

Getting Started with Power BI Desktop

Lab Time: 45-60 minutes

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

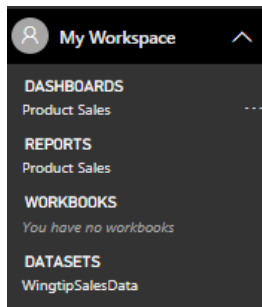
Lab Overview: In this module you will learn how to install and begin working with Power BI Desktop. You will learn how to import data using Power Query to create a dataset and you will also learn how model the data within the dataset using Power Pivot. After modeling the data to make it better suited for analysis, you will create a report and publish it to the Power BI service.

Lab Dependencies: This lab assume that you have already completed **Lab01: Getting Up and Running with the Power BI Service**. Therefore, you should have already created an Office 365 trial account that allows you to log into the Power BI service and you should have already created a dataset, a report and a dashboard in your personal workspace.

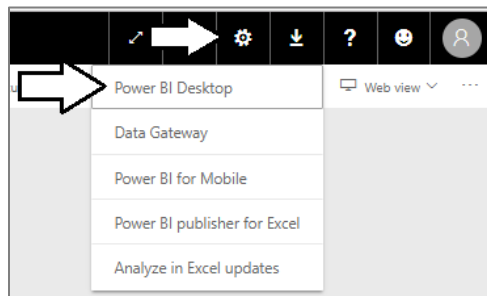
Exercise 1: Installing and Configuring Power BI Desktop

In this exercise you will download and install Power BI Desktop. Note that if Power BI desktop is already installed on your student workstation, you can skip ahead to exercise 2.

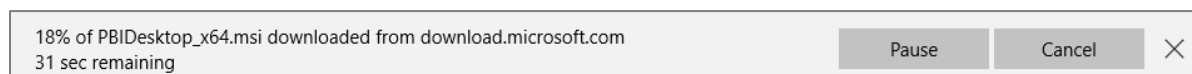
1. Launch the browser and navigate to the landing page of the Power BI service at <https://app.powerbi.com>.
 - a) Make sure you log on to the Power BI service using your student account.
 - b) Inside your personal workspace, you should see the dataset, report and dashboard you created in the previous lab.



2. Install Power BI Desktop
 - a) On the top right of the Power BI service window, drop down the **Downloads** menu and click the **Power BI Desktop** menu command to begin the download of the installation file.



- b) Wait for the MSI file to download.



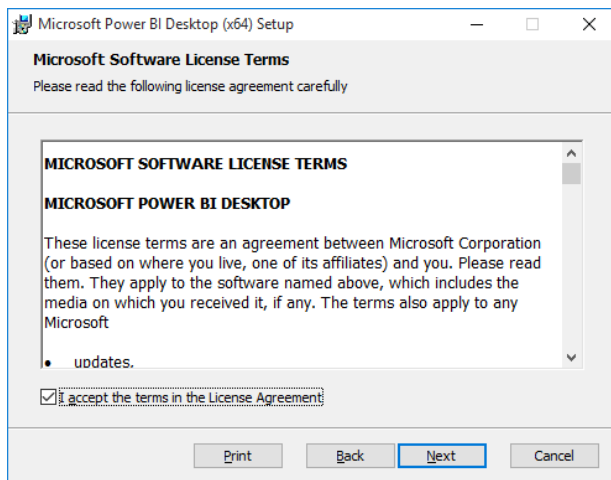
- c) Once the file has downloaded, click the **Run** button to begin the installation of Power BI Desktop.



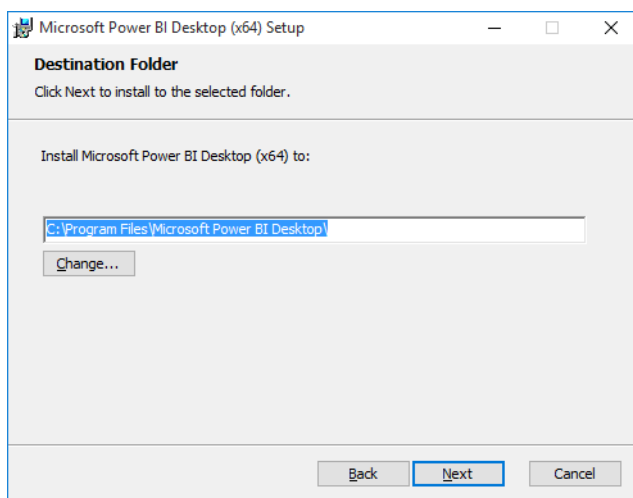
- d) When you see the Welcome screen, click **Next** to continue with the installation.



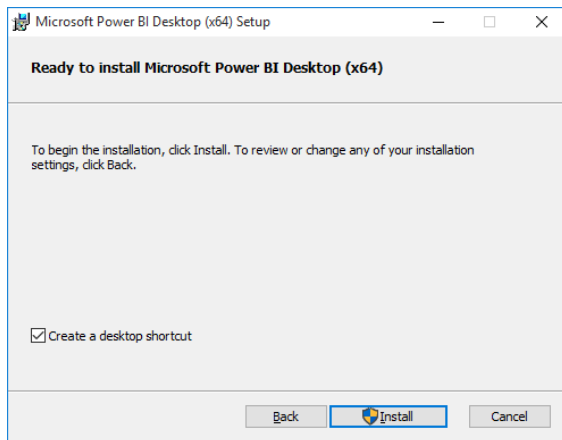
- e) Click the checkbox to accept the license agreement and click **Next**.



- f) Accept the default location for the installation and click **Next**.



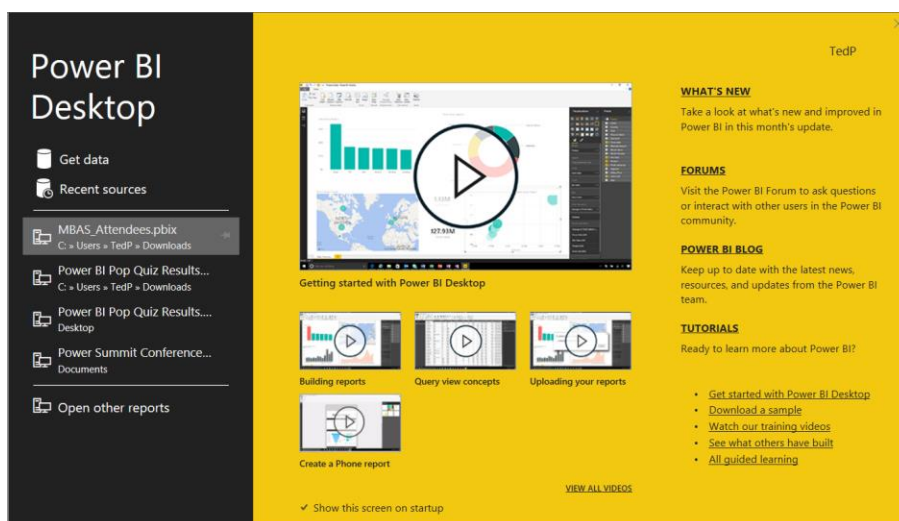
- g) On the next screen, click **Install**.



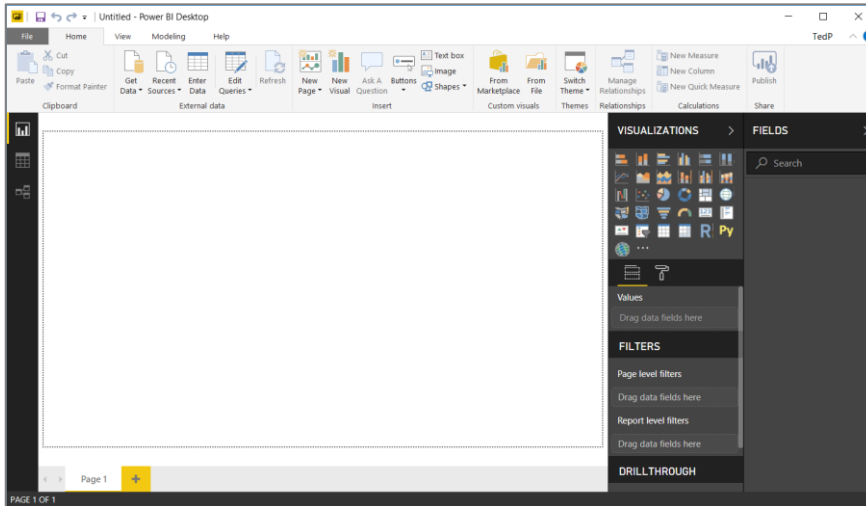
- h) When you see the **Completed the Microsoft Power BI Desktop Setup Wizard** screen, click **Finish** to launch Power BI Desktop.



- i) When Power BI Desktop launches for the first time, it displays a Welcome screen as shown in the following desktop. Click the (X) button in the upper right corner to close this window.



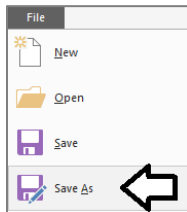
- At this point, you should have Power BI Desktop running as shown in the following screenshot. You are now ready to move ahead to the next exercise.



Exercise 2: Creating Queries to Import Data into Power BI Desktop

In this exercise you will use Power BI Desktop to import sales data from three separate CSV files. Moving through the steps of this exercise will give you a chance to begin working with the Power Query features that are built into Power BI Desktop.

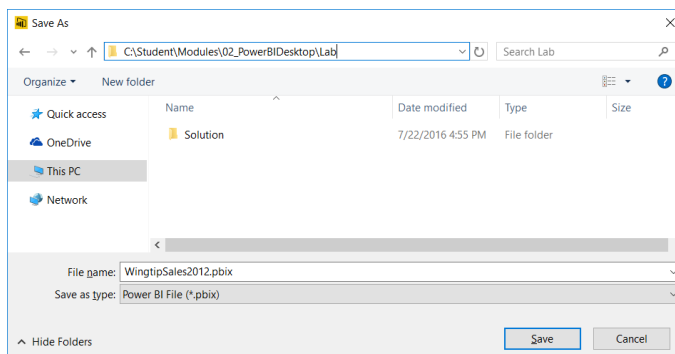
- Open Power BI Desktop if it is not already open.
- Power BI Desktop should now be open with a new and empty project.
- Before beginning your work, save the empty project as a PBIX file.
 - Drop down the **File** menu and click the **Save As** command.



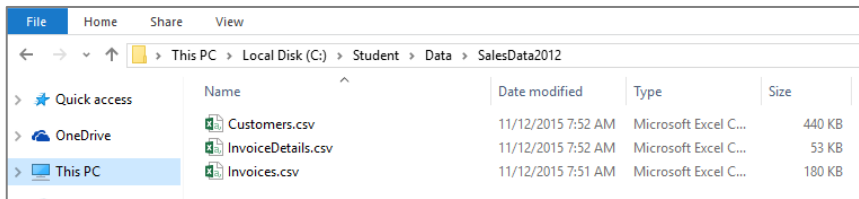
- Save the PBIX file as **WingtipSales2012.pbix** using the following path location.

C:\Student\Modules\02_PowerBIDesktop\Lab\wingtipsales2012.pbix

- Click the **Save** button on the **Save As** dialog to save the PBIX file.

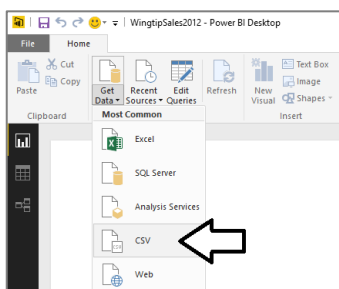


4. Using Windows Explorer, open the folder at **C:\Student\Data\SalesData2012** and examine what's inside. You should observe that there are three CSV files named **Customers.csv**, **Invoices.csv** and **InvoiceDetails.csv**. Over the next few steps, you will use these three files to import data into Power BI Desktop to create a new dataset.

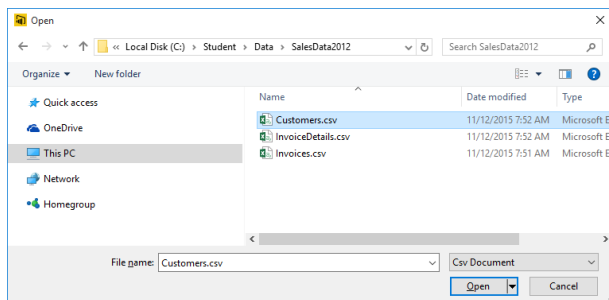


You can also get these files on GitHub at <https://github.com/CriticalPathTraining/PBI365/tree/master/Student/Data/SalesData2012>.

