Microsoft

Power BI

PARTNER PRACTICE
ENABLEMENT
BOOTCAMP

Lab 05A

Creating a Power BI Desktop Solution

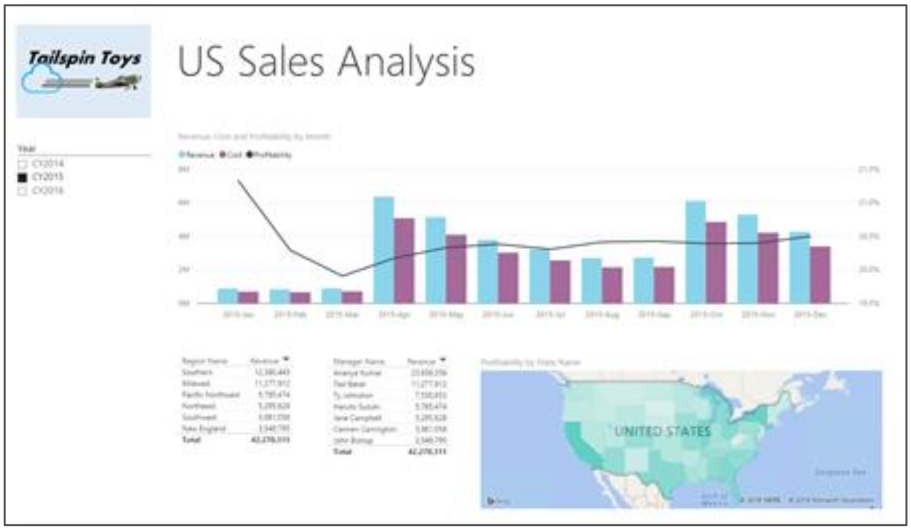
Overview

The estimated time to complete the lab is 75 minutes

In this lab, you will create a Power BI Desktop solution to enable the reporting and analysis of US sales activity. This will involve creating Power BI Desktop queries that source data from SQL Server, an Excel workbook and a CSV file.

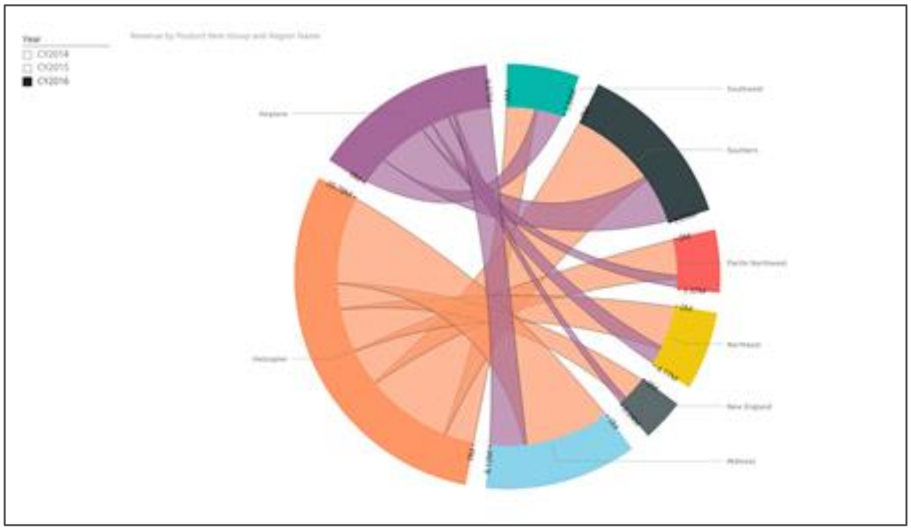
You will then create two report pages consisting of various data visualizations, including a visual downloaded from the Power BI Visual Gallery.

The final report pages will look like the following.



Region Name	Revenue
Southern	12,380,443
Midwest	11,277,912
Pacific Northwest	5,785,474
Northeast	5,295,628
Southwest	3,981,058
New England	3,546,785
Total	**42,270,311**

Manager Name	Revenue
Ananya Kumar	23,658,356
Tad Baker	11,277,912
Ty Johnston	7,530,855
Haruto Suzuki	5,785,474
Jane Campbell	5,295,628
Carmen Carrington	3,981,058
John Bishop	3,546,785
Total	**42,270,311**



You will learn how to:

- Create a Power BI Desktop solution
- Create queries based on a variety of data sources
- Prepare a model for reporting
- Create an interactive dashboard layout consisting of several data visualizations
- Download and import a visual from the Power BI Visual Gallery

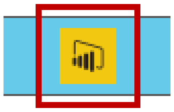
Creating Power BI Queries

In this exercise, you will create five Power BI Desktop queries. Four queries will source data from SQL Server, an Excel workbook and a CSV file. A fifth query will generate date data without requiring a data source.

Creating a Power BI Desktop File

In this task, you will create a Power BI file.

