9. Modify the query so that the **InvoiceType** column returns values that are more human readable.
 - a) Make sure the **Transform** tab is the active tab in the ribbon.
 - b) Select the **InvoiceType** column by clicking its column header.
 - c) Click the **Replace Values** button in the ribbon to 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.



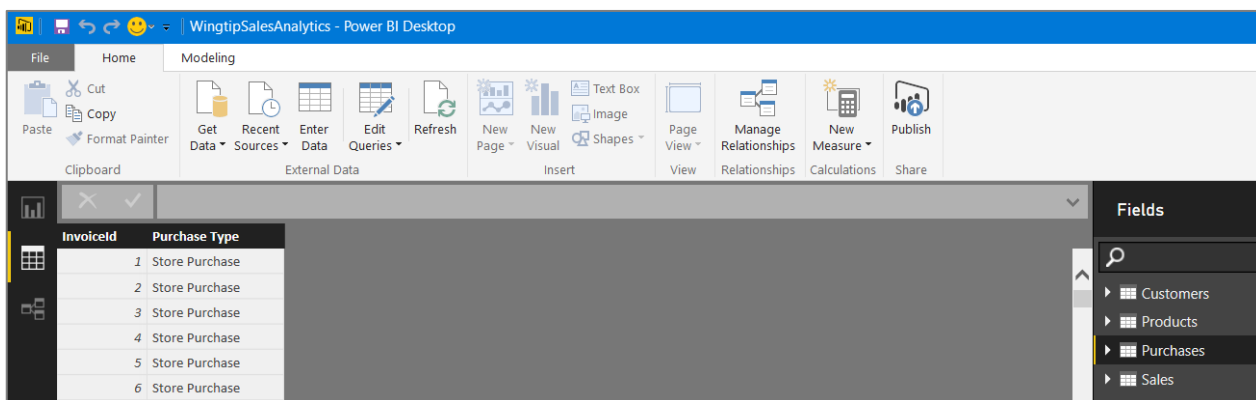
- k) If you scroll down and look at all the rows within the **Purchases** table, you should be able to see that each row has a **InvoiceType** column value of either **Store Purchase**, **Mail Order Purchase** or **Online Purchase**.

10. Change the name of the **InvoiceType** column to **Purchase Type**.

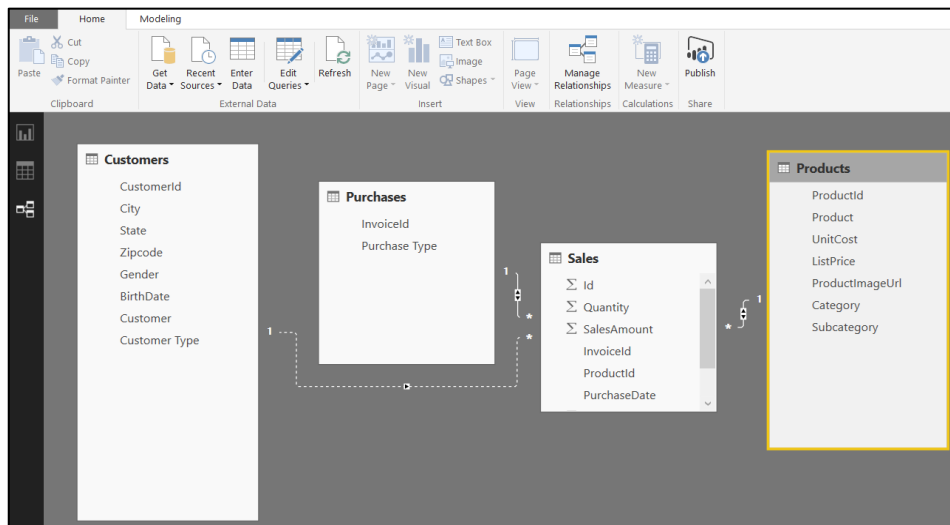
	123 InvoiceId	ABC Purchase Type
1		1 Store Purchase
2		2 Store Purchase
3		3 Store Purchase
4		4 Store Purchase
5		5 Store Purchase
6		6 Store Purchase
7		7 Store Purchase
8		8 Store Purchase