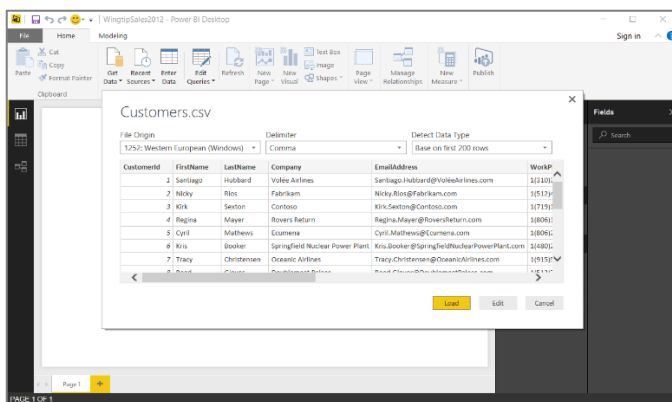
5. Return to Power BI Desktop and examine the ribbon. Drop down the **Get Data** menu button and select the **CSV** command.



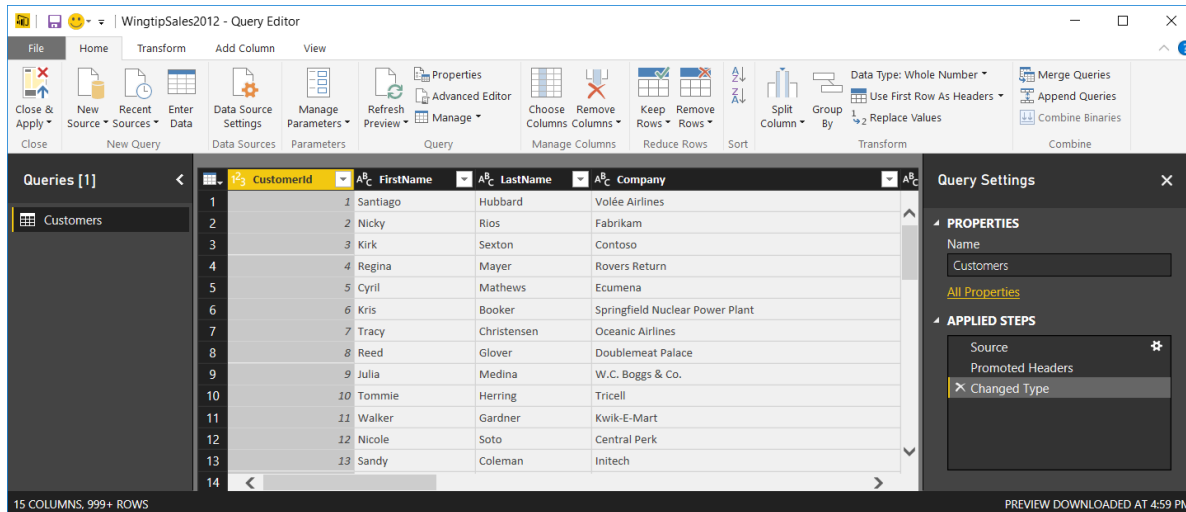
6. In the **Open** dialog, navigate to the folder at **C:\Student\Data\SalesData2012** and select the file named **Customers.csv**. Click the **Open** button in the bottom right of the dialog to begin the import process.



7. At this point you are presented with the **Customers.csv** dialog which displays a **Load** button, an **Edit** button and a **Cancel** button. Click the **Edit** button to view the data from **Customers.csv** in the Power Query Editor window.

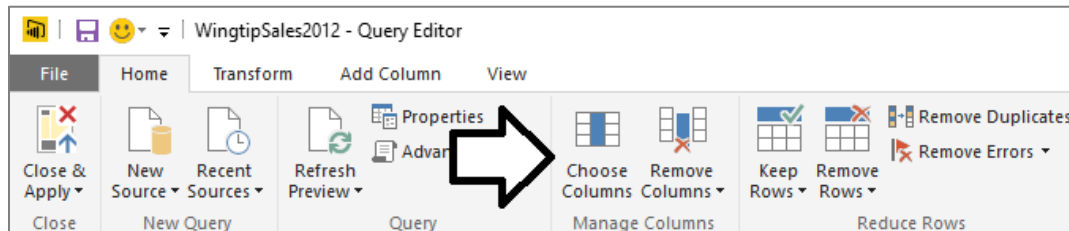


8. You should now be looking at the data from **Customers.csv** in the Query Editor window as shown in the following screenshot. If you examine the **Query Settings** task pane on the right-hand side, you can see that the query is named **Customers** and there are three steps in the **Applied Steps** list.

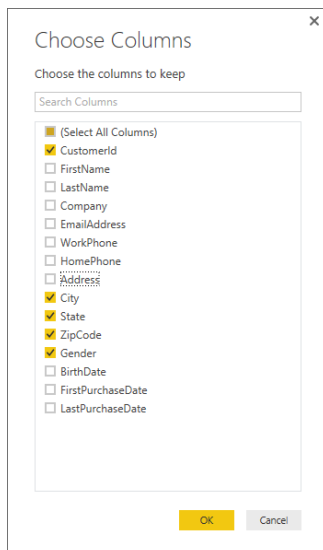


9. The **Customers** query contains several unneeded columns that will not be used to analyze sales data. In this step you will remove these unneeded columns to reduce the amount of data that is loaded into the data model.

- a) In the ribbon of the **Query Editor** window, click the **Choose Columns** button.



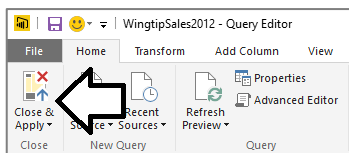
- b) In the **Choose Columns** dialog, begin by unselecting all columns.
c) Select the columns named **CustomerId**, **City**, **State**, **ZipCode** and **Gender** as shown in the following dialog.



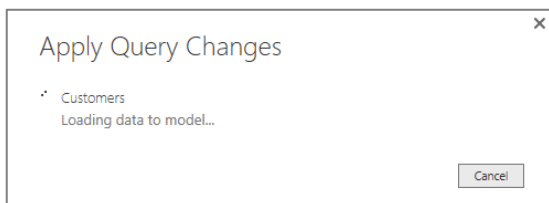
10. Once you have selected the desired columns, click the **OK** button to close the **Choose Columns** dialog. You should be able to observe that the Query Editor window now displays data for the columns you selected.

	1 ² ₃ CustomerId	A ^B _C City	A ^B _C State	1 ² ₃ ZipCode	A ^B _C Gender
1		1 Manhattan Beach	CA	90266	M
2		2 Austin	TX	78753	M
3		3 Colorado Springs	CO	80924	M
4		4 Amarillo	TX	79121	F
5		5 Lubbock	TX	79407	M
6		6 Tempe	AZ	85281	M
7		7 El Paso	TX	79936	M
8		8 Austin	TX	78730	M
9		9 San Antonio	TX	78253	F
10		10 Manhattan Beach	CA	90266	M

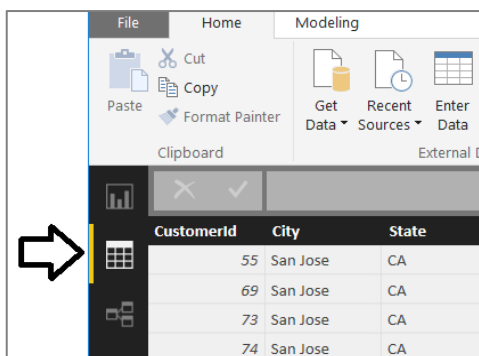
11. Now it's time to load the data from **Customer.csv** into the data model. Accomplish this by clicking the **Close & Apply** button in the ribbon of the Query Editor window to close the Query Editor window and import the data.



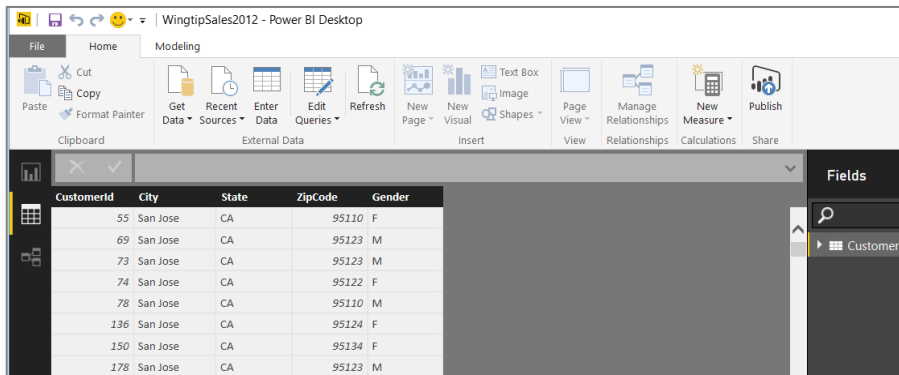
12. Power BI Desktop displays the **Apply Query Changes** dialog while it's executing the query and importing the data. Note that the **Apply Query Changes** dialog will close automatically after the data has been imported.



13. You should now be looking at the main window of Power BI Desktop in report view which displays a new and empty report. Locate the three buttons on the sidebar on the left-hand side of the screen. Click the button in the middle to move from report view to data view.



14. Once you have moved over to data view, you should be able to verify that the **Customers** table has been created. On the right side of the window, the **Fields** list shows the **Customers** table which is selected. The data from the **Customers** table is shown in a grid view in the middle of the screen.



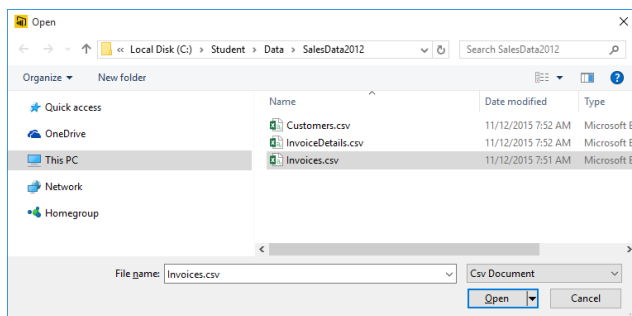
The screenshot shows the Power BI Desktop interface. The ribbon at the top includes File, Home, and Modeling tabs. The Home tab is active, showing options like Paste, Copy, Cut, and Format Painter. The Modeling tab shows options like Get Data, Recent, Enter Data, Edit Queries, Refresh, New Page, New Visual, Image, Shapes, Page View, Manage Relationships, New Measure, and Publish. The main area displays a table with the following data:

CustomerId	City	State	ZipCode	Gender
55	San Jose	CA	95110	F
69	San Jose	CA	95123	M
73	San Jose	CA	95123	M
74	San Jose	CA	95122	F
78	San Jose	CA	95110	M
136	San Jose	CA	95124	F
150	San Jose	CA	95134	F
178	San Jose	CA	95123	M

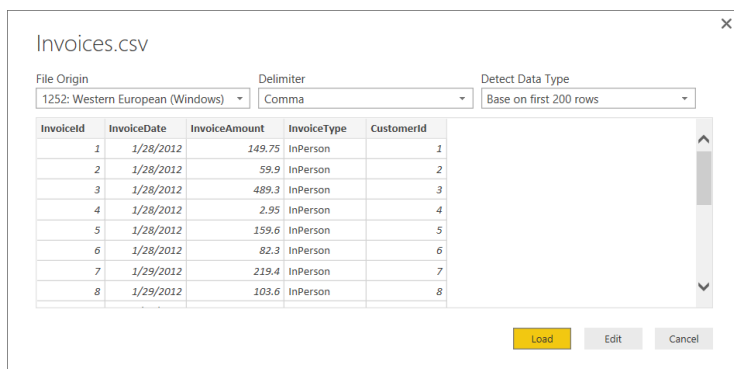
The right sidebar shows the Fields pane with the Customers table selected.

You might observe that the data from the **Customers** table is not sorted in an intuitive fashion. For example, you might expect the default sort order to be the same as the underlying CSV file which orders customers by **CustomerId**. Instead, the table has a seemingly random sort order which is a side effect of the way data is stored in memory given the column-based nature of the tabular model. You can right-click the column header for the **CustomerId** column and run a command to sort the data in the table if it makes you feel better, but this action will not really have any effect because changes to the sort order are only temporary and they are not saved as part of the underlying data model.

15. You have imported one table into the data model. Now you will import a second table to get additional data on invoices.
- In the Ribbon of the Power BI Desktop window, select the **Home** tab.
 - Drop down the **Get Data** menu button and select the **CSV** command.
 - In the **Open** dialog, navigate to the folder at **C:\Student\Data\SalesData2012** and select the file named **Invoices.csv**.



- Click the **Open** button in the bottom right of the dialog to begin the import process.
- In the **Invoices.csv** dialog, click the **Load** button to execute the query and import data to create the **Invoices** table.



