## Using the Power Query Features of Power BI Desktop

**Lab Time**: 40 minutes

**Lab Folder**: C:\Student\Modules\PowerQuery\Lab\

**Lab Overview**: In this lab you will work with Power Query to extract data from a SQL Azure database and transform the data as it is loaded into the data model of a Power BI Desktop project.

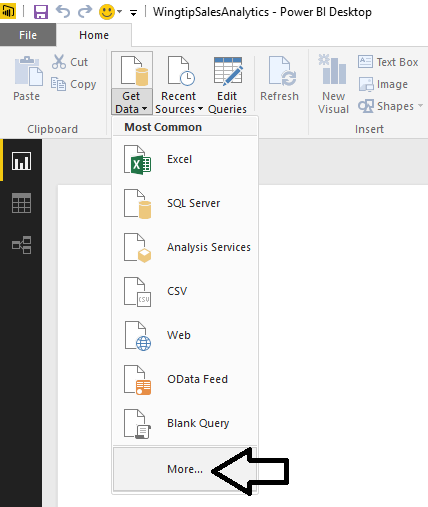
### Exercise 1: Importing Data from a SQL Azure Database

In this exercise you will connect to a SQL Azure database and import data into Power BI Desktop using its Power Query features.

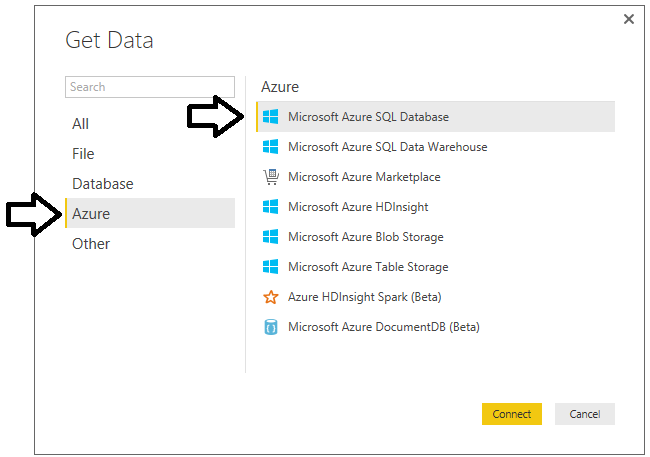
1. Launch Power BI Desktop to start a new project.
2. Save the new project as **WingtipSalesAnalytics.pbix** using the following path.

C:\Student\Projects\WingtipSalesAnalytics.pbix

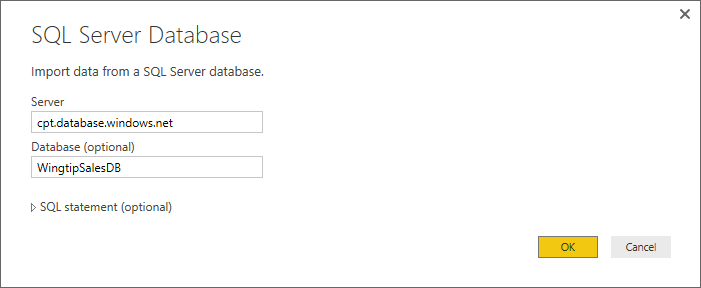
1. Drop down the **Get Data** menu button on the ribbon and click **More…**.



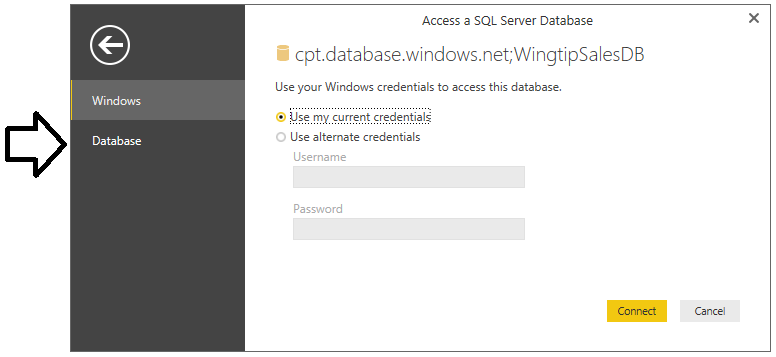
1. On the **Get Data** dialog, select Azure in the list on the left. After that select **Microsoft SQL Azure Database** on the right and then click the **Connect** button.



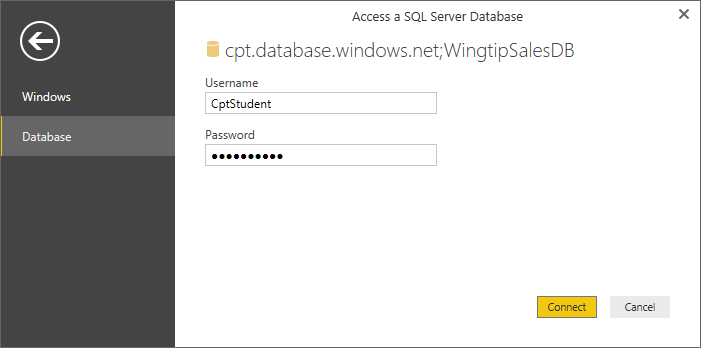
1. When you are prompted with the **SQL Server Database** dialog, enter the following values for the **Server** and **Database**.
   1. Server: **cpt.database.windows.net**
   2. Database: **WingtipSalesDB**
2. When the **SQL Server Database** dialog appears as it does in the following screenshot, click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



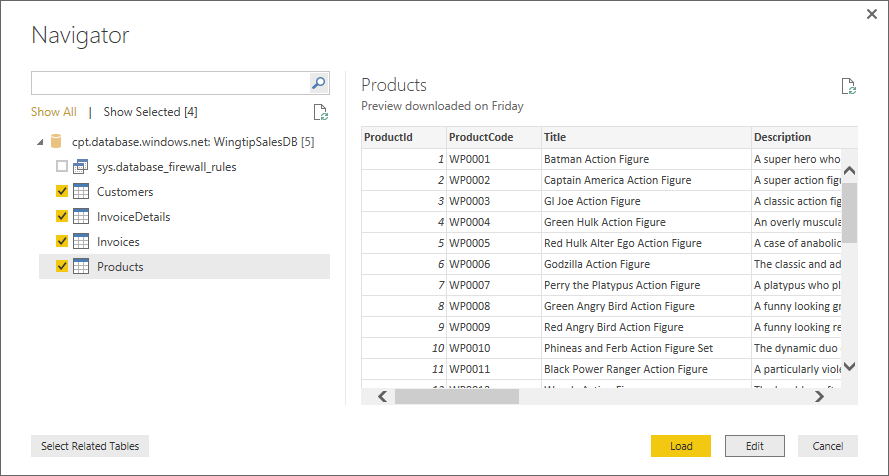
1. At this point, you will be prompted by the **Access a SQL Server Database** dialog. Click on **Database** on the left side of the dialog so that you can enter the credentials for a standard SQL account instead of using Windows authentication.



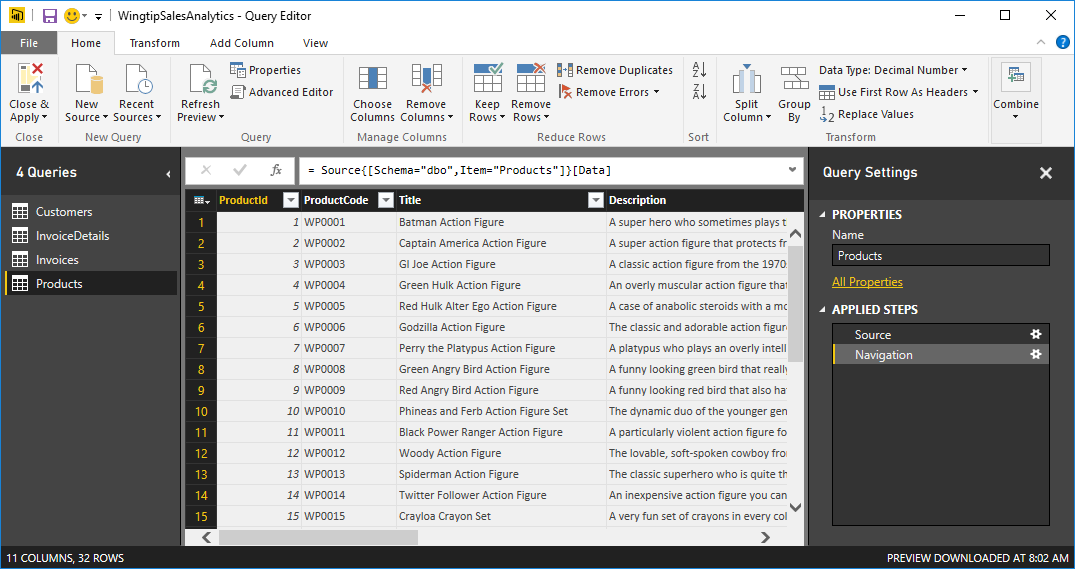
1. Enter the following credentials for a SQL user account that has been configured with read access to the database.
   1. Username: **CptStudent**
   2. Password: **pass@word1**
2. Once you have entered the credentials the **Access a SQL Server Database** dialog, click the **Connect** button to continue.