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

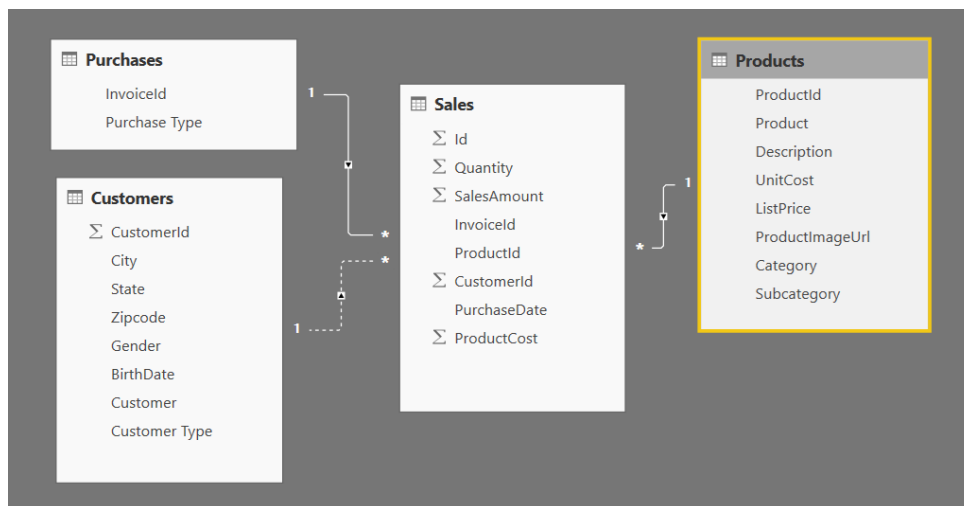
11. Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Purchases** query. Power BI Desktop will display the **Apply Query Changes** dialog while importing the data and transforming it to load it into the data model.
12. After the query changes have been applied, you should be able to see the results of your changes in the **Purchase** table that has been loaded into the project's data model.



13. Navigate to relationship 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 **Purchases** table. This is due to the fact that you removed the **CustomerId** column from the **Purchases** table.



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



15. Modify the relationship between the **Customers** table and the **Sales** table.

- Notice the relationship between Customers and sales has a dotted line instead of a solid line. This means the relationship has been created but it has not been configured as an active relationship.
- Double-click the dotted line between the **Customers** table and the **Sales** table to edit the relationship.
- In the **Edit Relationship** dialog, click the **Make this relationship active** checkbox so it is checked and then click **OK**.

Edit Relationship

Select tables and columns that relate to one another.

Sales

Id	Quantity	SalesAmount	InvoiceId	ProductId	CustomerId	PurchaseDate	ProductCost
95	9	179.55	46	6	46	Saturday, February 4, 2012	\$128..
96	9	179.55	47	6	47	Saturday, February 4, 2012	\$128..
307	9	179.55	155	6	142	Thursday, February 23, 2012	\$128..

Customers

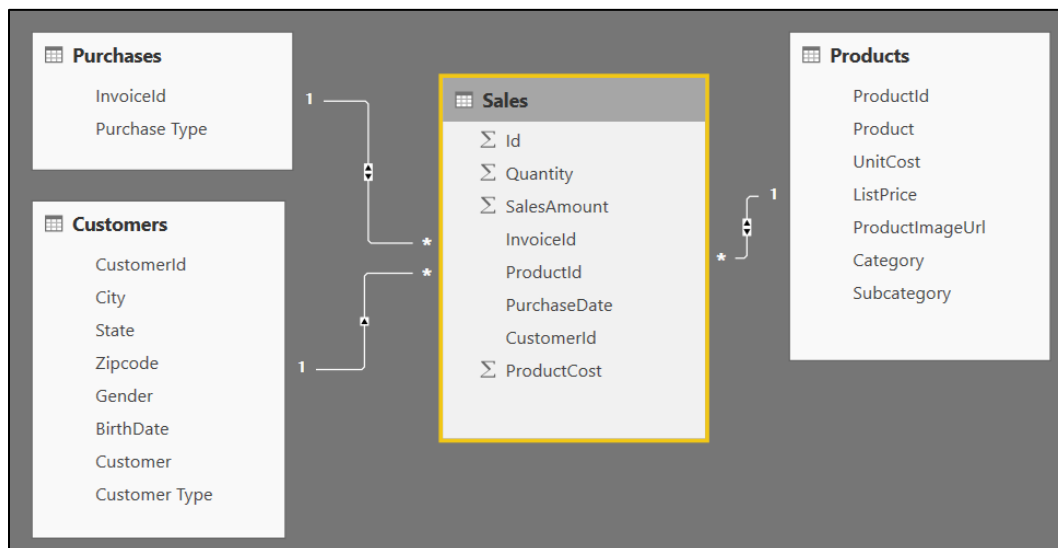
CustomerId	City	State	Zipcode	Gender	BirthDate	Customer	Customer Type
760	San Jose	CA	95133	Female	Saturday, March 16, 1968	Lucile Blake	One-time Customer
883	San Jose	CA	95133	Female	Sunday, July 15, 1942	Rochelle Owen	One-time Customer
940	San Jose	CA	95133	Female	Sunday, March 2, 1943	Corinne Finch	One-time Customer

Cardinality: Many to One (*:1) Cross filter direction: Single

☒ Make this relationship active
☐ Assume Referential Integrity

OK Cancel

- d) You should see a solid line between the **Customers** table and the **Sales** tables indicating that the relationship between these two tables is now an active relationship.



The tables in the data model have now been refactored into a **star schema** that is commonly used in data modeling for BI projects.

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

You are now done with your work modifying queries to transform data as it is being loaded into the data model of your project. Note that you will continue to work on the PBIX file with this project in the next lab and the ones that follow.