1. To open the Power BI Desktop, on the taskbar, click the **Microsoft Power BI Desktop** shortcut.



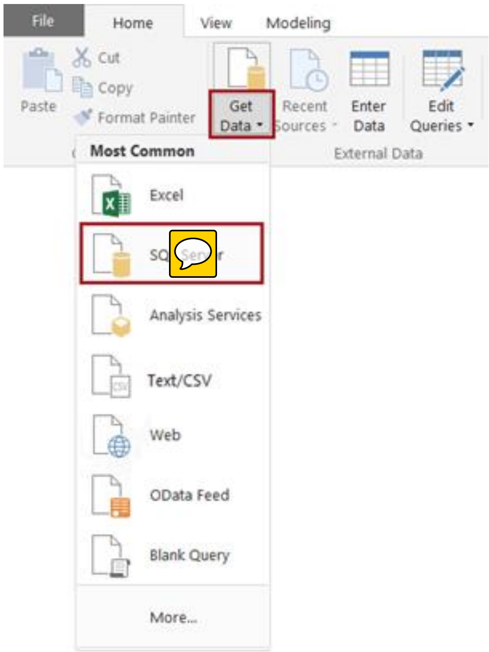
2. To close the startup screen, at the top-right corner, click **X**.

3. To save the file, click the **File** tab, and then select **Save As**.
4. In the **Save As** window, navigate to the **D:\PowerBI\Lab02** folder.
5. In the **File Name** box, enter **US Sales Analysis**.
6. Click **Save**.

Importing Data from SQL Server

In this task, you will create a query to retrieve sales order data from the **TailspinToys-US** SQL Server database.

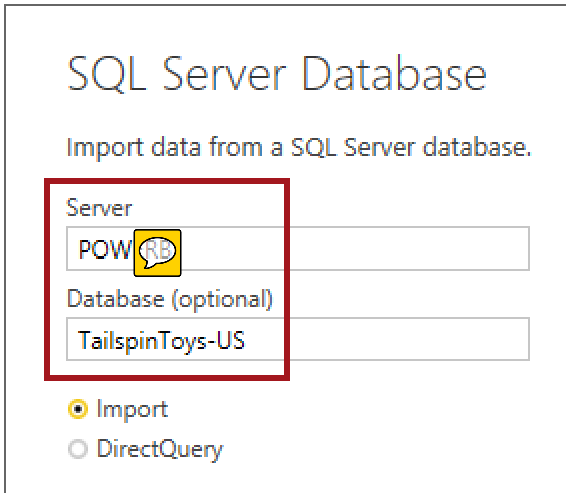
1. On the **Home** ribbon, from inside the **External Data** group, click the **Get Data** dropdown, and then select **SQL Server**.



2. In the **SQL Server Database** dialog window, in the **Server** box, enter **POWERBI**.

   ***POWERBI** is the name of the local machine.*

3. In the **Database** box, enter **TailspinToys-US**.



SQL Server Database

Import data from a SQL Server database.

Server

POWERBI

Database (optional)

TailspinToys-US

● Import

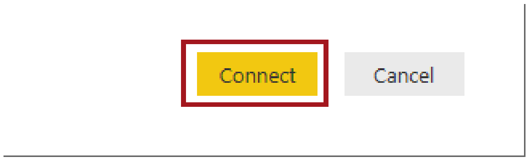
○ DirectQuery

4. Notice that the default option is to import (cache) the data.

   It is also possible to connect directly to the SQL Server data source by using DirectQuery. In this mode, no data is imported or copied into Power BI Desktop. The selected tables and columns *appear in your Power BI Desktop Fields list. As you create or interact with a visualization, Power BI Desktop queries the underlying data source, which means you're always viewing current data.*

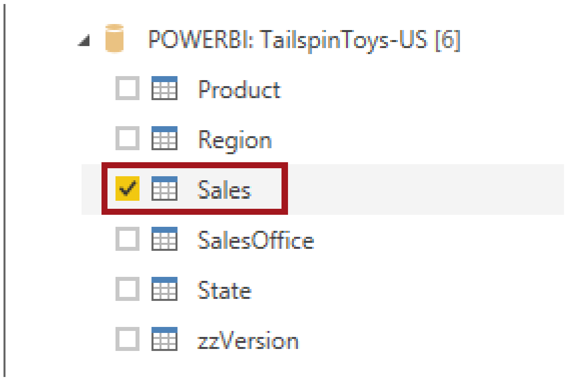
5. Click **OK**.