The screenshot shows the Invoices.csv import dialog. The File Origin is set to 1252: Western European (Windows). The Delimiter is set to Comma. The Detect Data Type is set to Base on first 200 rows. The table preview shows the following data:

InvoiceId	InvoiceDate	InvoiceAmount	InvoiceType	CustomerId
1	1/28/2012	149.75	InPerson	1
2	1/28/2012	59.9	InPerson	2
3	1/28/2012	489.3	InPerson	3
4	1/28/2012	2.95	InPerson	4
5	1/28/2012	159.6	InPerson	5
6	1/28/2012	82.3	InPerson	6
7	1/29/2012	219.4	InPerson	7
8	1/29/2012	103.6	InPerson	8

The Load button is highlighted in yellow.

- WingtipSales2012 - Power BI Desktop

FileHomeModeling

Paste

Cut

Copy

Format Painter

Get Data

Recent Sources

Enter Data

Edit Queries

Refresh

New Page

New Visual

Text Box

Image

Shapes

Page View

Manage Relationships

New Measure

Publish

ClipboardExternal DataInsertViewRelationshipsCalculationsShare

InvoiceId

InvoiceDate

InvoiceAmount

InvoiceType

CustomerId

1	Saturday, January 28, 2012	149.75	InPerson	1
2	Saturday, January 28, 2012	59.9	InPerson	2
3	Saturday, January 28, 2012	489.3	InPerson	3
4	Saturday, January 28, 2012	2.95	InPerson	4
5	Saturday, January 28, 2012	159.6	InPerson	5
6	Saturday, January 28, 2012	82.3	InPerson	6
7	Sunday, January 29, 2012	219.4	InPerson	7
8	Sunday, January 29, 2012	103.6	InPerson	8
9	Sunday, January 29, 2012	17.7	InPerson	9
10	Sunday, January 29, 2012	20.65	InPerson	10
11	Sunday, January 29, 2012	95.6	InPerson	11
12	Sunday, January 29, 2012	64.8	InPerson	12

Fields

Customers

Invoices

TABLE: Invoices (5,228 rows)

-
- The screenshot shows a Windows File Explorer window titled 'Open'. The address bar indicates the current location is 'Local Disk (C:) > Student > Data > SalesData2012'. The left sidebar shows the 'This PC' view with 'Quick access', 'OneDrive', 'This PC', 'Network', and 'Homegroup' listed. The main pane displays a list of files in the 'SalesData2012' folder:
- | Name | Date modified | Type |
|--------------------|--------------------|-----------------------------|
| Customers.csv | 11/12/2015 7:52 AM | Microsoft Excel spreadsheet |
| InvoiceDetails.csv | 11/12/2015 7:52 AM | Microsoft Excel spreadsheet |
| Invoices.csv | 11/12/2015 7:51 AM | Microsoft Excel spreadsheet |
- The 'InvoiceDetails.csv' file is selected. At the bottom of the window, the 'File name' field contains 'InvoiceDetails.csv' and the file type is set to 'Csv Document'. The 'Open' button is highlighted.

- File** Home Modeling Sign in

Paste Cut Copy Format Painter Clipboard

Get Data Recent Sources Enter Data Edit Queries Refresh New Page View New Visual Insert Text Box Image Shapes View Manage Relationships Calculations Publish Share

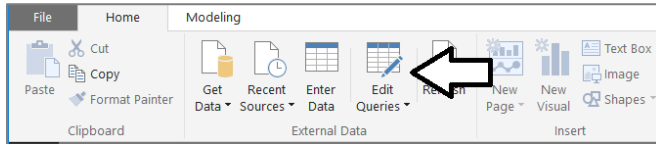
ID	Quantity	SalesAmount	InvoiceId	ProductId
46	5	99.75	28	6
81	5	99.75	44	6
194	5	99.75	109	6
487	5	99.75	258	6
581	5	99.75	307	6
601	5	99.75	317	6
641	5	99.75	337	6

Fields

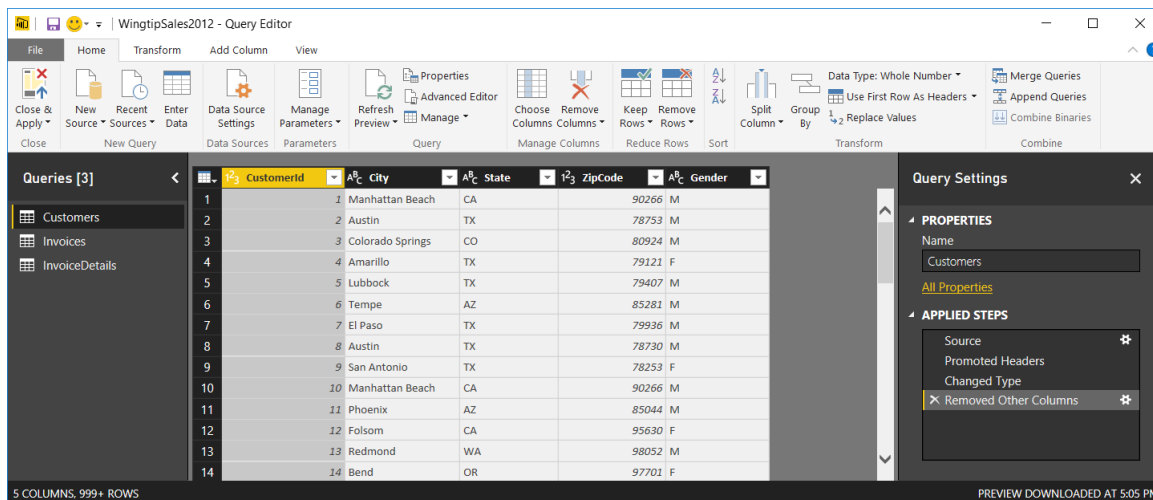
 - Customers
 - InvoiceDetails
 - Invoices

17. While you are done working with the query and data import aspects of Power BI Desktop for this particular lab exercise, it's still important that you know how to open and edit queries whenever it is required. You will now go through the steps to open the Query Editor window to see how to get back to query editing mode.

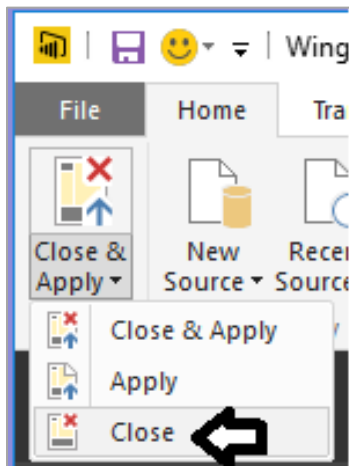
- a) In the ribbon of the Power BI Desktop window, click the **Edit Queries** button to open the Query Editor window.



- b) You should now see the Query Editor window displaying queries named **Customers**, **Invoices** and **InvoiceDetails** in the **Queries** list in the right-hand side of the screen. You can select any of these queries and continue to edit them if required. However, in this case there is no need to edit the queries any more.



- c) Close the Query Editor window by dropping down the **Close and Apply** menu button and clicking the **Close** command.



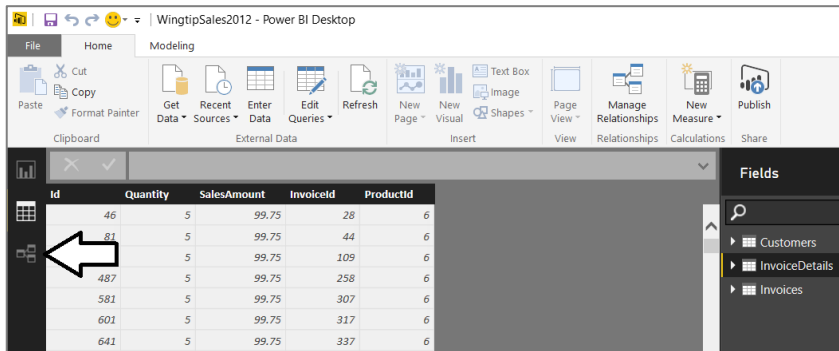
18. Save your work by clicking the Save button on the top left of the Power BI Desktop window.

You have now finished importing data into your data model using Power Query. The next step is to begin modeling the data using the Power Pivot and tabular database modeling features of Power BI Desktop.

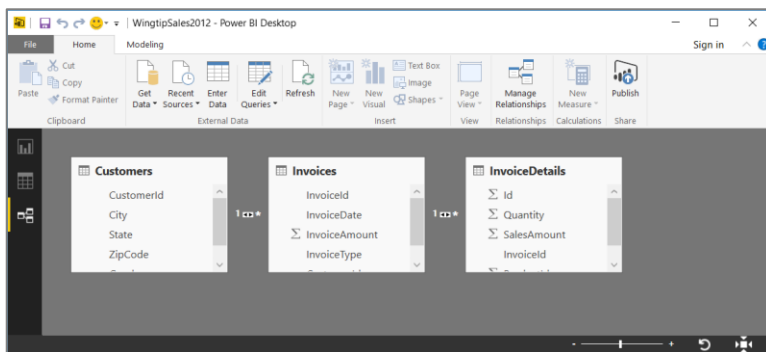
Exercise 3: Extending a Data Model using Calculated Columns and Measures

In this exercise you will begin to use the data modeling features of Power BI Desktop. You will begin by ensuring the appropriate relationships have been created between the three tables in the data model. After that, you will extend the tables within the data model by adding calculated columns and measures.

1. Currently, the Power BI Desktop window should be displaying in data view. Click on the bottom button in the sidebar on the left-hand side of the screen to move to relationship view.

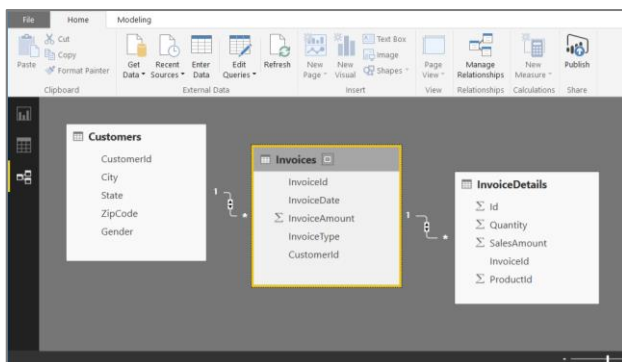


2. Once you navigate to relationship view, you should be able to see all three tables as shown in the following screenshot.



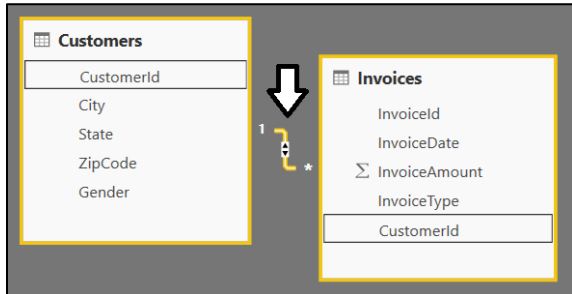
Note that Power BI desktop automatically created relationships between these tables because it determined that the **Customers** table and **Invoices** table both contain a common field named **CustomerId** and also that the **Invoices** tables and the **InvoicesDetails** table both contain a common field named **InvoiceId**.

3. Using the mouse, reposition and resize each of the three tables in relationship view so you can see all the fields in each one. Also space the tables out a bit so it is easier to see the relationships between them. You can see that these relationships are displayed using a line that starts with a 1 and end with a (*) to indicate a one-to-many relationship.



4. Take a moment to inspect the relationships that have been created.

- a) Click on the line that represents the relationship between the **Customers** table and the **Invoices** table. Once the relationship has been selected, you can see that the **CustomerId** field in each table becomes highlighted to indicate that those are the two fields used to define the one-to-many relationship.



- b) Now, double-click on the line that represents the relationship between the **Customers** table and the **Invoices** table to display the **Edit Relationship** dialog. There is no need to change anything as you are just observing. Once you have seen what's inside this dialog, click the **OK** button to close it.

Edit Relationship

Select tables and columns that relate to one another.