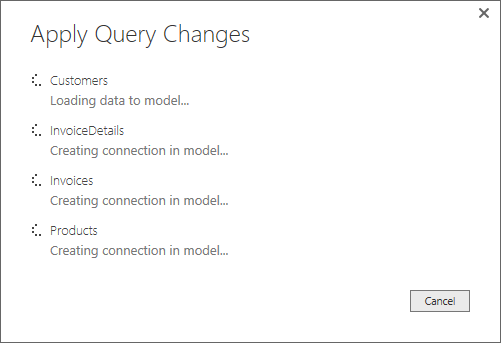
1. At this point, Power BI Desktop should be able to establish a connection to the database and then prompt you with the **Navigator** dialog so you can select the tables you would like to import into your PBIX project. In the Navigator dialog, select the **Customers** table, the **InvoiceDetails** table,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



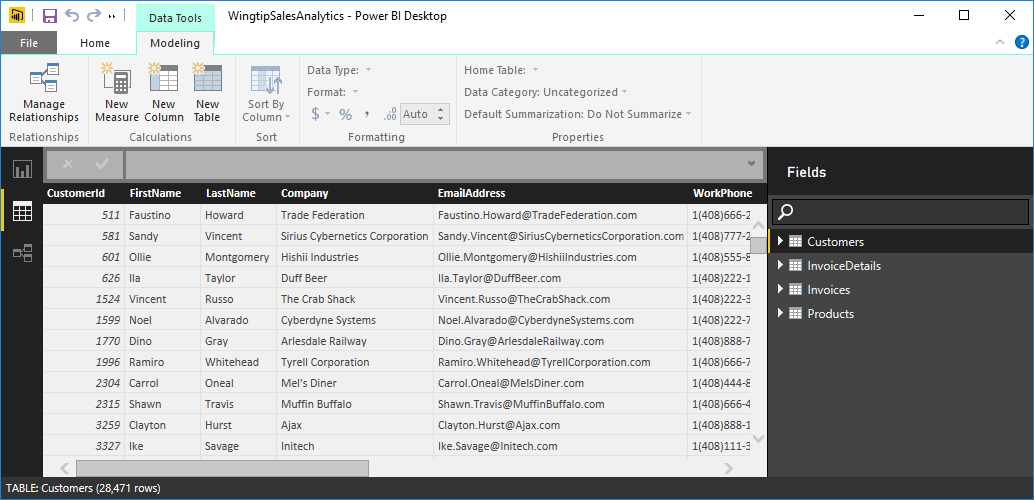
1. When you inspect the Query Window, you should now be able to observe that Power BI Desktop has created a new query for each of the four tables that you selected in the **Navigator** dialog. You will not begin to edit these queries until the next exercise. However, now you should take a moment inspect 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 the data just as it is defined in the underlying table.



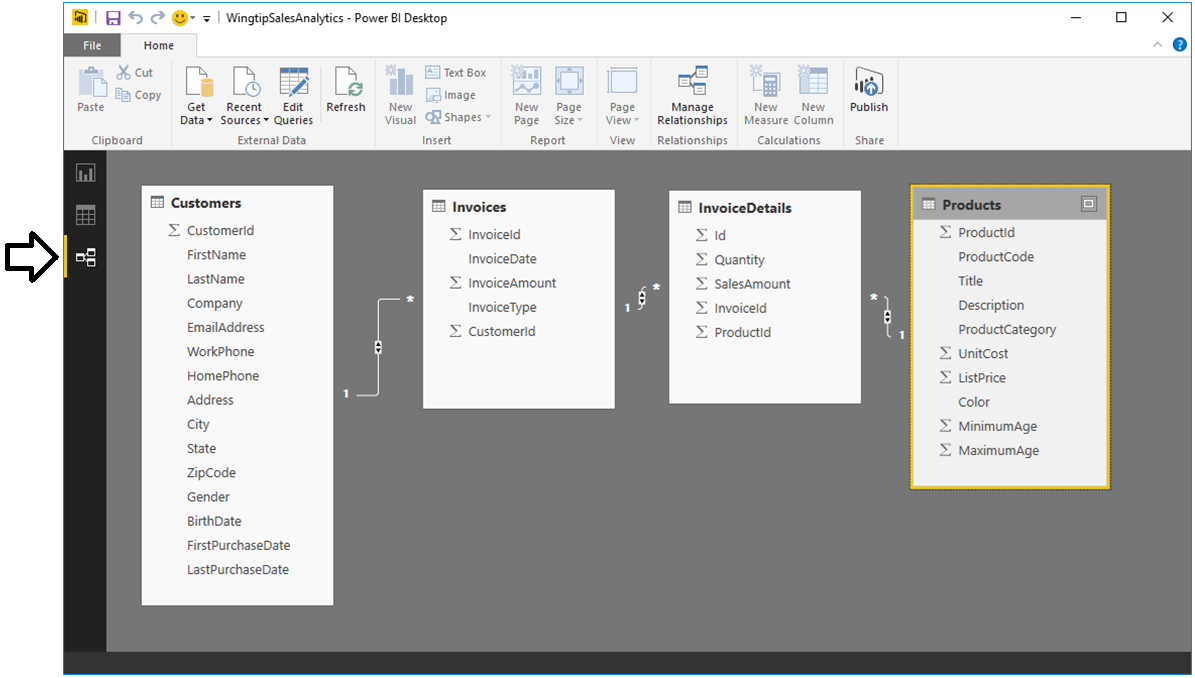
1. Click the **Close and Apply** button in the Query Editor window. This will close the Query Editor window and start the process of extracting the data from the SQL Azure database and importing it into the current project's data model using a local cache on your student workstation. The **Apply Query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



1. Once the data from all four queries has been imported, 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.



1. 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.

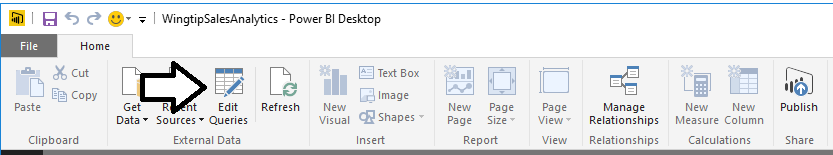


1. Save your work in the current project by clicking the Save button in the upper left corner of the Power BI Desktop window.

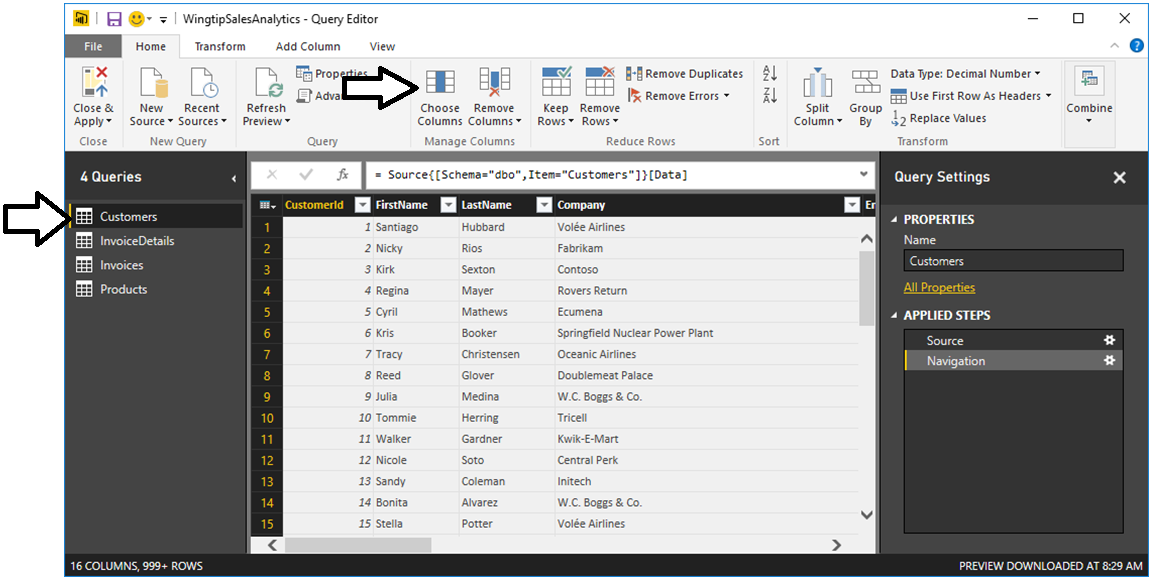
### Exercise 2: Using Power Query to Transform and Reshape Customer Data

In the following exercise, you will use the Query Editor to modify the Customers query to transform customer data before it is loaded into the data model.