6. When prompted to authenticate, to use your Windows credentials, click **Connect**.



7. When prompted to confirm the use of an unencrypted connection, click **OK**.

8. In the **Navigator** dialog window, check the **Sales** table.

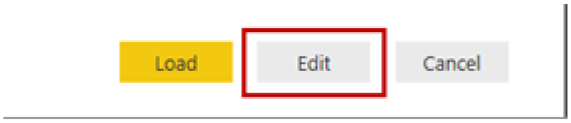

POWERBI: TailspinToys-US [6]

- □ Product
- □ Region
- ☑ Sales
- □ SalesOffice
- □ State
- □ zzVersion

9. Review the data in the preview pane (located at the right).

   The data represents the US retail sales operations, including the last two columns which are foreign key columns that relate to the ***Product*** *and* ***State*** *tables. In this lab, you will develop the query by introducing columns from these related tables, and by defining friendly names for the columns.*

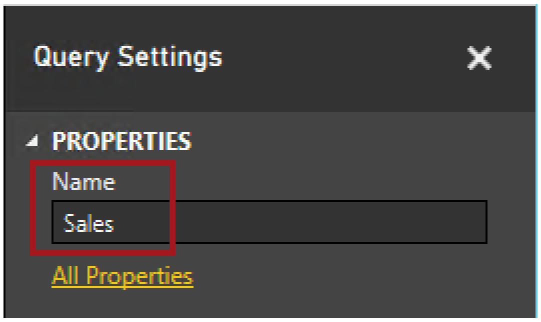
10. To develop the query, click **Edit**.



11. Notice that the **Query Editor** window opens, and that this window has its own ribbon.

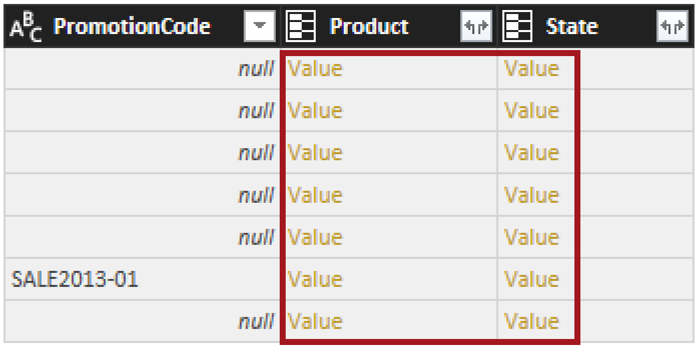
    This window is used to define query steps to transform data, and to preview the query result.

12. In the **Query Settings** pane (located at the right), in the **Name** box, notice that the query name was derived from the selected source table: **Sales**.



Name
Sales

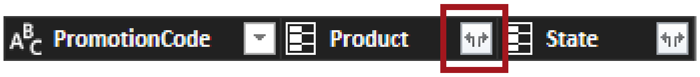
13. In the data pane (the large pane containing the data grid), notice that the **Product** and **State** columns (the last two columns) contain **Value** links, enabling the introduction of columns from the related tables.



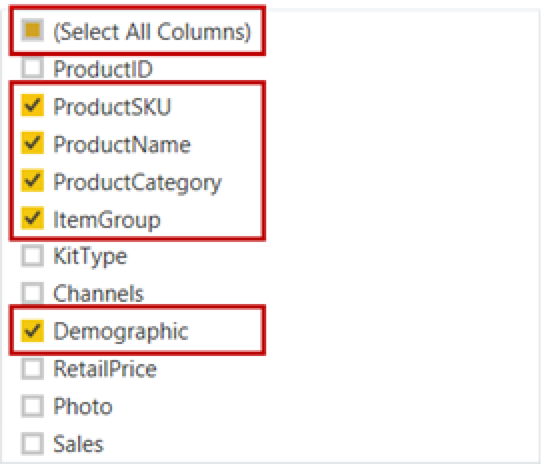
PromotionCode	Product	State
null	Value	Value
null	Value	Value
null	Value	Value
null	Value	Value
null	Value	Value
SALE2013-01	Value	Value
null	Value	Value

*These columns were added automatically because the **CustomerStateID** and **ProductID** columns of the **Sales** table are foreign key columns.*

14. To introduce columns from the **Product** table, in the **Product** column header, click **Expand**.

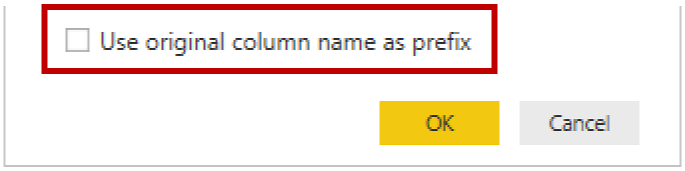


15. Uncheck **(Select All Columns)**, and then check only the following five columns.



- ■ (Select All Columns)
- □ ProductID
- ☑ ProductSKU
- ☑ ProductName
- ☑ ProductCategory
- ☑ ItemGroup
- □ KitType
- □ Channels
- ☑ Demographic
- □ RetailPrice
- □ Photo
- □ Sales

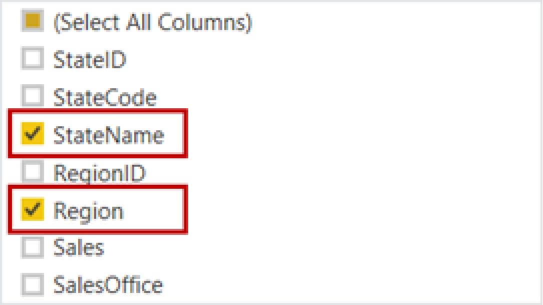
16. Uncheck the **Use Original Column Name as Prefix** checkbox.