Invoices

InvoiceId	InvoiceDate	InvoiceAmount	InvoiceType	CustomerId
1	Saturday, January 26, 2012	143.79	InPerson	1
2	Saturday, January 26, 2012	59.9	InPerson	2
3	Saturday, January 26, 2012	489.3	InPerson	2

Customers

CustomerId	City	State	ZipCode	Gender
55	San Jose	CA	95110	F
69	San Jose	CA	95123	M
73	San Jose	CA	95123	M

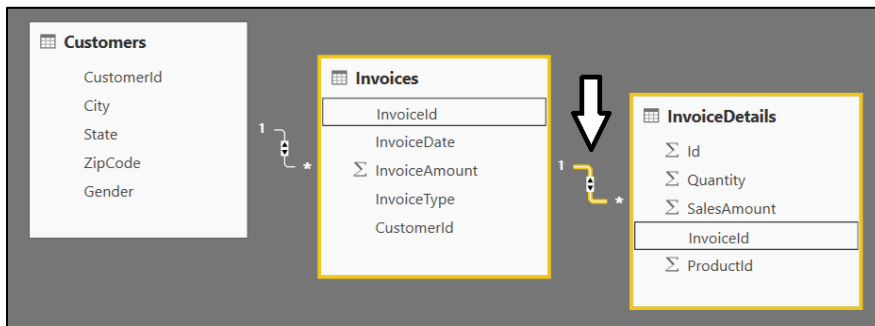
Cardinality: Many to One (1:M) Cross filter direction: Both

☒ Make this relationship active

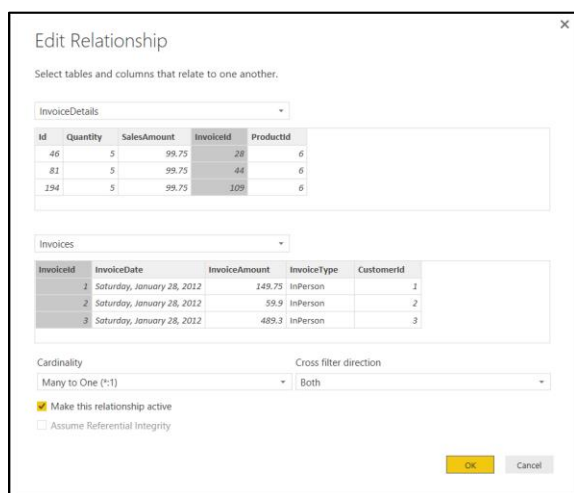
☐ Assume Referential Integrity

OK Cancel

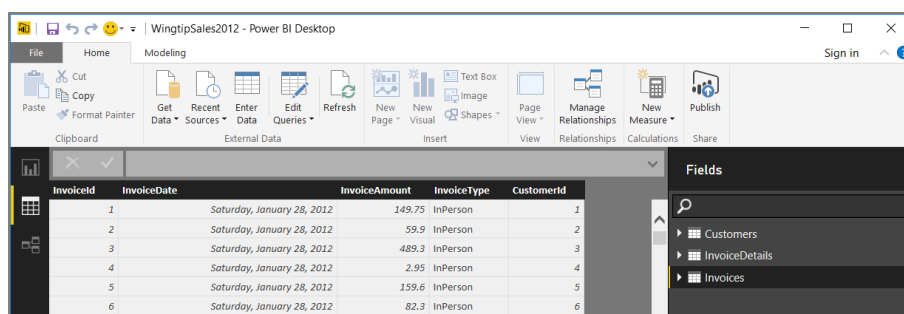
- c) Click on the line that represents the relationship between the **Invoices** table and the **InvoiceDetails** table. When you do, you can see that the **InvoiceId** field in each table becomes highlighted to indicate that those are the two fields used to define the one-to-many relationship.



- d) Now, double-click on the line that represents the relationship between the **Invoices** table and the **InvoiceDetails** table to display the **Edit Relationship** dialog. Once you have seen the dialog, click the **OK** button to close the dialog.



5. Modify the formatting on the **InvoiceDate** field of the **Invoices** table.
 - a) Navigate back to data view and select the **Invoices** table from the **Fields** list.



- b) Select the **InvoiceDate** column by clicking its column header.
 - c) Drop down the **Format** menu in the ribbon and select the date format of **3/14/2001 (M/d/yyyy)**.

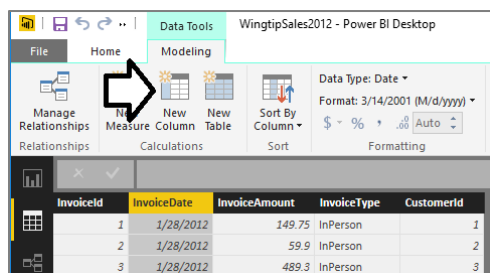


- d) The data in the **InvoiceDate** field should now be display in the new format.

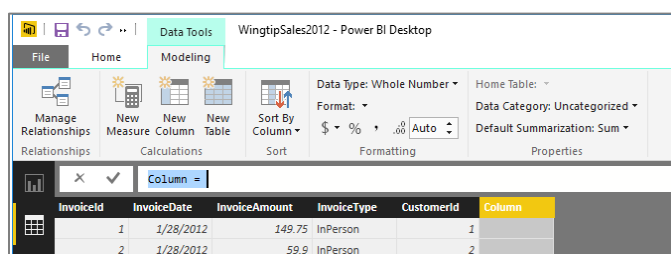
InvoiceId	InvoiceDate	InvoiceAmount	InvoiceType	CustomerId
1	1/28/2012	149.75	InPerson	1
2	1/28/2012	59.9	InPerson	2
3	1/28/2012	489.3	InPerson	3
4	1/28/2012	2.95	InPerson	4
5	1/28/2012	159.6	InPerson	5
6	1/28/2012	82.3	InPerson	6
7	1/29/2012	219.4	InPerson	7

6. Add a new calculated column to the **Invoice** table to display the current month.

a) Click the **New Column** button in the ribbon to create a new column.



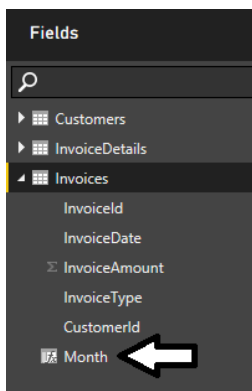
b) The column is initially created with a generic name of **Column** and it has no formula in the formula bar.



c) Give the new column a name of **Month** and provide a DAX formula for determining the month from the **InvoiceDate** column by replacing what's inside the formula bar with the following text. When you are done typing the formula, press the **Enter** key.

Month = FORMAT(Invoices[InvoiceDate], "MMM yyyy")

d) At this point, you should be able to see the **Month** column in the **Fields** list. Note that the **Month** column is displayed with a special function icon (i.e. **Fx**) to indicate that it is a calculated column.



Now that you have created the **Month** column, you will be able to use it to display data on reports that is broken out by month. However, this can cause a sorting issue because month names will be sorted alphabetically by default. For example, February will be

sorted before January which will cause confusion and produce strange results. In the next step you will create an extra column that will be used assist in sorting the Month column in a chronologic fashion which is what people expect.

7. Create an additional calculated column named **MonthSort** to supply the proper sorting behavior for the **Month** column.

- Click the **New Column** button in the ribbon to create a new column.
- Give the new column a name of **MonthSort** and use the following DAX formula.

```
MonthSort = Format(Invoices[InvoiceDate], "yyyyMM")
```

- Once you add the **MonthSort** column, you should be able to see it on the **Fields** list. You should also be able to see that the data in the **MonthSort** column contains a sortable text value which concatenates a 4-digit year value together with the 2-digit month value.

InvoiceId	InvoiceDate	InvoiceAmount	InvoiceType	CustomerId	Month	MonthSort
1	1/28/2012	149.75	InPerson		Jan 2012	201201
2	1/28/2012	59.9	InPerson		Jan 2012	201201
3	1/28/2012	489.3	InPerson		Jan 2012	201201
4	1/28/2012	2.95	InPerson		Jan 2012	201201
5	1/28/2012	159.6	InPerson		Jan 2012	201201
6	1/28/2012	82.3	InPerson		Jan 2012	201201
7	1/29/2012	219.4	InPerson		Jan 2012	201201
8	1/29/2012	103.6	InPerson		Jan 2012	201201
9	1/29/2012	17.7	InPerson		Jan 2012	201201
10	1/29/2012	20.65	InPerson		Jan 2012	201201
11	1/29/2012	95.6	InPerson		Jan 2012	201201
12	1/29/2012	64.8	InPerson		Jan 2012	201201
13	1/29/2012	69.65	InPerson		Jan 2012	201201
14	1/29/2012	39.9	InPerson		Jan 2012	201201

- Now it is time to configure the **Month** column to use **MonthSort** as its sort column. Begin by clicking on the column header of the **Month** column to select it. Next, drop down the **Sort By Column** button and select the **MonthSort** column.

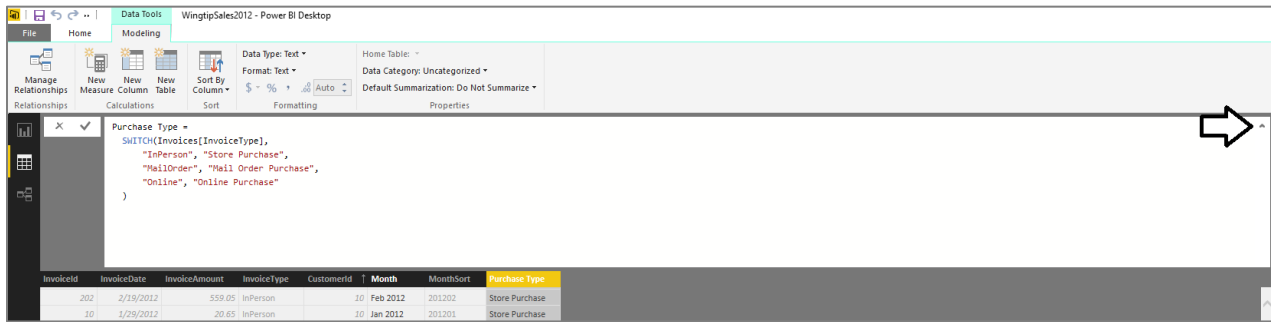
InvoiceId	InvoiceDate	InvoiceAmount	InvoiceType	CustomerId	Month	MonthSort
1	1/28/2012	149.75	InPerson		Jan 2012	201201
2	1/28/2012	59.9	InPerson		Jan 2012	201201
3	1/28/2012	489.3	InPerson		Jan 2012	201201
4	1/28/2012	2.95	InPerson		Jan 2012	201201
5	1/28/2012	159.6	InPerson		Jan 2012	201201
6	1/28/2012	82.3	InPerson		Jan 2012	201201
7	1/29/2012	219.4	InPerson		Jan 2012	201201
8	1/29/2012	103.6	InPerson		Jan 2012	201201

8. Create a calculated column named **Purchase Type** to provide a more human readable value for the **InvoiceType** column.

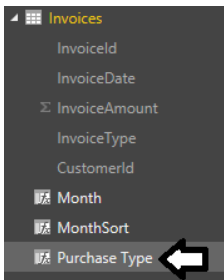
- Click the **New Column** button in the ribbon to create a new column.
- Given the new column a name of **Purchase Type** using the following formula.

```
Purchase Type =
SWITCH(Invoices[InvoiceType],
    "InPerson", "Store Purchase",
    "MailOrder", "Mail Order Purchase",
    "Online", "Online Purchase"
)
```

- Note that you can expand the height of the formula bar by clicking the downward facing button on the right-hand side. This will make it much easier to write multi-line DAX expressions which are far more readable and also more maintainable. Also remember that you can add a line break into a DAX formula by holding down the **SHIFT** key and then pressing the **ENTER** key. Once you have typed in the DAX formula for the **Purchase Type** field, press the **Enter** key.



- d) At this point, you should be able to see the **Purchase Type** column in the **Fields** list.

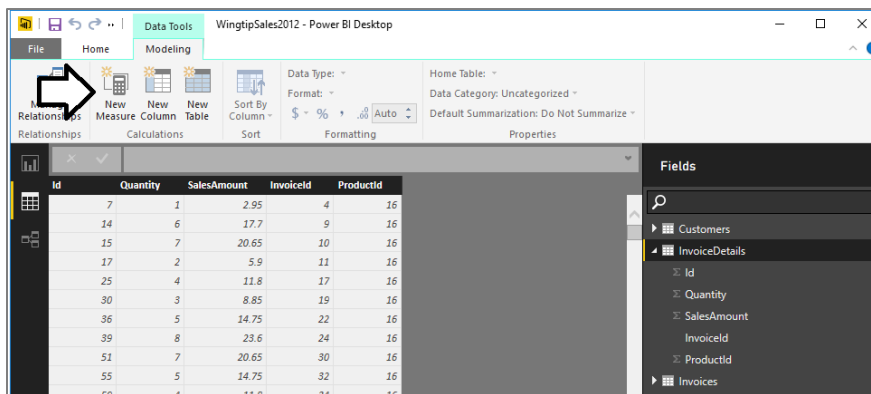


- e) Inspect the values in the **Purchase Type** column to see the effect of the DAX formula you have written.