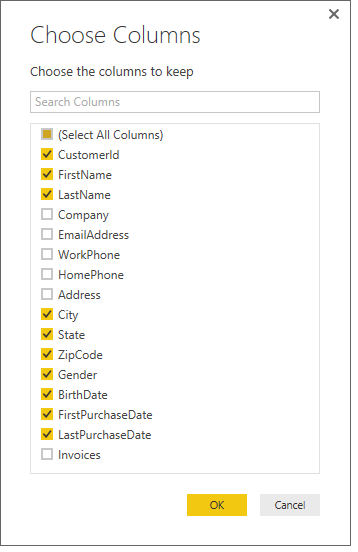
1. Make sure you have the **WingtipSalesAnalytics.pbix** project open that you started in the previous exercise.
2. Click on the **Edit Queries** button in the ribbon to display the Query Editor window.



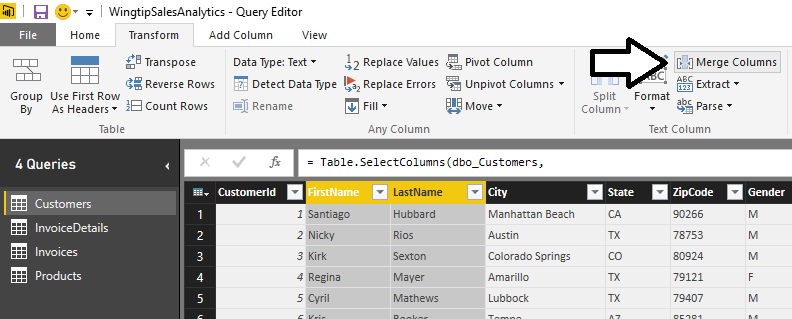
1. Make sure the **Customers** query is selected in the **Queries** list on the left-hand side of the Query Editor window. Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.



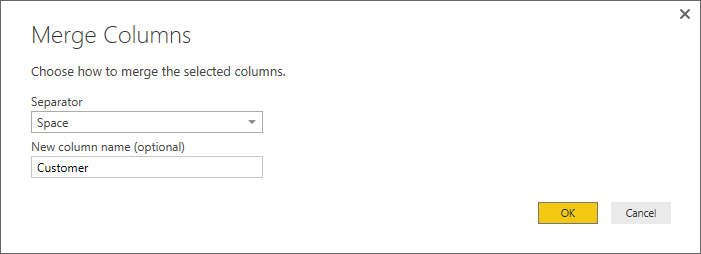
1. In the **Choose Columns** dialog, begin by clicking on the (Select all Columns) checkbox at the top to unselect all column. Next, select the checkboxes for **CustomerId**, **FirstName**, **LastName**, **City**, **State**, **ZipCode**, **Gender**, **BirthDate**, **FirstPurchaseDate** and **LastPurchaseDate**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



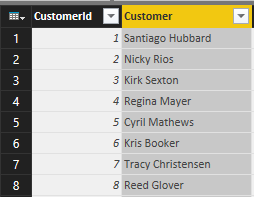
1. You should be able to see that the Query Editor window now only shows the columns that you selected.
2. In this step you will merge the **FirstName** column and the **LastName** column together into a single column named **Customer**.
   1. Begin by click on the **Transform** tab in the ribbon.
 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



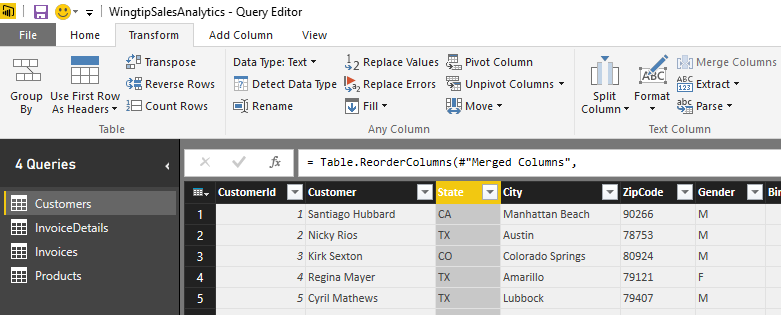
* 1. In the **Merge Column** dialog, drop down the **Separator** control and select a value of **Space**. Add a **New column name** value of **Customer** and click the **OK** button to modify the underlying query with your changes.



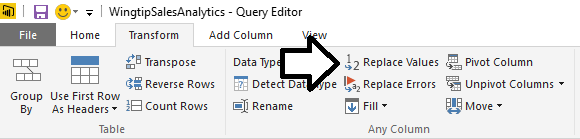
* 1. You should now be able to see that the **FirstName** column and the **LastName** column have been replaced with a single merged column named **Customer**.



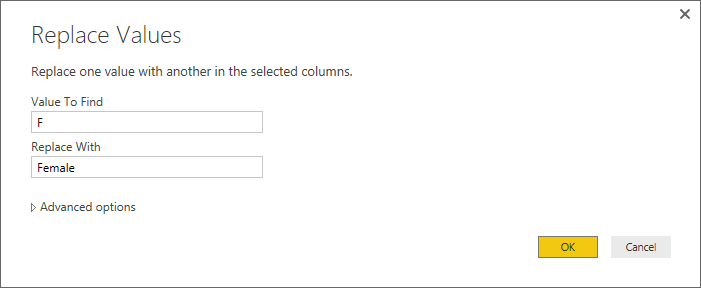
1. The **State** column is currently to the right of the **City** column. Move the **State** column so that it is repositioned to the left of the **City** column. Accomplish this by clicking on the column header for the **State** column and leaving the mouse button down. Move the **State** column to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



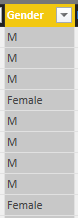
1. Modify the query so that the **Gender** column returns values of **Male** and **Female** instead of **M** and **F**.
   1. Make sure the **Transform** tab is the active tab in the ribbon.
   2. Select the **Gender** column by clicking its column header.
   3. Click the **Replace Values** button in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



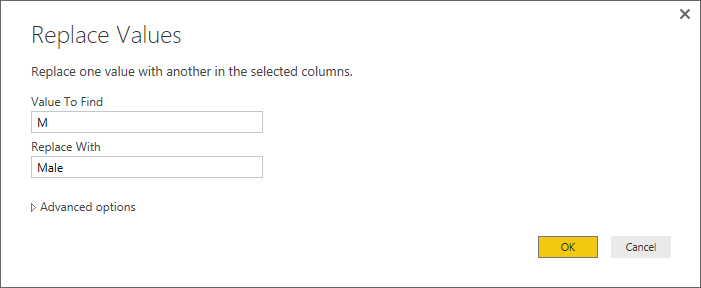
* 1. In the **Replace Value** dialog, enter a value of **F** in the **Value to Find** textbox and enter a value of **Female** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.



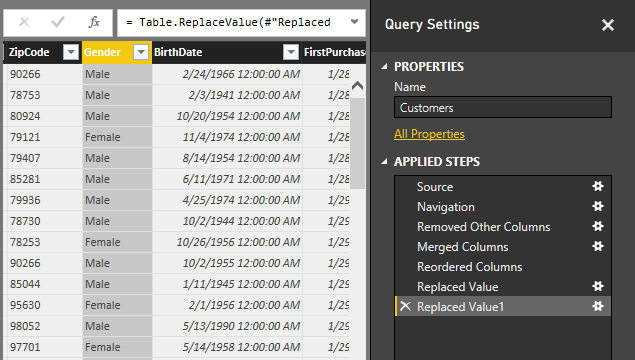
* 1. You should be able to see that all values of **F** in the **Gender** column have been replaced with a value of **Female**.



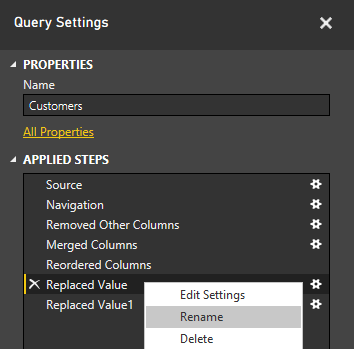
* 1. Make sure the **Gender** column is still selected.
  2. Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.
  3. In the **Replace Value** dialog, enter a value of **M** in the **Value to Find** textbox and enter a value of **Male** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.