☐ Use original column name as prefix

17. Click **OK**.

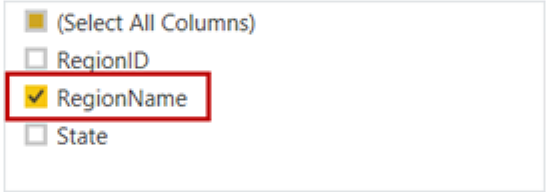
18. Expand the **State** column to include only the **StateName** and **Region** columns.



■ (Select All Columns)
☐ StateID
☐ StateCode
☑ StateName
☐ RegionID
☑ Region
☐ Sales
☐ SalesOffice

19. Expand the **Region** column to include only the **RegionName** column.

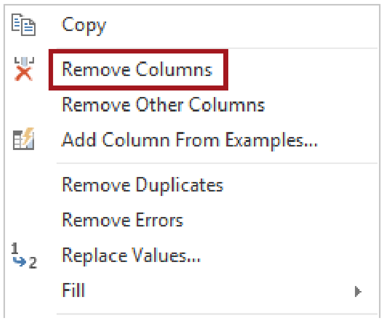
*The **Region** column represents an additional table, related to the **State** table.*



■ (Select All Columns)
☐ RegionID
☑ RegionName
☐ State

20. To remove unnecessary columns, first select the **OrderNumber** column header (first column), and then while pressing the **Control** key, select also the **ShipDate**, **CustomerStateID**, **ProductID**, **DiscountAmount**, and **PromotionCode** column headers.

21. Right-click the column selection, and then select **Remove Columns**.



22. To rename the column, right-click the **UnitPrice** header column, and then select **Rename**.

Tip: It is also possible to rename the column by double-clicking the column header.

23. Modify the name to **Unit Price**, and then press **Enter**.

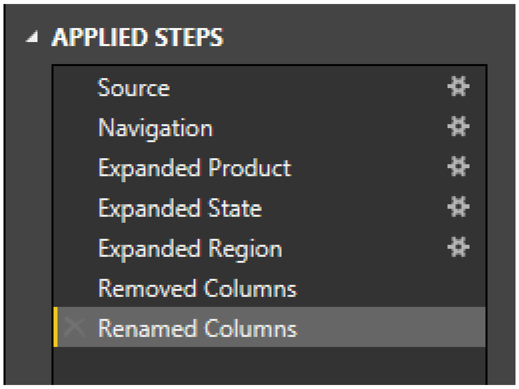
    Defining concise—yet friendly names—ensures that data visualization captions are easy to understand.

    In addition, while it is possible to name a column the same name as the table it belongs to, it is important to use column names that differ. This will avoid ambiguity when using Q&A (natural language querying) once the Power BI Desktop file is published to the Power BI service.

24. Rename also the following seven columns.

Old Column Name	New Column Name
ProductSKU	Product SKU
ProductName	Product Name
ProductCategory	Product Category
ItemGroup	Product Item Group
Demographic	Product Demographic
StateName	State Name
RegionName	Region Name

25. In the **Query Settings** pane, notice the applied steps that define the logic to source and transform the query result.



APPLIED STEPS

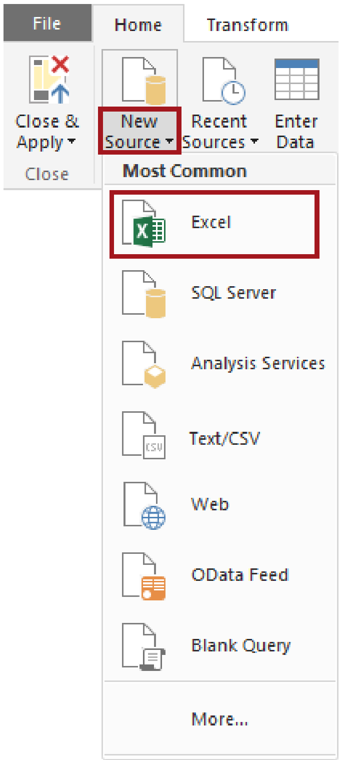
- Source
- Navigation
- Expanded Product
- Expanded State
- Expanded Region
- Removed Columns
- Renamed Columns

Later in this lab, you will combine the result of this query with another query to extend it with cost data to enable profit analysis.

Importing Data from an Excel Workbook

In this task, you will create a second query, this time to retrieve product cost data from an Excel workbook.