InvoiceId	InvoiceDate	InvoiceAmount	InvoiceType	CustomerId	Month	MonthSort	Purchase Type
5220	12/31/2012	25.85	InPerson	2653	Dec 2012	201212	Store Purchase
5219	12/31/2012	319.3	InPerson	2652	Dec 2012	201212	Store Purchase
5218	12/31/2012	162.6	InPerson	2651	Dec 2012	201212	Store Purchase
5217	12/31/2012	183.6	InPerson	2650	Dec 2012	201212	Store Purchase
5216	12/31/2012	114.6	InPerson	2649	Dec 2012	201212	Store Purchase
5215	12/31/2012	59.8	InPerson	2648	Dec 2012	201212	Store Purchase
5214	12/31/2012	41.8	InPerson	2647	Dec 2012	201212	Store Purchase

9. Add a measure to the **InvoiceDetails** table to calculate total sales revenue.

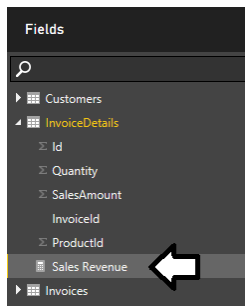
- Select the **InvoiceDetails** table by clicking on it in the **Fields** list.
- Click the **New Measure** button in the ribbon to create a new measure.



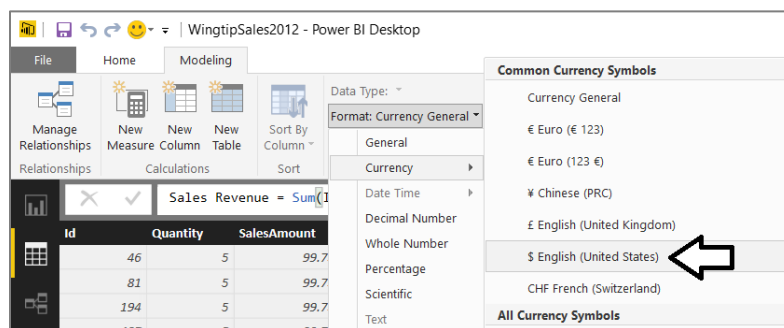
- Create a new measure named **Sales Revenue** using the following DAX formula. Once you have typed the formula into the formula bar, press the **Enter** key to create the new measure.

Sales Revenue = Sum(InvoiceDetails[SalesAmount])

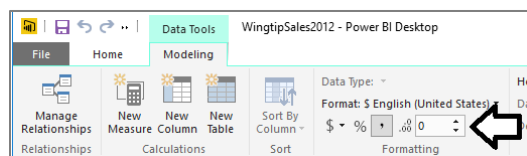
- d) At this point you should be able to see the **Sales Revenue** measure in the **InvoiceDetails** table in the **Fields** list. Note that the **Sales Revenue** field is displayed with a special calculator icon to indicate that it is a measure.



- e) Make sure the **Sales Revenue** measure is still selected in the **Fields** list. Modify the formatting of this measure by dropping down the **Format** menu on the ribbon and selecting a format type of **Currency > \$ English (United States)**.



- f) Enhance the formatting of the **Sales Revenue** column by reducing the number of decimal places shown from **2** to **0**.

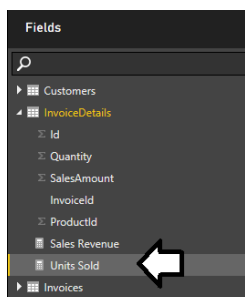


10. Create a second measure in the **InvoiceDetails** table to calculate the total number of units sold.

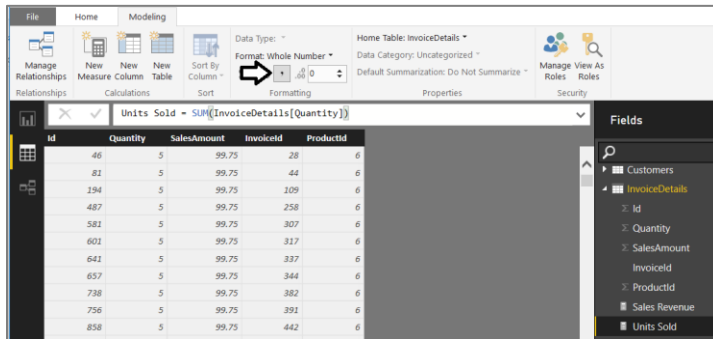
- a) Click the **New Measure** button in the ribbon to create a new measure.
- b) Create a new measure named **Units Sold** using the following DAX formula. Once you have type the formula into the formula bar, press the **Enter** key to create the new measure.

```
Units Sold = SUM(InvoiceDetails[Quantity])
```

- c) You should be able to see the new **Units Sold** measure in the **Fields** list.



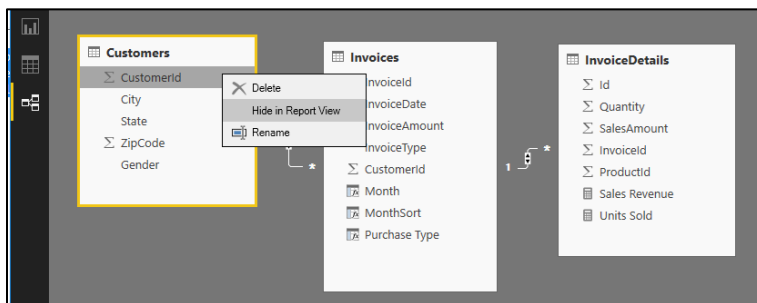
- d) Make sure the **Units Sold** measure is selected in the **Fields** list and modify the formatting by clicking and selecting the button on the ribbon with the comma which adds a thousands separator so larger numbers are formatted with commas (e.g. 2,500).



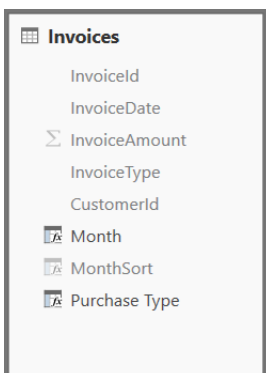
You are now done adding calculated columns and measures. You will now complete you modeling work by hiding fields that do not need to be seen when designing reports.

11. Hide all the fields that are not needed when designing reports.

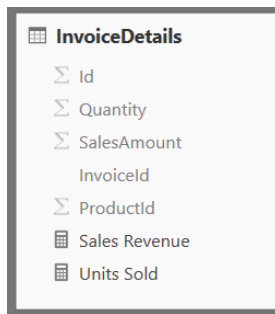
- a) Navigate to relationship view.
b) Right-click on the **CustomerId** field in the **Customers** table and click the **Hide in Report View** command.



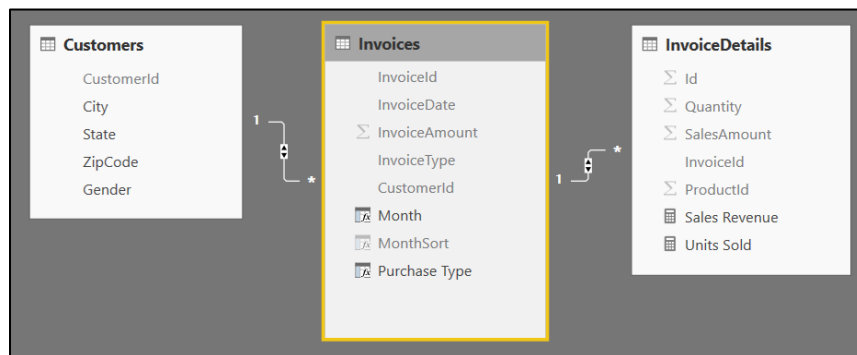
- c) Use the same technique of right-clicking a field and selecting the **Hide in Report View** command to hide all the fields in the **Invoices** table except for the **Month** column and the **Purchase Type** column.



- d) Use the same technique of right-clicking a field and selecting the **Hide in Report View** command to hide all the fields in the **InvoiceDetails** table except for the **Sales Revenue** measure and the **Units Sold** measure.



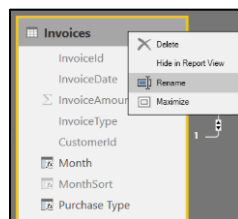
e) The tables in relationship view should appear as they do in the following screenshot. You are now done modeling the data.



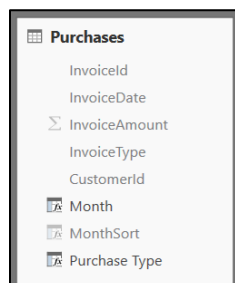
12. Save your work by clicking the **Save** button on the top left of the Power BI Desktop window.

13. Change the name of tables inside the data model to make them more user friendly.

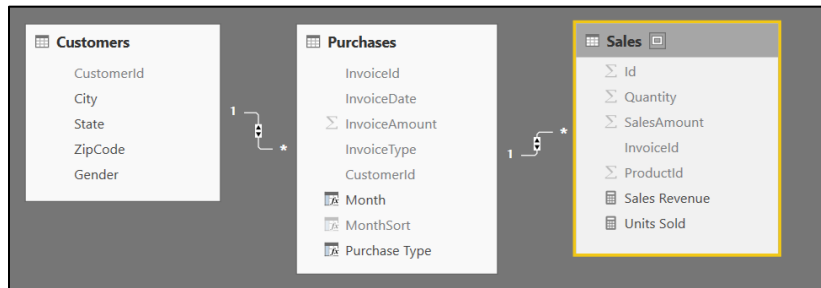
a) Right-click on the title header of the **Invoices** table and click the Rename command.



b) Rename the Invoices table to Purchases.



c) Use the same technique to rename the **InvoiceDetails** table to **Sales**. When you are done, your project in Relationship view should match the following screenshot.

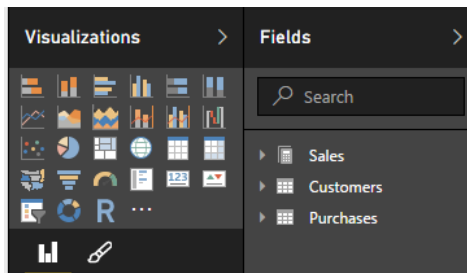


When you examine the **Sales** table, you can observe that all fields in this table have been hidden except for the two measures names **Sales Revenue** and **Units Sold**. When you have designed a table such as the **Sales** that only displays measures, Power BI Desktop will display that table at the top of the tables list when you are in report view. However, Power BI Desktop has an unfortunate bug where it will not recognize a table as having only measures until you close and reopen the project. In the next step, you will close Power BI Desktop. Then you will restart Power BI Desktop and reopen the current project. This will allow you to see how Power BI Desktop displays measure-only tables in Report View.

14. Close and restart Power BI Desktop to see the effects of a measure-only table.
 - a) Close Power BI Desktop.
 - b) Restart Power BI Desktop
 - c) Reopen the Power BI Desktop project file named **WingtipSales2012.pbix**.
15. Examine the tables in Report View.
 - a) Navigate to Report view.



- b) Examine the tables shown in **Fields** list on the right side of the Power BI Desktop application window. You can see that the **Sales** table is displayed first with a calculator icon. The two other tables shown below are displayed with a standard table icon.

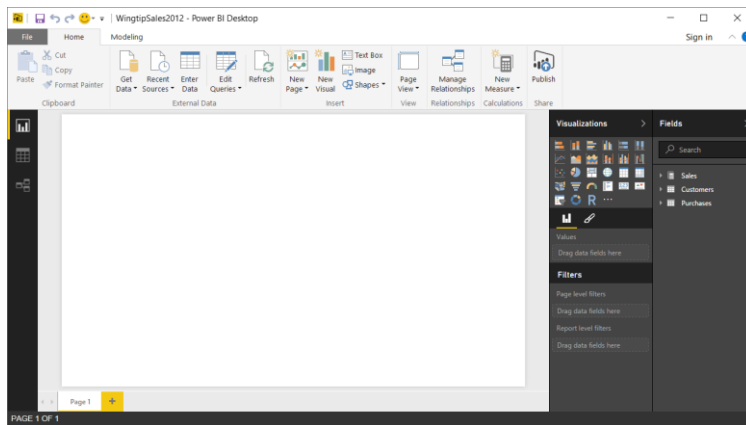


In terms of standard BI intelligence terminology, you can say that Power BI Desktop is recognizing the **Sales** table as a **fact table** because it only displays measures. The other two tables are recognized as **dimension tables** because they display one or more fields that are not measures.

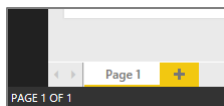
Exercise 4: Create a Report using Power BI Desktop

In this exercise you will design a report with two pages of visuals using the data model you created in the previous exercise.

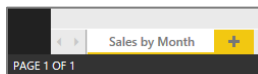
1. Make sure you still have the **Wingtipsales2012.pbix** file open that you created in the previous exercise.
2. Click on the top bottom in the sidebar to place Power BI Desktop into report view.
3. At this point, you should see the default report which contain an empty page named **Page 1**.