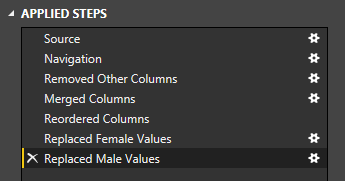
* 1. You should be able to confirm that all values in the **Gender** column have been replaced with a value of either **Male** or **Female**. If you inspect the **APPLIED STEPS** list in the **Query Settings** pane, you should be able to see that there are two steps at the end that have been given the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



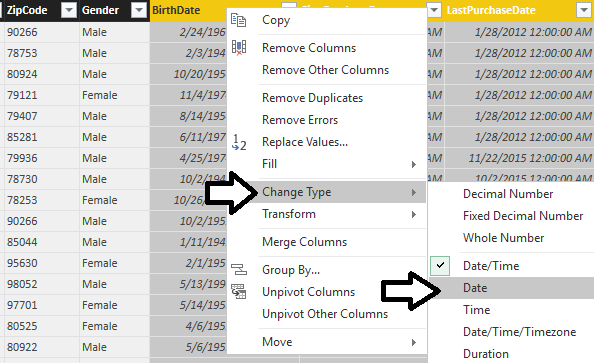
* 1. In order to promote higher levels of maintainability, it's often a good idea to rename steps with names such as of **Replaced Value** and **Replaced Value 1**.Rename the **Replaced Values** step by right-clicking it and clicking the **Rename** command to place the step name in edit mode. Modify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



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



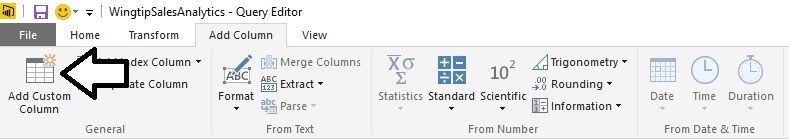
1. Change the column type of **BirthDate**, **FirstPurchasedDate** and **LastPurchasedDate** from **Date/Time** to **Date**.
   1. Select the **BirthDate** column by clicking its column header.
   2. Hold down the **SHIFT** key and click the column headers for **FirstPurchasedDate** and **LastPurchasedDate** so that all three columns are selected.
   3. Right-click on any one of the selected columns and select the **Change Type > Date** command.



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



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



* 1. In the **Add Custom Column** dialog, add a value of **CustomerType** in the **New column name** textbox.

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

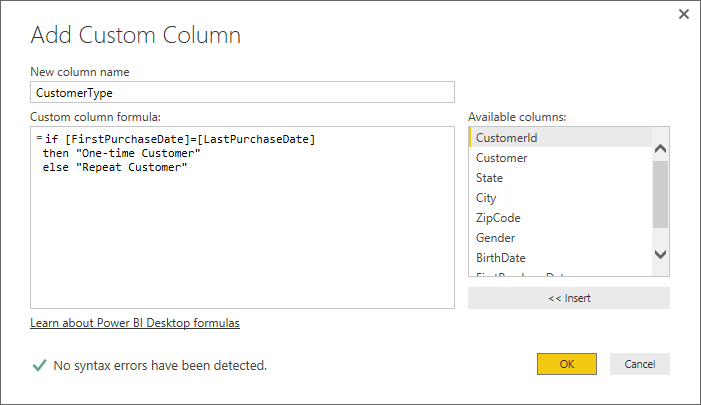
* 1. In the **Custom column formula** textbox, enter the following formula.

if [LastPurchaseDate]=[FirstPurchaseDate]

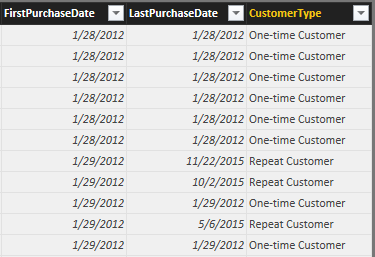
then "One-time Customer"

else "Repeat Customer"

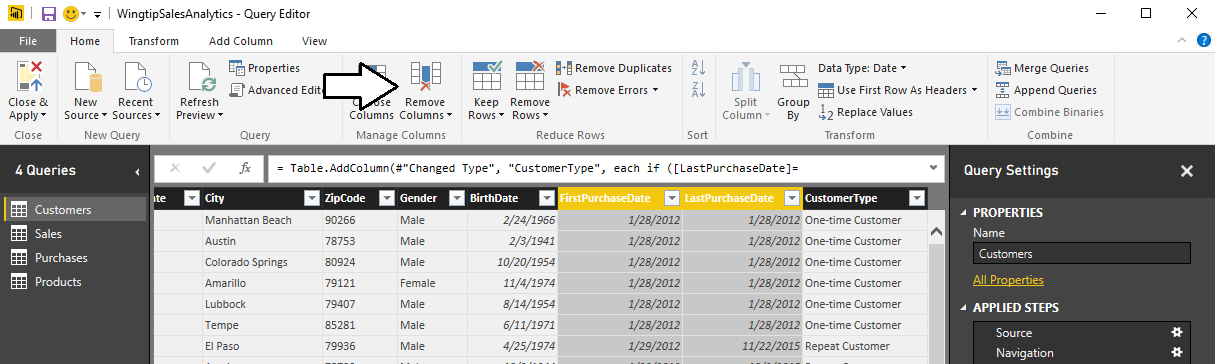
* 1. When the **Add Custom Column** dialog appears as the following screenshot, click the **OK** button to add the new column.



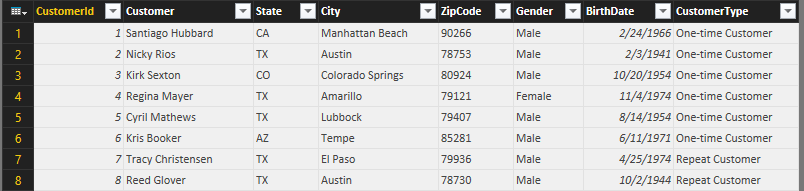
* 1. You should be able to verify that the new **CustomerType** column has a value of **Repeat Customer** when the current customer has a **FirstPurchaseDate** column value that is not equal to the **LastPurchaseDate** column value. When these column values are equal, the **CustomerType** column has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



1. Now, that you have used the **FirstPurchaseDate** column and the **LastPurchaseDate** column to calculate the value of the **CustomerType** column, you can delete them because they are now longer needed.
   1. Activate the **Home** tab on the ribbon.
   2. Select the **FirstPurchaseDate** column by clicking its column header.
   3. Hold down the **SHIFT** key and click the column header for **LastPurchaseDate** so that both columns are selected.
   4. Click the **Remove Columns** button in the ribbon to remove both columns from the query results.

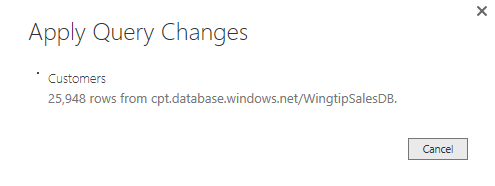


* 1. You should be able to confirm that the **FirstPurchaseDate** column and the **LastPurchaseDate** columns have been removed from the query results. However, the **Repeat Customer** column is still there.

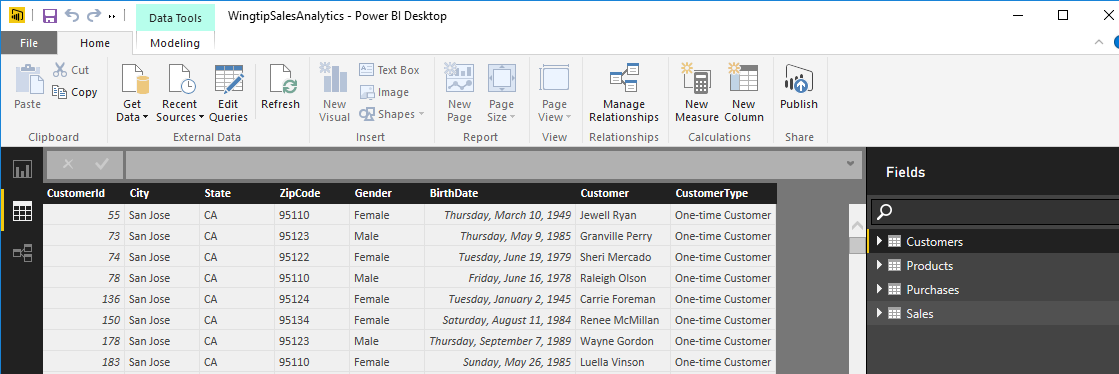


You are now done working with the **Customers** query.

1. 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 and transforming it to load it into the data model.



1. 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.

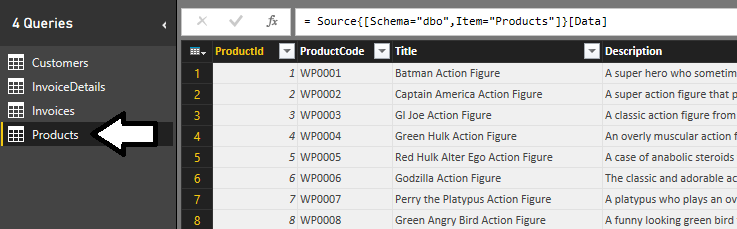


1. Save the work you have done to **WingtipSalesAnalytics.pbix** by clicking the Save button in the upper left corner of the Power BI Desktop window.