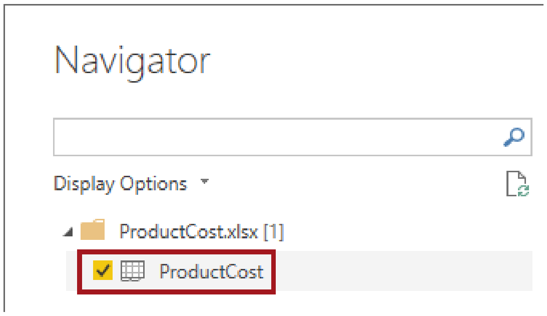
1. In the **Query Editor** window, on the **Home** ribbon, from inside the **New Query** group, click the **New Source** dropdown, and then select **Excel**.



2. In the **Open** window, navigate to the **D:\PowerBI\Lab02\Assets** folder.

3. Select the **ProductCost.xlsx** file, and then click **Open**.

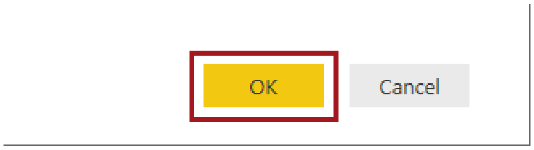
4. In the **Navigator** dialog window, check the **ProductCost** worksheet.



5. Review the data in the preview pane.

   *The data represents the cost of each product. Product cost data is stored externally from the SQL Server sales database, and will be integrated with the **Sales** query (created in the previous task) to enable the calculation of profit.*

6. To load the query, click **OK**.



7. In the **Query Editor** window, in the **Query Settings** pane, in the **Applied Steps** list, notice that four steps exist, applying default transformations to the Excel data.

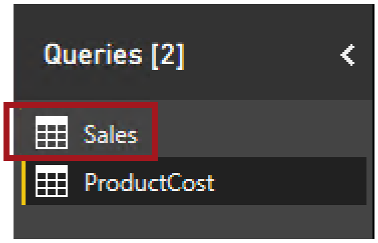
8. In the grid, remove the **Product** column.

   *The **Product** column is not required as this data has already been retrieved by the **Sales** query.*

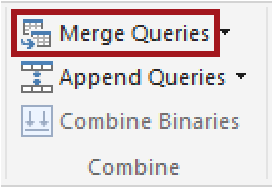
Combining Queries

In this task, you will merge the **Sales** query with the **ProductCost** query in order to create the **Cost** column.

1. In the **Queries** pane (located at the left), select the **Sales** query.



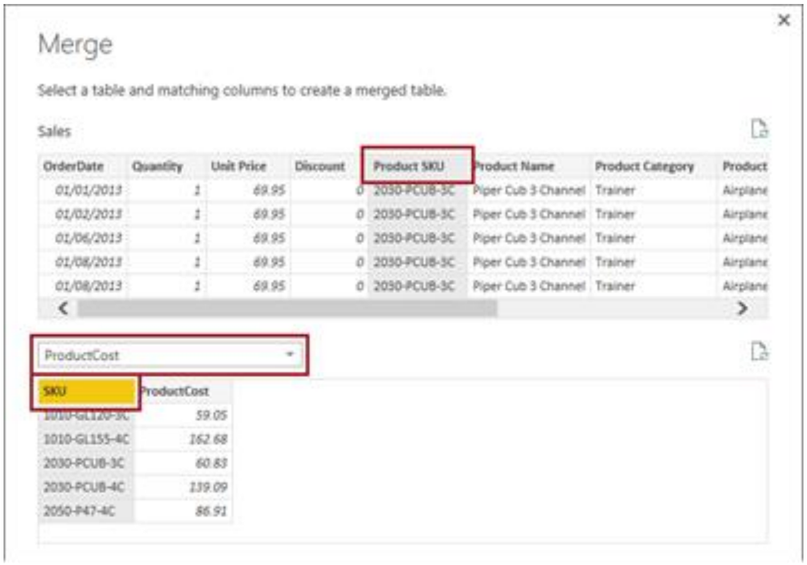
2. On the **Home** ribbon, from inside the **Combine** group, click **Merge Queries**.



3. In the **Merge** dialog window, in the data grid, select the **Product SKU** column header.

4. In the dropdown list, select the **ProductCost** query.

5. In the lower data grid, select the **SKU** column header.



Merge

Select a table and matching columns to create a merged table.