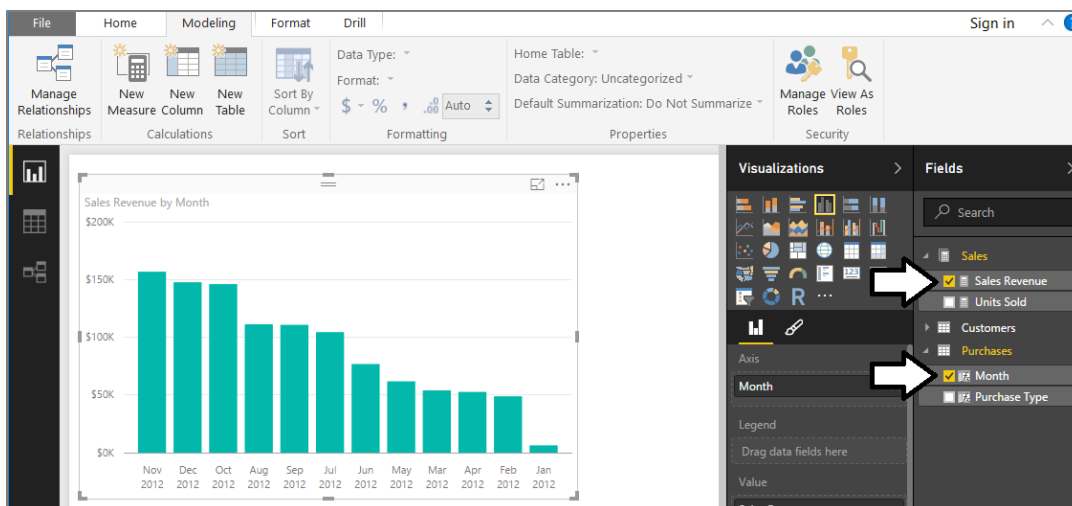
4. Change the title of the report's one and only page.
- a) Locate page title with a default value of **Page 1** on the bottom left corner of the report and double-click on it to edit its value.



- b) Modify the page title to **Sales by Month** and press the **Enter** key to enter your changes

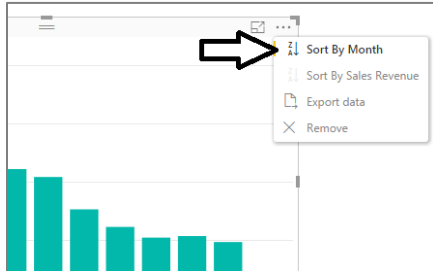


5. Add a new visual to display sales revenue by month.
- a) At this point the **Sales by Month** page should be empty without any visuals.
- b) In the **Fields** list, click the checkbox to select the **Sales Revenue** measure. Note that this action will add a new visual to the page based on a clustered column chart.
- c) Next, click the checkbox to select the **Month** column which will automatically add that column into the **Axis** well and produce a column chart.
- d) Use the mouse to resize the column chart so that it displays all 12 columns as shown in the following screenshot.

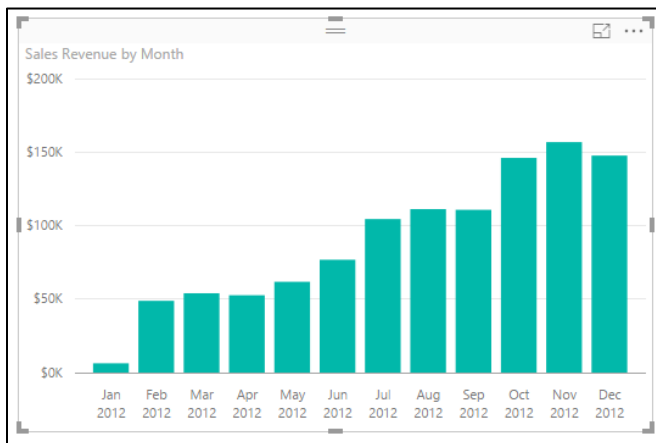


Note that by default the column chart displays its columns with the largest values sorted to the left. However, that doesn't make much sense when the column axis is based on a time value such as month. You need to sort the columns so that the columns are sorted chronologically from left to right.

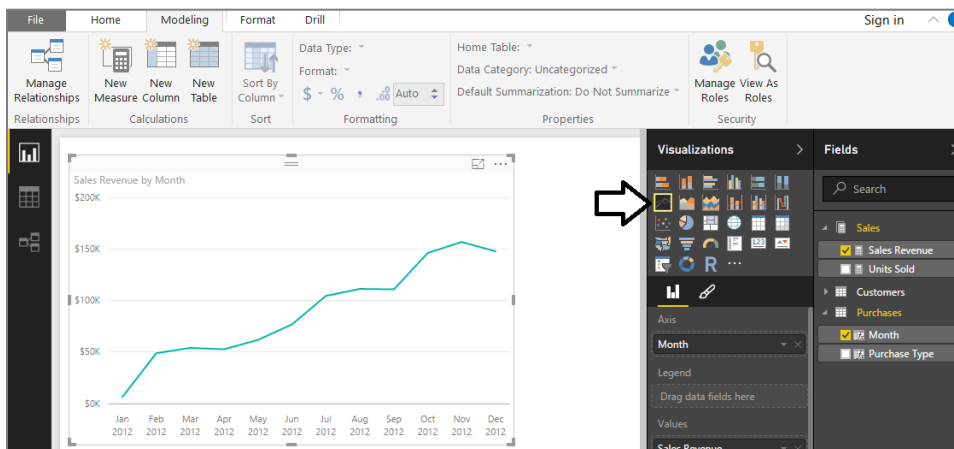
- e) Click on the ellipse flyout menu (...) at the top right corner of the column chart visual and click the **Sort by Month** menu command to sort the columns by month. Note you might have to click the **Sort by Month** menu command twice to get the desired effect of display the columns chronologically from left to right.



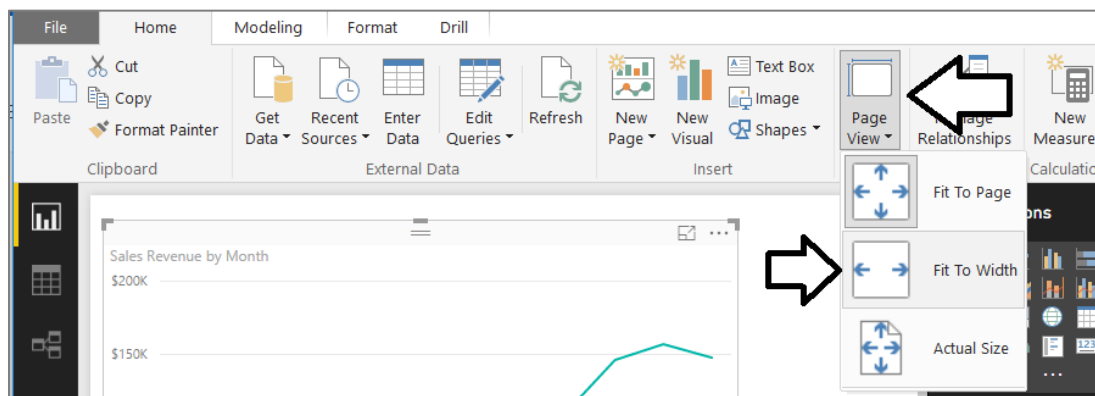
- f) When you are done, your column chart should match the one shown in the following screenshot.



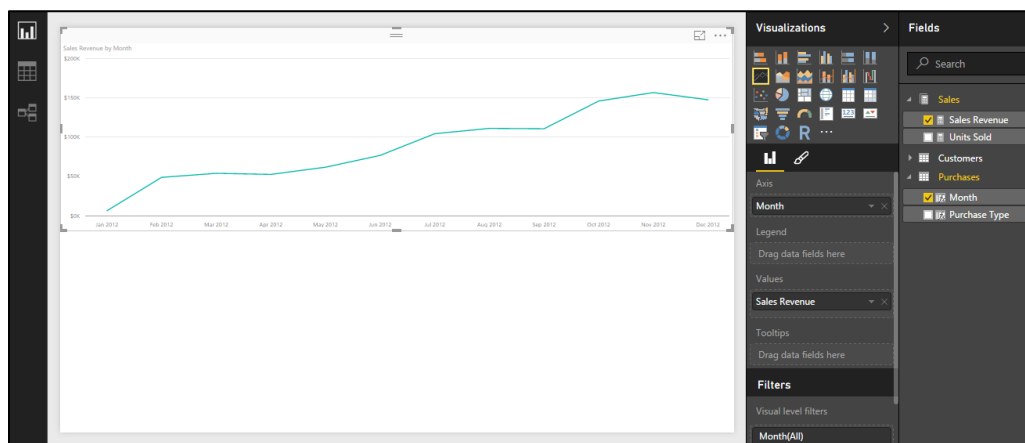
- g) With the column chart selected, change the visual type to a line chart by clicking the Line Chart button in the Visualizations Pane.



- h) Navigate to the Home tab in the ribbon and locate the **Page View** dropdown menu. Drop down the **Page View** menu and select the **Fit to Width** menu command so you can see the entire report page.

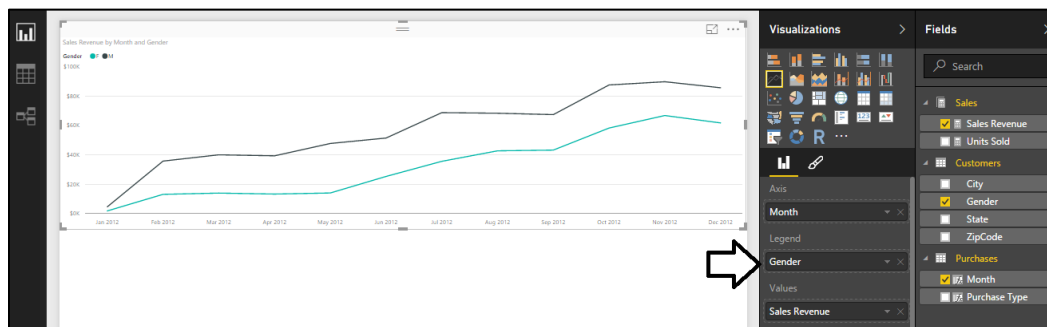


- i) Use the mouse to reposition the Line chart so it takes up the top half of the report page as shown in the following screenshot.



6. Add a legend to the line chart to display sales revenue by gender.

- a) Select the line chart. Drag and drop the **Gender** field from the **Customers** into the **Legend** well in the **Visualizations** pane.



- b) Locate the **Legend** section in the **Format** properties pane and set the legend **Position** property to **Right**.



7. Create a second visual to display sales revenue by month and purchase type.
 - a) Select the visual you created in the previous step and copy it to the Windows clipboard.
 - b) Use the Windows **Paste** command to clone a copy of the first visual.
 - c) Use the mouse to move the second visual so it appears under the first visual.



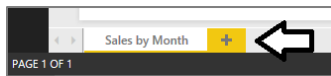
- d) Make sure the bottom visual is selected.
- e) In the Field properties pane for the bottom visual, remove the Gender field from the Legend well and replace with the **Purchase Type** field from the **Purchases** table in the **Fields** list.



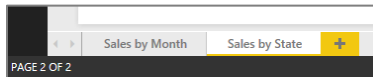
8. Save your work by clicking the **Save** button on the top left of the Power BI Desktop window.

9. Add a second page to the report.

- a) Click the (+) button in the bottom left corner of report view to add a second page to the report.

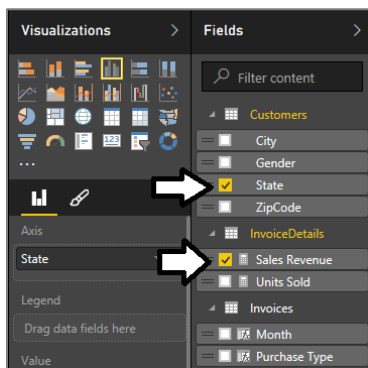


- b) Once the page has been created, modify its title to **Sales by States**.

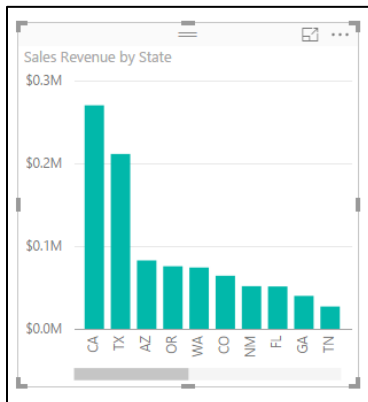


10. Add a new visual to the **Sales by States** page to display numeric data in a matrix.

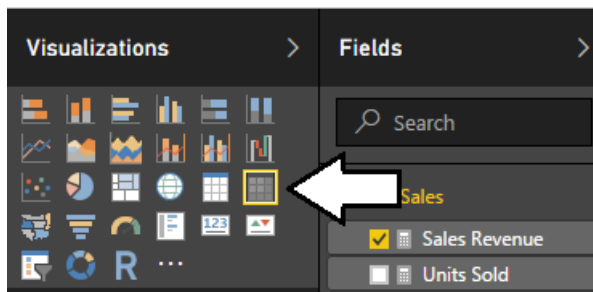
- a) In the **Fields** list, select the **Sales Revenue** measure to create a new clustered column chart visual.
b) Next, select the **State** column.