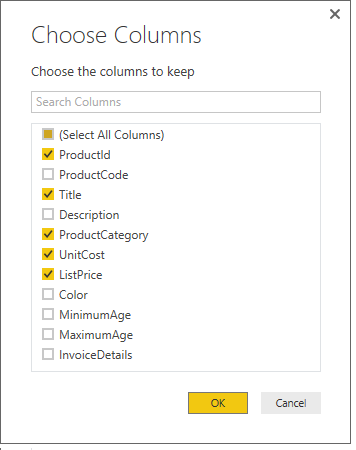
### Exercise 3: Using Power Query to Transform and Reshape Product Data

In the following exercise, you will use the Query Editor to modify the Products query to transform customer data before it is loaded into the data model.

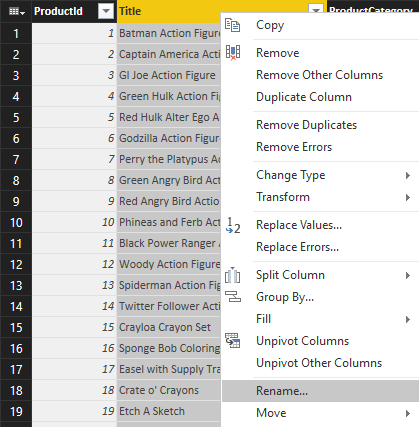
1. Make sure you have the **WingtipSalesAnalytics.pbix** project open that you started in the previous exercise.
2. Click on the **Edit Queries** button in the ribbon to display the Query Editor window.
3. Make sure the **Products** query is selected in the **Queries** list on the left-hand side of the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



1. Remove the columns that are not required in the **Products** query results.
   1. Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
   2. 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 **ProductId**, **Title**, **ProductCategory**, **UnitCost** and **ListPrice** as shown in the following screenshot. Once you have these columns selected, click the **OK** button to close the dialog and to modify the underlying query.



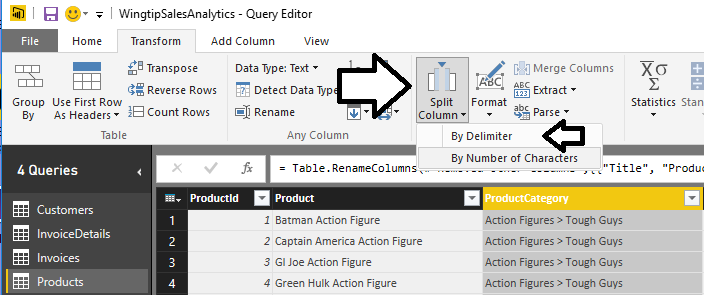
1. Rename the **Title** column to **Product**.
   1. Right-click on the **Title** column and click **Rename**.



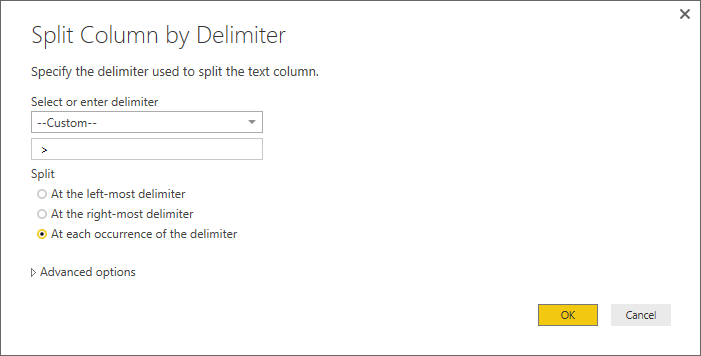
* 1. Modify the column name to **Product**.



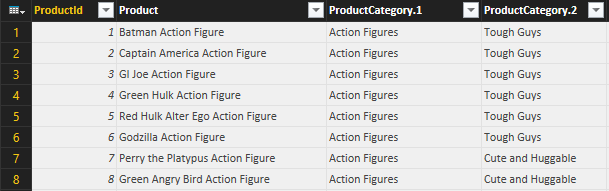
1. Split the **ProductCategory** column up into two separate columns named **Category** and **Subcategory**.
   1. Activate the **Transform** tab on the ribbon.
   2. Select the **ProductCategory** by clicking its column header.
   3. Drop down the **Split Column** menu button and click by **By Delimiter** to display the **Split Column By Column** dialog.



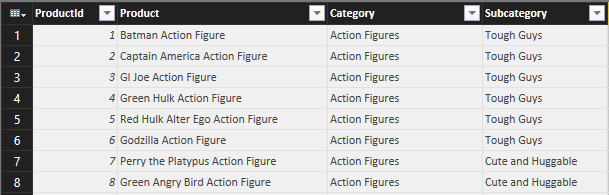
* 1. In the **Split Column By Column** dialog, drop down the Select or enter delimiter combo box and select a value of --Custom--.
  2. In the textbox below the combo box, enter a three character text value which includes a space follow by the > character followed by another space.
  3. When the **Split Column By Column** dialog appears as the one in the following screenshot, click the OK button to add the step to the **Products** query.



* 1. 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**.

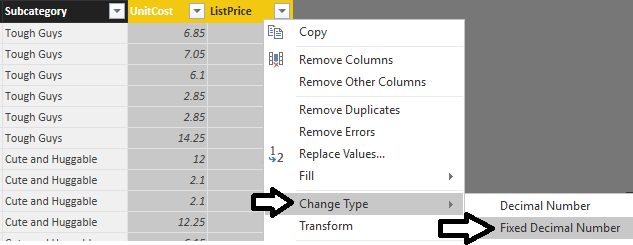


* 1. Rename the **ProductCategory.1** column to **Category.** Also rename the **ProductCategory.2** column to **Subcategory** as shown in the following screenshot.



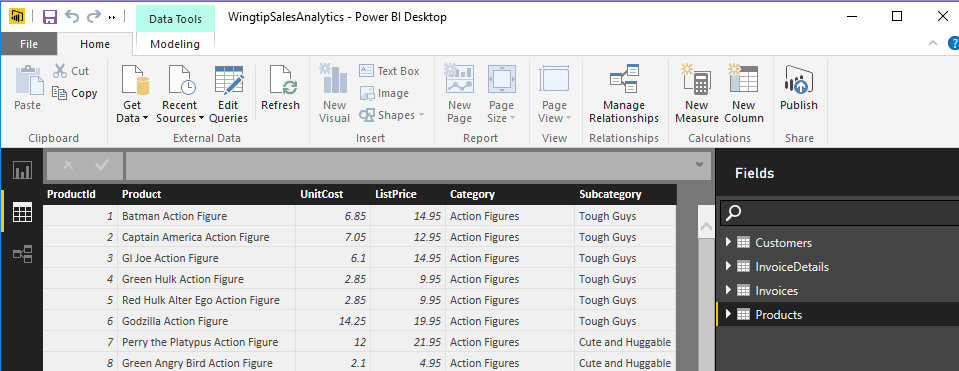
When you have query columns based on numeric currency values, it is best to change their column type to **Fixed Decimal Number**.

1. Modify the column type of the **UnitCost** column and the **ListPrice** column to the **Fixed Decimal Number** type.
   1. Select the **UnitCost** column by clicking its column header.
   2. Hold down the **SHIFT** key and click the ListPrice column so that both columns are selected.
   3. Right-click either one of the selected columns and click the **Change Type > Fixed Decimal Number** menu command.



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

1. 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.
2. 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.



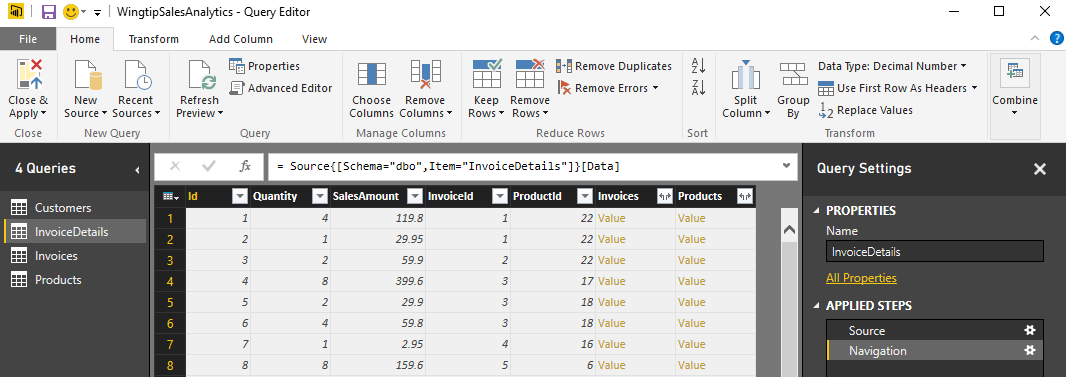
You might have noticed that the columns are displayed in the **Products** table do not appear in the same order as they do in the **Products** query. Hopefully, future updates to Power BI Desktop should correct this column ordering issue.