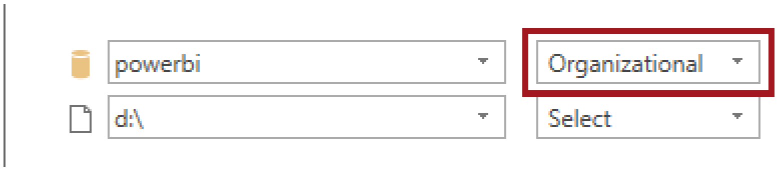
Sales

OrderDate	Quantity	Unit Price	Discount	Product SKU	Product Name	Product Category	Product
01/01/2013	1	69.95	0	2030-PCUB-3C	Piper Cub 3 Channel	Trainer	Airplane
01/02/2013	1	69.95	0	2030-PCUB-3C	Piper Cub 3 Channel	Trainer	Airplane
01/06/2013	1	69.95	0	2030-PCUB-3C	Piper Cub 3 Channel	Trainer	Airplane
01/08/2013	1	69.95	0	2030-PCUB-3C	Piper Cub 3 Channel	Trainer	Airplane
01/08/2013	1	69.95	0	2030-PCUB-3C	Piper Cub 3 Channel	Trainer	Airplane

ProductCost

SKU	ProductCost
1010-GL120-3C	59.05
1010-GL155-4C	162.68
2030-PCUB-3C	60.83
2030-PCUB-4C	119.09
2050-P47-4C	86.91

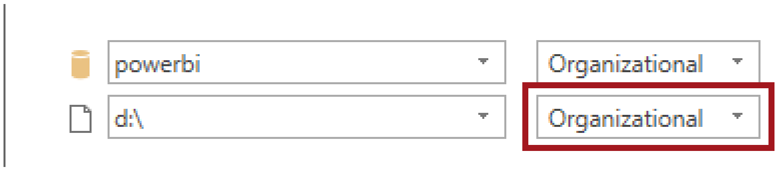
6. In the **Privacy Levels** dialog window, for the **powerbi** data connection, in the adjacent dropdown list, select **Organizational**.



powerbi	Organizational
d:\	Select

*A privacy level specifies an isolation level that defines the degree that one data source will be isolated from other data sources. An **Organizational** data source limits the visibility of a data source to a trusted group of people, and includes other organizational data sources.*

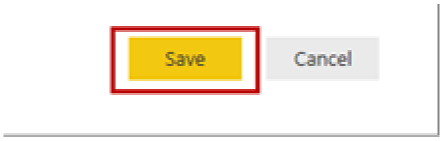
7. Configure the privacy level for the **d:\** file location to **Organizational** also.



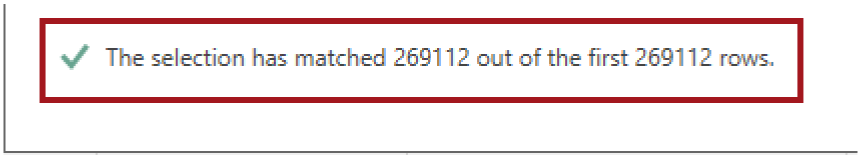
powerbi	Organizational
d:\	Organizational

*As it now stands, the **Sales** query can be combined with the **ProductCost** query.*

8. Click **Save**.

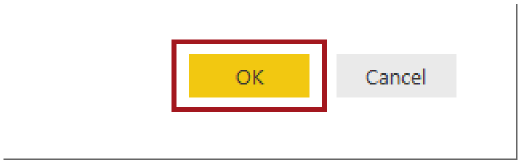


9. Located at the bottom-left corner of the **Merge** dialog window, notice that all rows have matched.



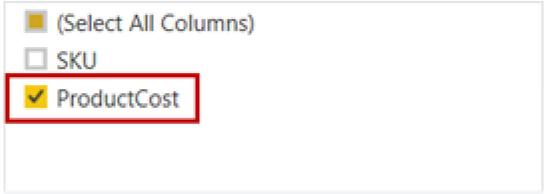
✓ The selection has matched 269112 out of the first 269112 rows.

10. Click **OK**.



11. In the data pane, at the end of the columns, notice the addition of a new column named **NewColumn**.

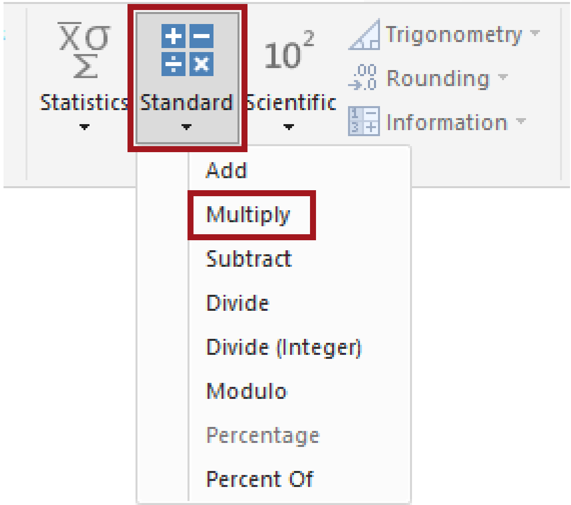
12. Expand the **NewColumn** column to include only the **ProductCost** column.



■ (Select All Columns)
☐ SKU
☑ ProductCost

13. To calculate the cost of each sales order, first select the **Quantity** column header, and then while pressing the **Control** key, select the **ProductCost** column header.