- c) At this point, your column chart visual should appear like the one shown in the following screenshot.



- d) Change the visual type by clicking the **Matrix** button in the **Visualizations** pane.

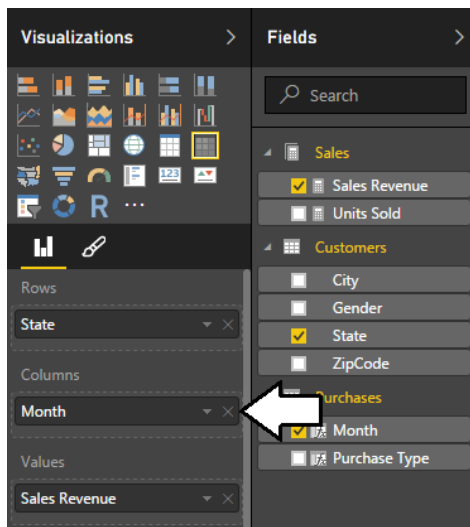


- e) The visual should now display numeric data in a tabular format as shown in the following screenshot.

The screenshot shows a table visual with two columns: 'State' and 'Sales Revenue'. The table lists sales revenue for various states.

State	Sales Revenue
CA	\$270,926.32
TX	\$212,085.08
AZ	\$83,217.79
OR	\$76,153.05
WA	\$74,614.94
CO	\$64,683.47
NM	\$52,066.79
FL	\$51,730.85
GA	\$40,305.80
TN	\$27,443.02
NY	\$24,207.43
UT	\$20,101.20
VA	\$18,568.32
PA	\$16,389.99
CT	\$11,512.65
NC	\$11,018.02
AL	\$8,537.95

- f) In the **Fields** list, locate the **Month** column.
- g) Using the mouse, drag the **Month** field from the **Fields** list and drop it into the **Columns** well in the **Visualizations** pane.

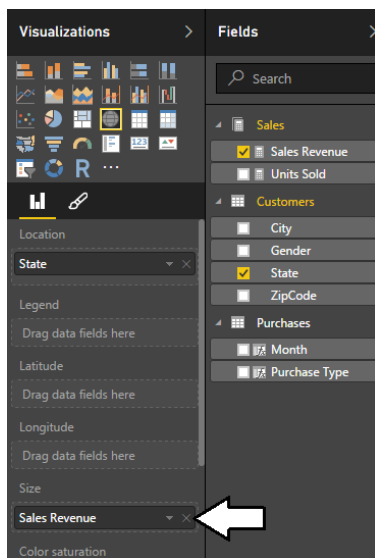


- h) You should now see the visual expand to the right to display a column for each month.
- i) Using the mouse, resize the visual to display all of its contents. Also, position the visual so it occupies the bottom of the page as shown in the following screenshot.

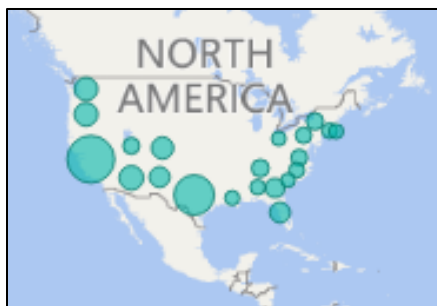
11. Add a second visual to display a map visual which shows how sales revenue is distributed across states.
 - a) Click on the white space below the matrix visual to make sure no visual is selected.
 - b) Select the **State** column in the **Fields** list to create a new visual. Note that Power BI Desktop recognizes the **State** column as a geographic field so it creates the new visual using the **Map** visual type.



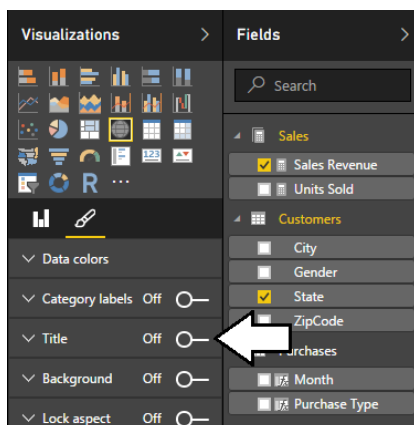
- c) Using the mouse, drag the **Sales Revenue** measure from the **Fields** list into the **Values** well in the **Visualizations** pane.



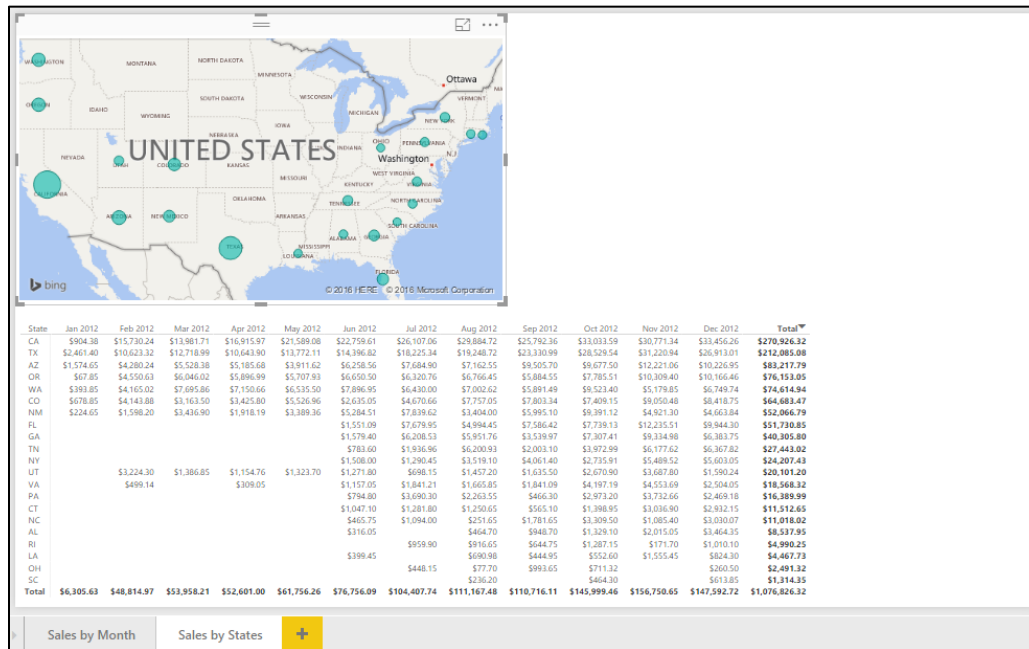
- d) The Map visual should now show dots of different sizes to visually depict how sales revenue is distributed across states.



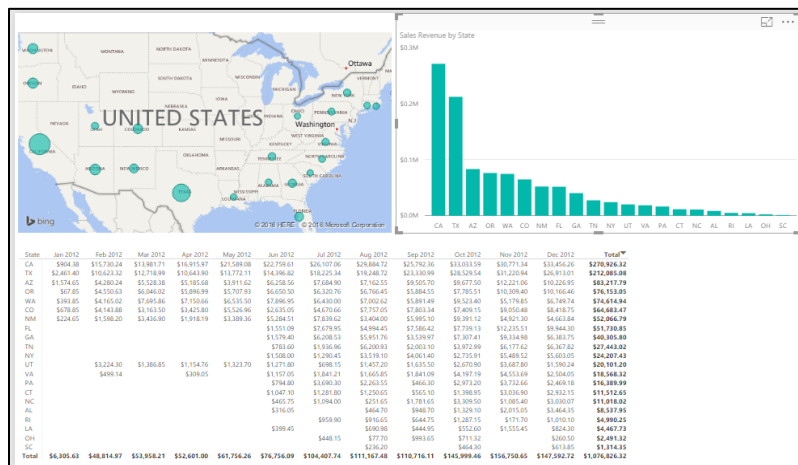
- e) With the Map visual selected, locate the **Title** property in the **Format** property pane and update its value from **On** to **Off**.



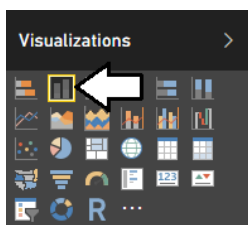
- f) Using the mouse, rearrange the visuals on the page to match the following screenshot.



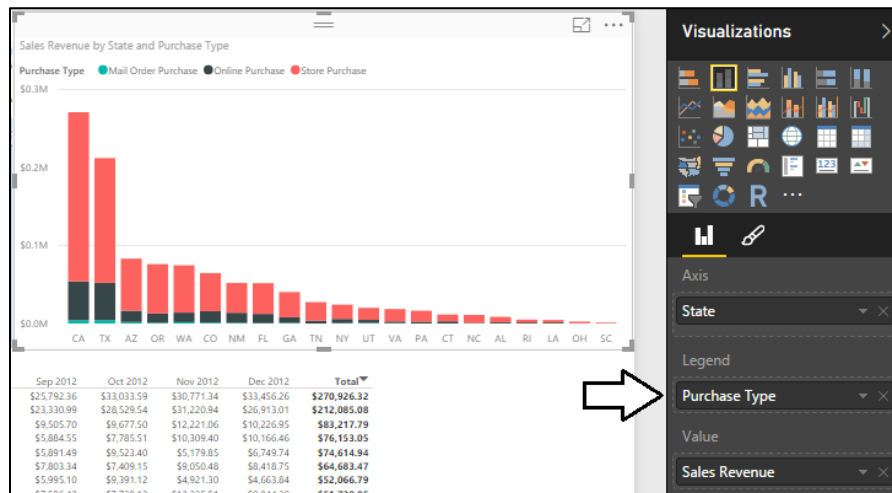
12. Add a third visual to the page to display sales revenue by state and purchase type.
 - a) Click on the white space in the bottom right corner of the page to make sure no visual is selected.
 - b) Select the **Sales Revenue** measure in the **Fields** list to create a new clustered column chart visual.
 - c) Select the **State** column in the **Fields** list to add a set of axis value to the column chart.
 - d) Using the mouse, rearrange the visuals on the page to match the following screenshot.



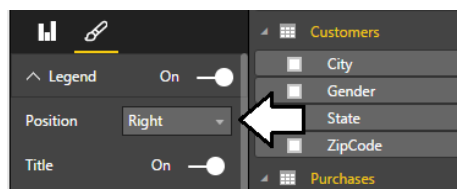
- e) Change the visual type from a cluster column chart to a stacked column chart.



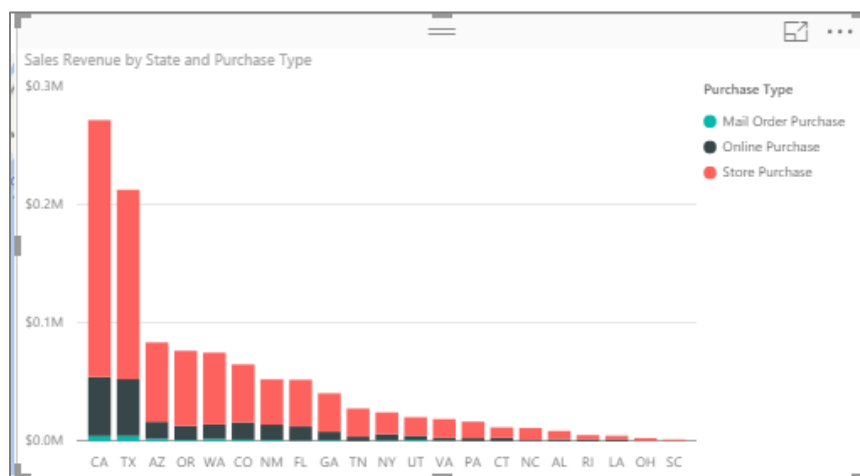
- f) Using the mouse, drag the **Purchase Type** column from the **Fields** list to the **Legend** well in the **Visualizations** pane.