1. 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 Data

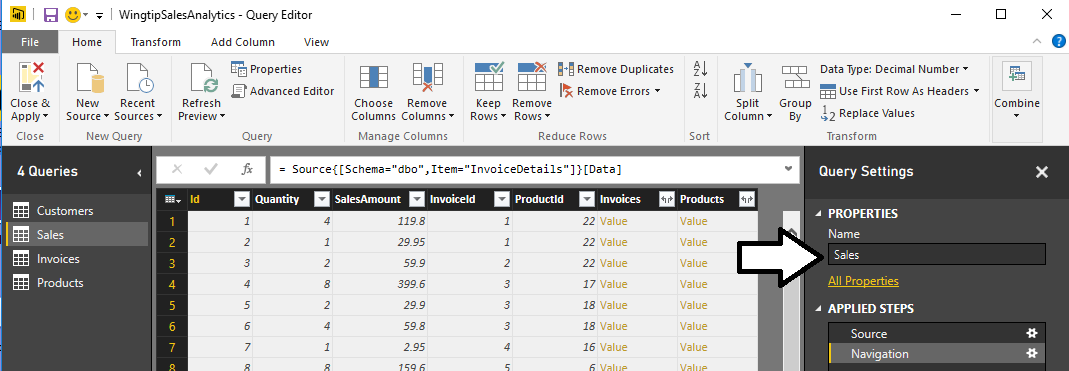
In the following exercise, you will use the Query Editor to modify the **InvoiceDetails** query to transform data related to sales data before it is loaded into the data model.

1. Make sure you have the **WingtipSalesAnalytics.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.

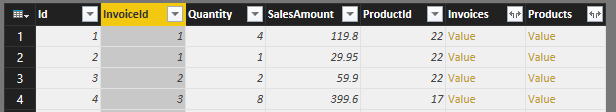


One important point of flexibility in the import process is that you can change the name of a query, and therefore the name of the resulting table to make the data model more intuitive business users. The **InvoiceDetails** query is returning data that will be used to calculate sales results at the most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

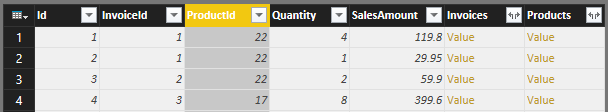
1. Update the name of the **InvoiceDetails** query to **Sales** by replacing the text in the **Name** textbox in the **Query Settings** pane.



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



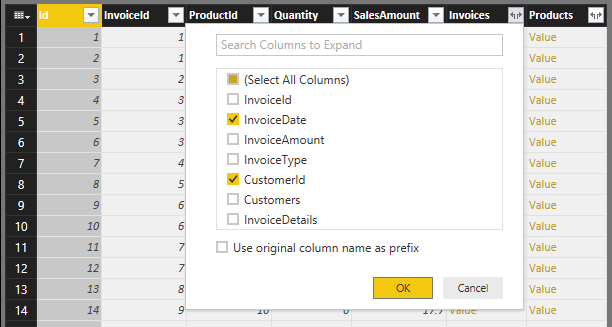
1. Using the mouse, drag and drop the column header for the **ProductId** column to move it to the right of the **InvoiceId** column.



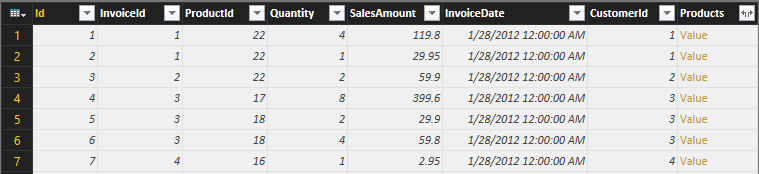
1. Modify the column type of the **SalesAmount** column to the **Fixed Decimal Number** type.
   1. Select the **SalesAmoungt** column by clicking its column header.
   2. Right-click the selected column and click the **Change Type > Fixed Decimal Number** menu command.
2. Expand the **Invoices** column to add the **InvoiceDate** column and the **CustomerId** column to the **Sales** query.
   1. Click the Expand button inside the column header of the **Invoices** column to display the **Columns to Expand** dialog.



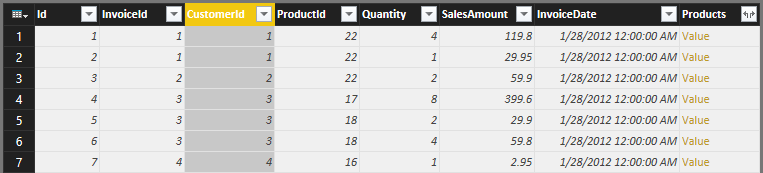
* 1. In the **Columns to Expand** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkboxes for the **InvoiceDate** column and the **CustomerId** column. Also make sure to uncheck the checkbox with the caption **Use original column 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.



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



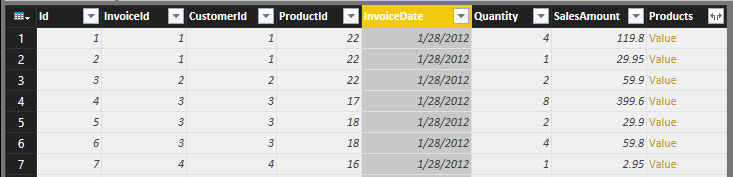
1. Use the mouse to drag and drop the **CustomerId** column to move it to the right of the **InvoiceId** column.



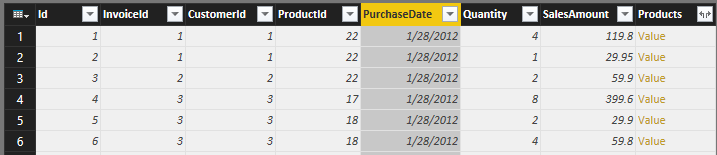
1. Use the mouse to drag and drop the **InvoiceDate** column to move it to the right of the **ProductId** column.



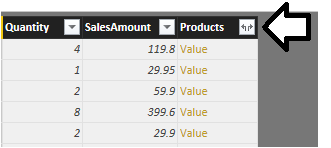
1. Change the column type of the **InvoiceDate** to the **Date** type by right-clicking its column header and selecting the **Change Type > Date** command. When you are done, the data in the column should show a date value without a time.



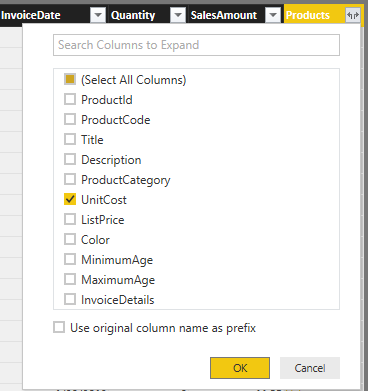
1. Change the name of the **InvoiceDate** column to **PurchaseDate**.



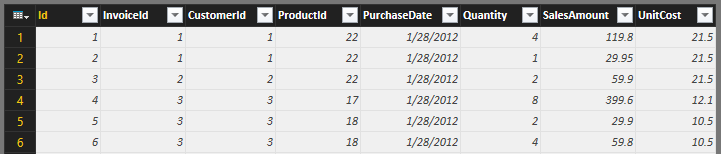
1. Expand the **Products** column to add the **UnitCost** column to the **Sales** query.
   1. Click the Expand button inside the column header of the **Products** column to display the **Columns to Expand** dialog.



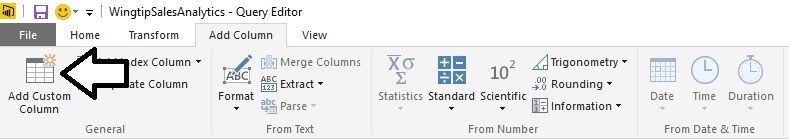
* 1. In the **Columns to Expand** dialog, begin by clicking on the **(Select all Columns)** checkbox at the top to unselect all columns. Next, select the checkbox for the **UnitCost** column. Also make sure to uncheck the checkbox with the caption **Use original column name as prefix**. Once the **Columns 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.



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



1. Add a new custom column named **ProductCost** to calculate the product of the **Quantity** field multiplied by the **UnitCost** field.
   1. Begin by activating the **Add Column** tab in the ribbon.
   2. Click the **Add Custom Column** button in the ribbon to display the **Add Custom Column** dialog.



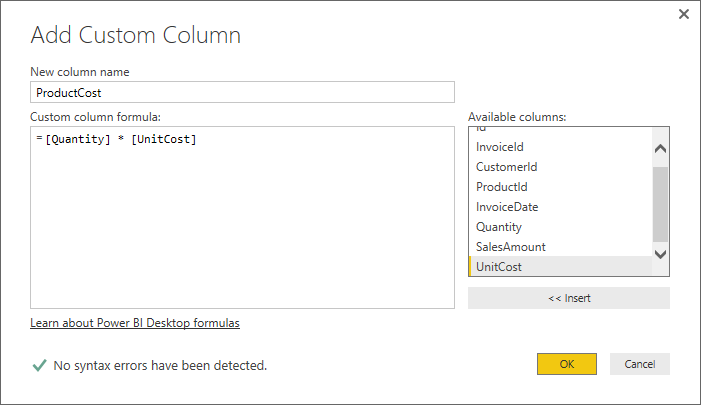
* 1. In the **Add Custom Column** dialog, add a value of **ProductCost** in the **New column name** textbox.