14. On the **Add Column** ribbon, from inside the **From Number** group, click **Standard**, and then select **Multiply**.



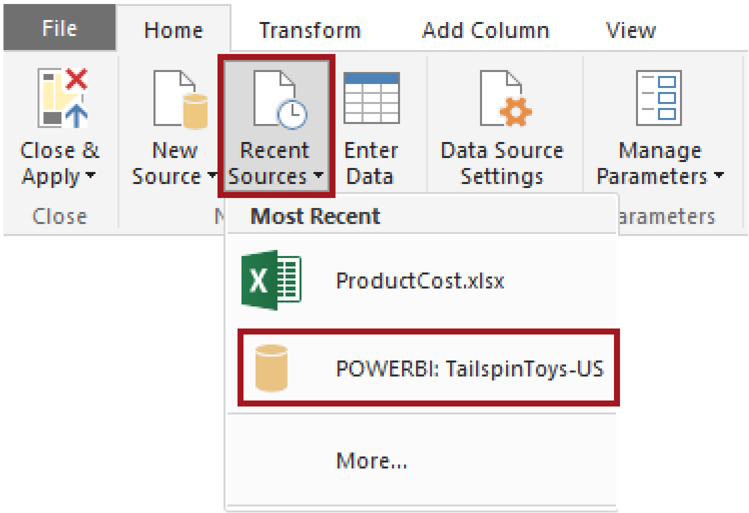
15. Rename the **Inserted Multiplication** column as **Cost**.

16. Remove the **ProductCost** column.

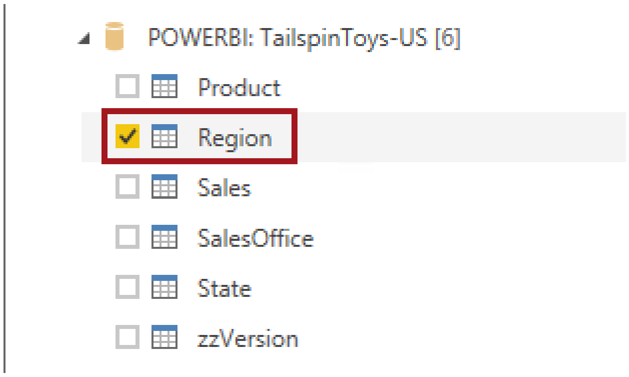
Importing Additional Data from SQL Server

In this task, you will create an additional query from the **TailspinToys US** SQL Server database to retrieve **Region** data.

1. On the **Home** ribbon, from inside the **New Query** group, click **Recent Sources**, and then select **POWERBI: TailspinToys-US**.



2. In the **Navigator** dialog window, check the **Region** table.

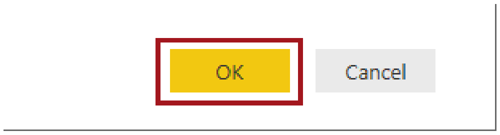


3. Review the data in the preview pane.

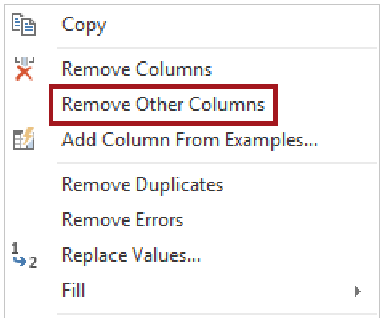
   The data represents all regions, and this data will be used to add a query that will support a many-to-many relationship between sales managers (who can be jointly responsible for multiple regions) and sales.

   The manager data will be loaded in the next task.

4. To load the query, click **OK**.



5. Right-click the **RegionName** column header, and then select **Remove Other Columns**.

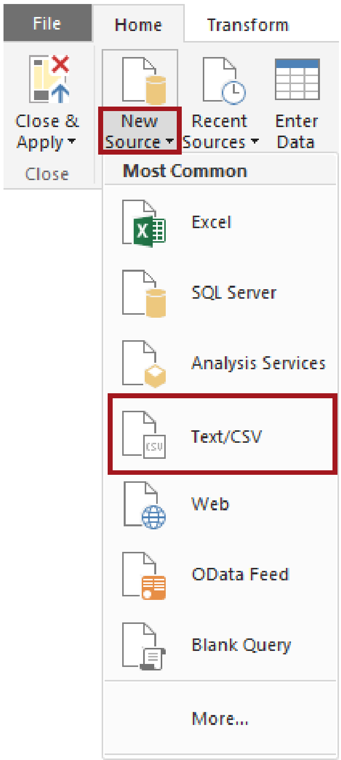


6. Rename the **RegionName** column as **Region Name**.

Importing Data from a CSV File

In this task, you will create a query from a CSV file to retrieve manager data.

1. In the **Query Editor** window, on the **Home** ribbon, from inside the **New Query** group, click the **New Source** dropdown, and then select **Text/CSV**.