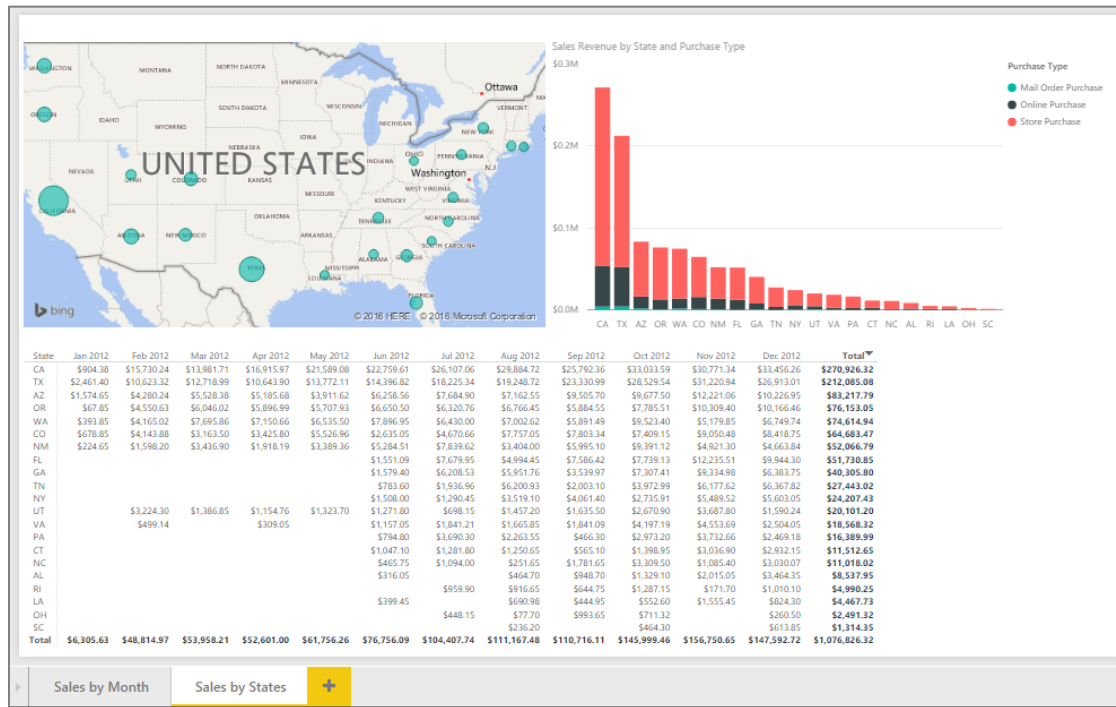
- g) Locate the **Legend** section in the **Format** properties pane and set the legend **Position** property to **Right**.



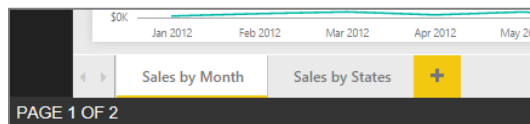
- h) The visual should now match the following screenshot.



- i) At this point, you have now finished designing the **Sales by State** page and your report should match the following screenshot..



13. Click the **Sales by Month** page tab to make that the active page.



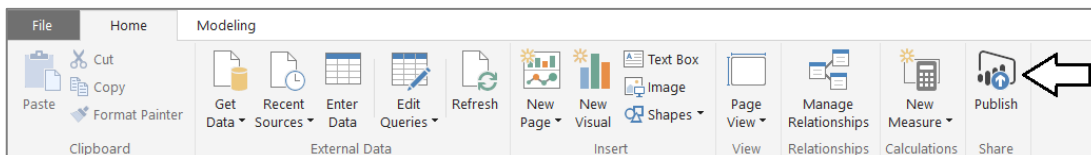
14. Save your work by clicking the **Save** button on the top left of the Power BI Desktop window.

Think about why it's important to make the **Sales by Month** page the active page before saving this report. That's because the active page is saved along with the report definition and this page will be shown first when the report is accessed in the Power BI service.

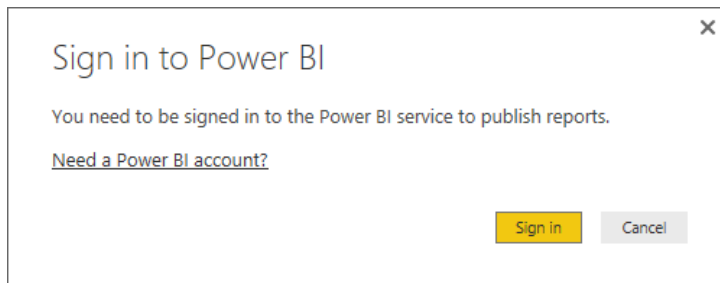
Exercise 5: Publishing a PBIX File to the Power BI Service

In this exercise you will complete your work by publishing the PBIX file to your personal workspace in the Power BI service.

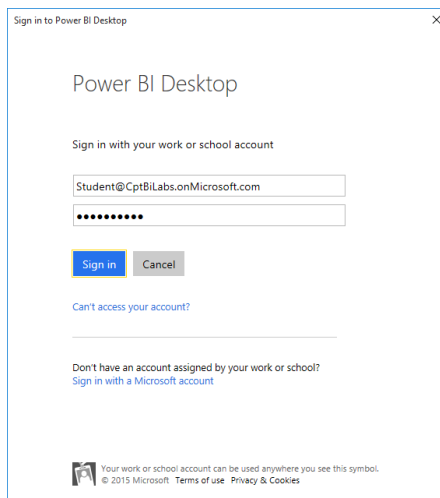
1. Make sure you still have the **Wingtipsales2012.pbix** file open that you created in the previous exercise.
2. Click the **Publish** button on the far right-hand side of the ribbon.



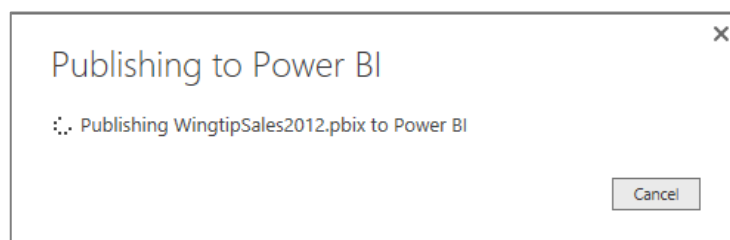
3. When prompted with the **Sign in to Power BI** dialog, click the **Sign In** button



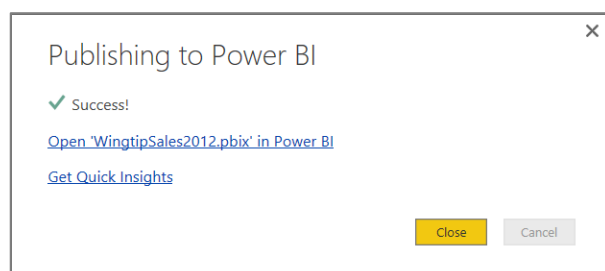
4. Sign into the Power BI service using your primary Office 365 account to give Power BI Desktop the access to publish the PBIX file.



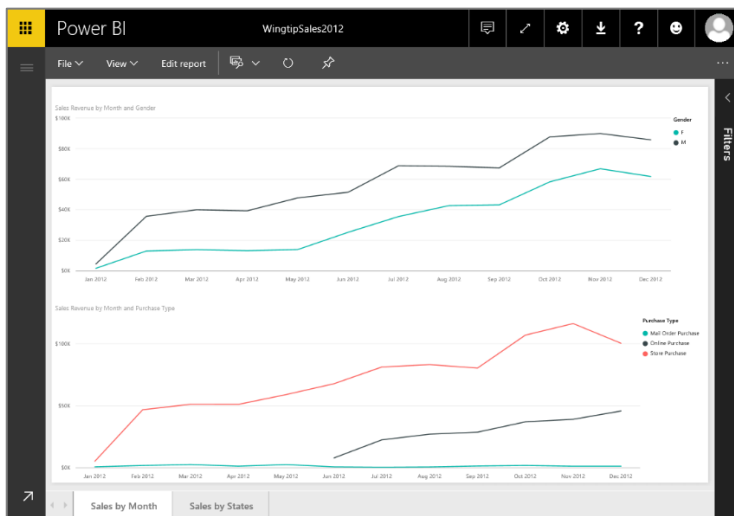
5. After you have signed in, Power BI Desktop will display the **Publishing to Power BI** dialog showing you that the publishing process is underway.



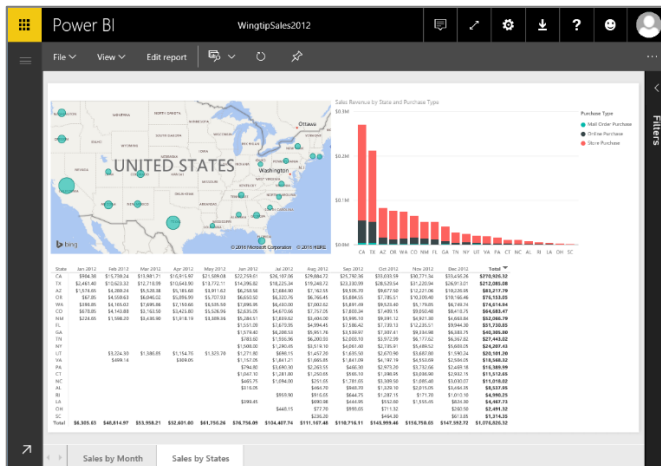
6. Once the publishing process has completed, the **Publishing to Power BI** dialog will display a success message and provide you with a link to **Open WingtipSales2012.pbix in Power BI**. Click on that link to navigate to the Power BI service using the browser.



7. You should now be able to see the **Sales by Month** page of the report you just created.

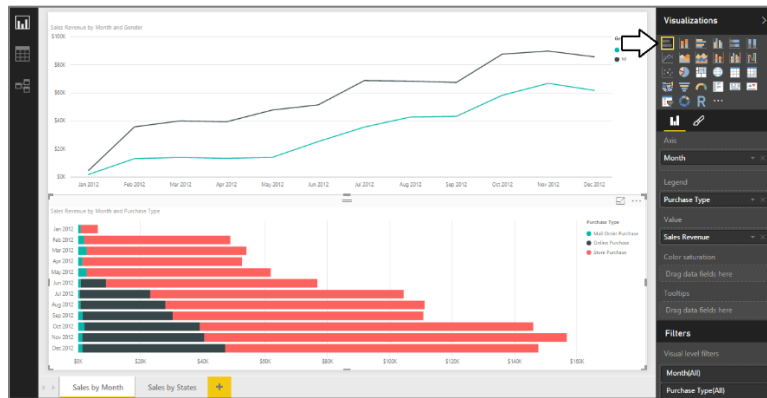


8. Click on the **Sales by State** link at the bottom of the screen to see the second page of the report.



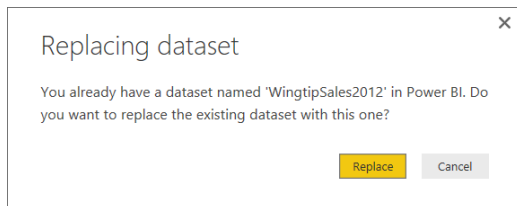
You have now successfully created and published a simple BI solution using Power BI Desktop. But what happens when you want to make a change to a report after it has been published? It's very easy because you can make changes to your Power BI Desktop project and republish it on top a previous version of the same project that has already been published.

9. Change the type of the visual that displays sales revenue by month and purchase type.
 - a) Return back to Power BI Desktop and make sure you are in report view.
 - b) Return to the Sales by Month page.
 - c) Select the bottom visual and change its visual type to be a stacked bar chart.

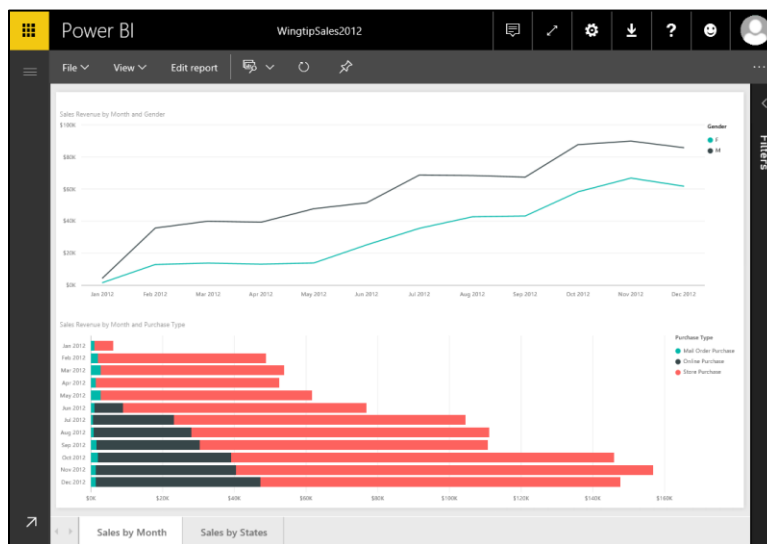


10. Republish the project to the Power BI service.

- Save the project by clicking the **Save** button in the ribbon.
- Click the **Publish** button on the far right-hand side of the **Home** tab in the ribbon.
- When prompted with the **Replacing dataset** dialog, click **Replace** to begin the publishing process.



- Once the publishing process has completed, inspect the published report in the Power BI service using the browser.



Congratulations, you have now finished this lab. If you finish early before other student and you still have extra time, experiment by clicking the **Edit report** button in the browser and seeing how you can continue to modify the pages of the report after the report has been published to the Power BI service. Note that any changes you make to the report will be overwritten if you republish the report with Power BI Desktop.