In this particular scenario, you are working under the assumption that the customer is a repeat customer when the **FirstPurchaseDate** column and the **LastPurchaseDate** column have value that are not equal.

* 1. In the **Custom column formula** textbox, enter the following formula.

[Quantity] \* [UnitCost]

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

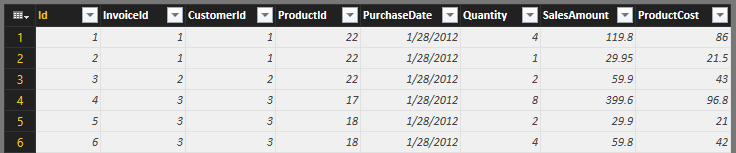


* 1. You should be able to verify that the new **ProductCost** column has a value calculated by multiply the value of the **Quantity** column together with the value of the **UnitCost** value.



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.

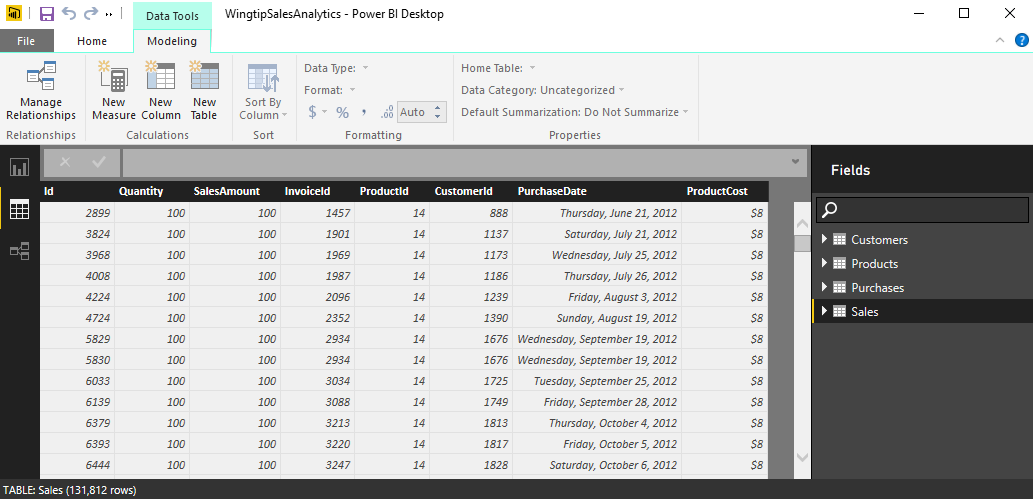
1. Remove the **UnitCost** column by selecting its column header and pressing the DELETE key on the keyboard.



1. Modify the column type of the **SalesAmount** column and the **ProductCost** column to the **Fixed Decimal Number** type.
   1. Select the **SalesAmount** column by clicking its column header.
   2. Hold down the **SHIFT** key and click the **ProductCost** column so that both columns are selected.
   3. Right-click either one of the selected columns and click the **Change Type > Fixed Decimal Number** menu command.

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

1. 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.
2. 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.



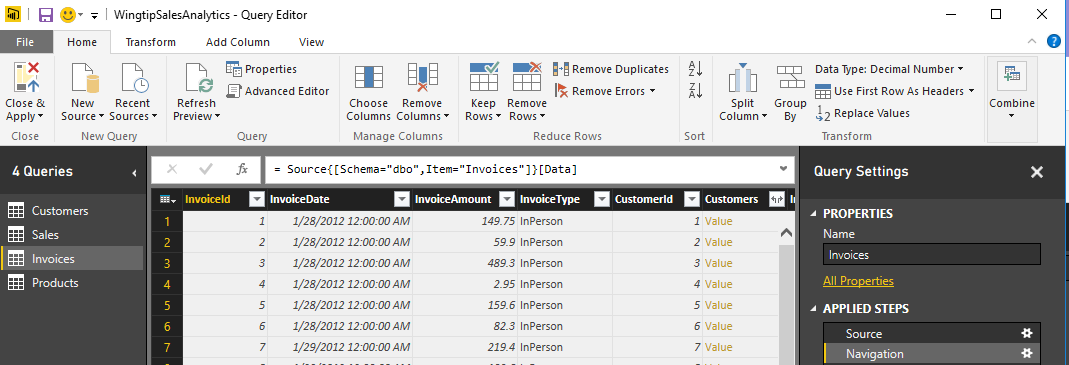
Unfortunately, the current version of Power BI Desktop does not retain the order of the columns that you configured in the **Sales** query.

1. 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 Purchase Data

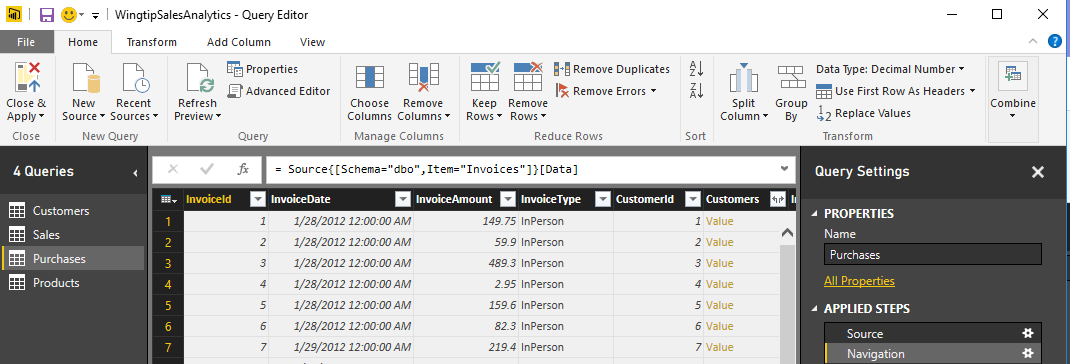
In the following exercise, you will use the Query Editor to modify the **Invoices** query to transform data related to purchase data before it is loaded into the data model.

1. Make sure you have the **WingtipSalesAnalytics.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.

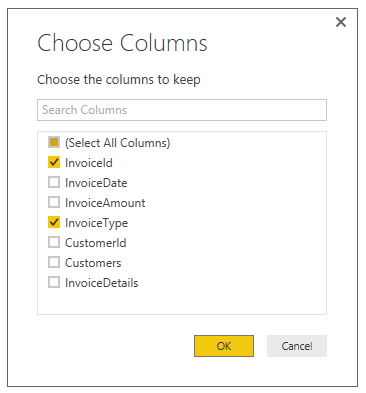


In the last exercise you changed the name of the **InvoiceDetails** to **Sales** to make the data model easier to understand. In this exercise you will change the name of the **Invoices** table to **Purchases** for the same reason.

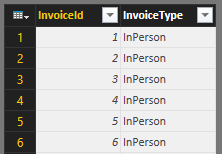
1. Update the name of the **Invoices** query to **Purchases** by replacing the text in the **Name** textbox in the **Query Settings** pane.



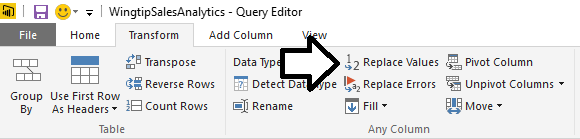
1. Make sure the **Purchases** query is selected in the **Queries** list on the left-hand side of the Query Editor window. Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
2. In the **Choose Columns** dialog, begin by clicking on the (Select all Columns) checkbox at the top to unselect all column. Next, select the checkboxes for **InvoiceId** and **InvoiceType** as shown in the following screenshot. Once you have these columns selected, click the **OK** button to close the **Choose Columns** dialog and to modify the underlying query.



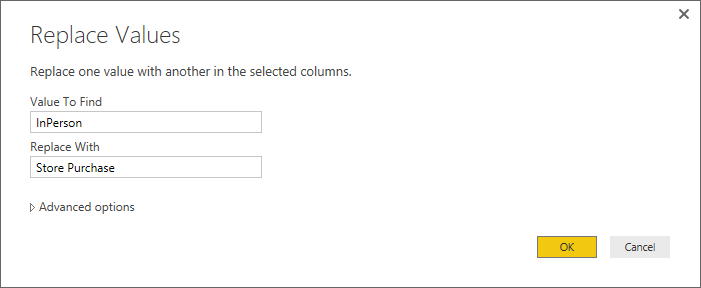
1. You should be able to see that the Query Editor window now only shows the columns that you selected.



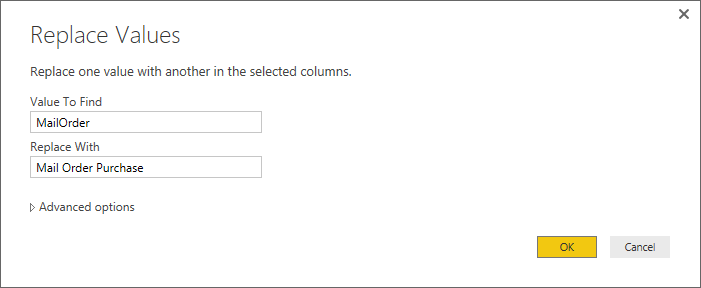
1. Modify the query so that the **InvoiceType** column returns values that are more human readable.
   1. Make sure the **Transform** tab is the active tab in the ribbon.
   2. Select the **InvoiceType** column by clicking its column header.
   3. Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.



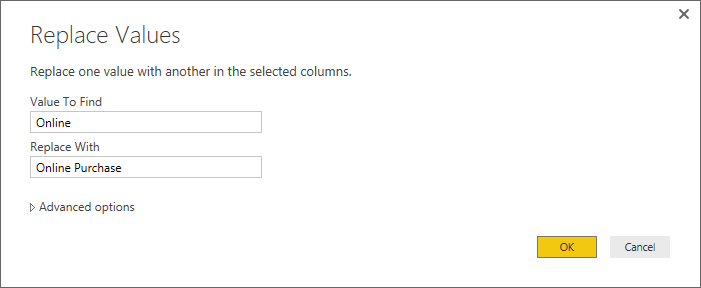
* 1. In the **Replace Value** dialog, enter a value of **InPerson** in the **Value to Find** textbox and enter a value of **Store Purchase** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.



* 1. Make sure the **InvoiceType** column is still selected.
  2. Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.
  3. 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.

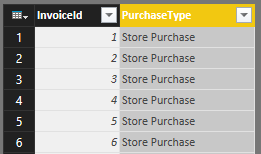


* 1. Make sure the **InvoiceType** column is still selected.
  2. Click the **Replace Values** button in the ribbon to display the **Replace Values** dialog.
  3. In the **Replace Value** dialog, enter a value of **Online** in the **Value to Find** textbox and enter a value of **Online Purchase** in the **Replace With** textbox. Click to **OK** button add your changes to the underlying query.



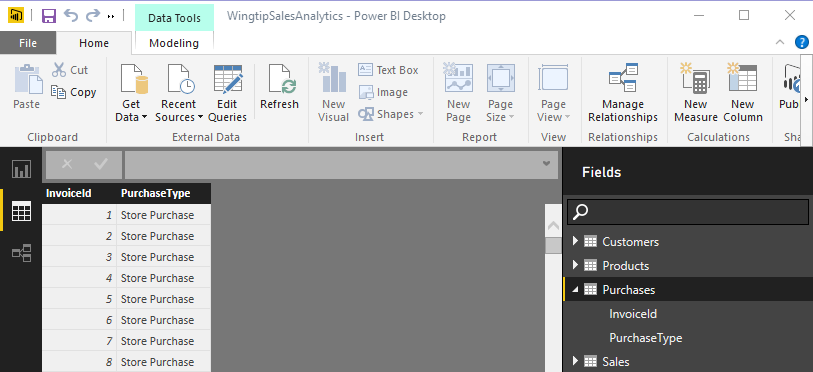
* 1. If you scroll down and look at rows within the **Purchases** table, you should be able to see that each row has a **InvoiceType** column value of either **Store Purchase**, **Mail Order Purchase** or **Online Purchase**.

1. Change the name of the **InvoiceType** column to **PurchaseType**.

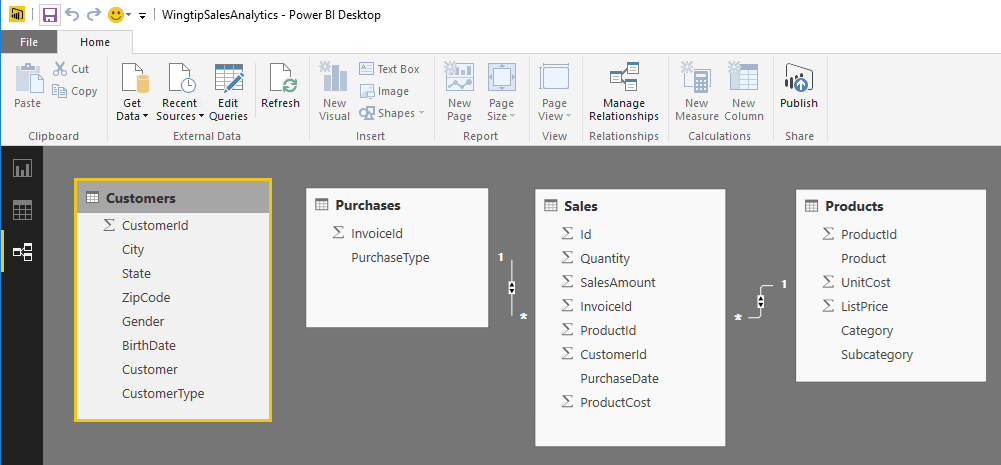


You are now done working with the **Purchase** query.

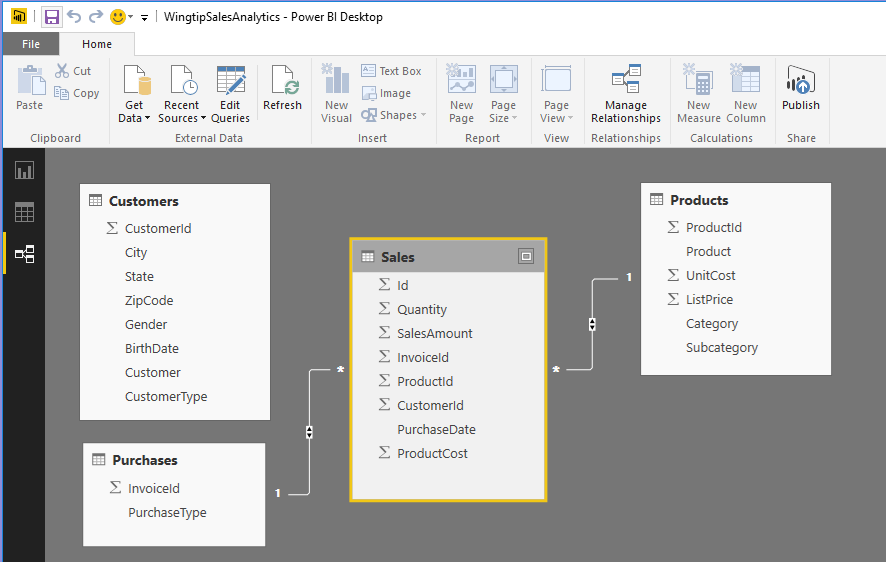
1. Click the **Close and Apply** button on the **Home** tab of the ribbon to execute the updated **Purchases** query. Power BI Desktop will display the **Apply Query Changes** dialog while importing the data and transforming it to load it into the data model.
2. After the query changes have been applied, 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.



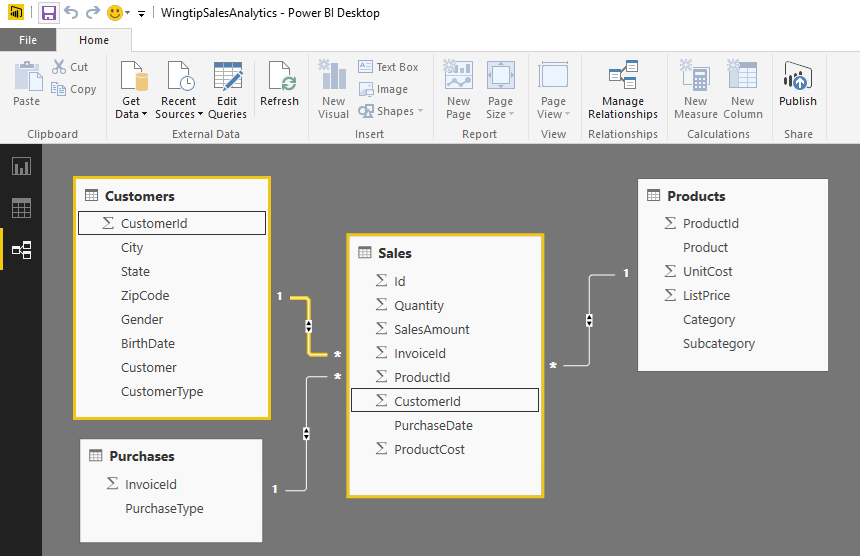
1. Navigate to relationship view to see how your updates to the queries in this project have affected the relationships between the tables. You should be able to see that there is no longer a relationship between the **Customers** table and the **Purchases** table. This is due to the fact that you removed the **CustomerId** column from the **Purchases** table.



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



1. Create a new relationship between the **Sales** table and the **Customers** table by clicking on the **CustomerId** column in the **Sales** table and then dragging and dropping it on the **CustomerId** column of the **Customers** table. Once you have successfully completed this step, you should see a visual line between these two tables to indicate that the data model now contains the new 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.

1. You are now done with your work modify all queries in the project. Save your work by clicking the Save button in the upper left corner of the Power BI Desktop window. Note that you will continue to work on this project in the lab exercises at the end of the next module of this course.