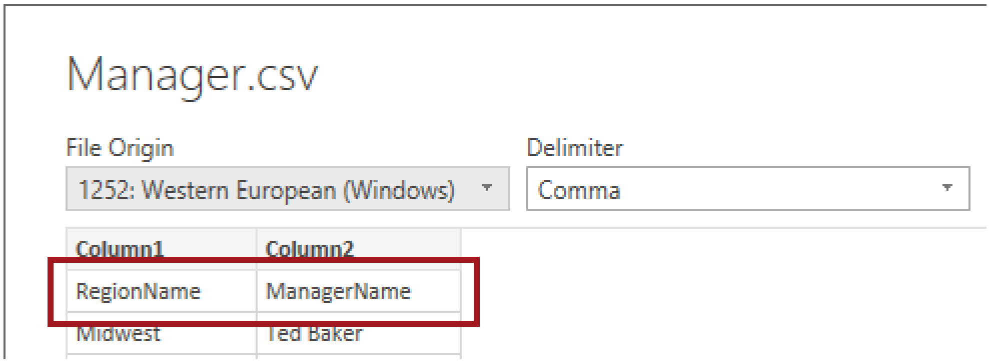


2. In the **Open** window, navigate to the **D:\PowerBI\Lab02\Assets** folder.

3. Select the **Manager.csv** file, and then click **Open**.

   The data represents sales managers and their assignment to one, or possibly more, sales regions.

4. In the **Manager.csv** dialog window, notice that the first row contains the column names.



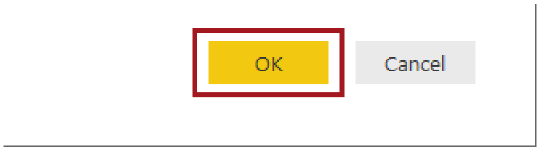
Manager.csv

File Origin: 1252: Western European (Windows)

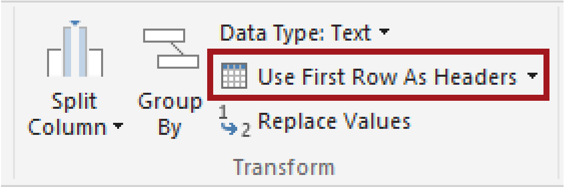
Delimiter: Comma

Column1	Column2
RegionName	ManagerName
Midwest	Ted Baker

5. Click **OK**.

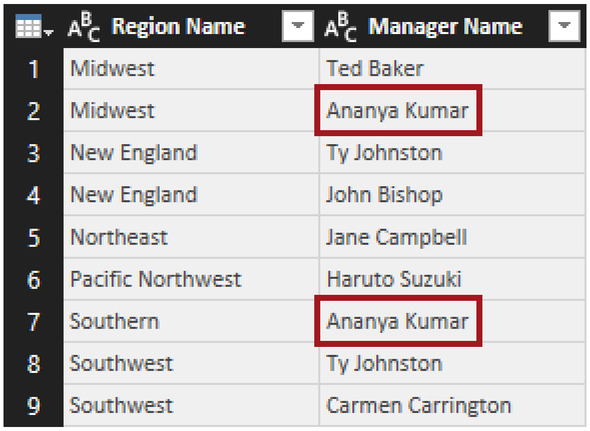


6. To promote the first row values as column headers, on the **Home** ribbon, from inside the **Transform** group, click **Use First Row as Headers**.



7. Rename the columns as **Region Name**, and **Manager Name**.

8. Notice that some sales regions have two managers assigned, for example **Ananya Kumar** is assigned to both the **Midwest** and **Southern** sales regions.



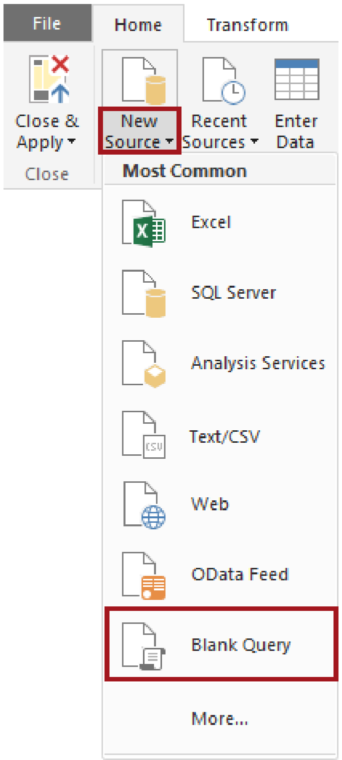
	Region Name	Manager Name
1	Midwest	Ted Baker
2	Midwest	Ananya Kumar
3	New England	Ty Johnston
4	New England	John Bishop
5	Northeast	Jane Campbell
6	Pacific Northwest	Haruto Suzuki
7	Southern	Ananya Kumar
8	Southwest	Ty Johnston
9	Southwest	Carmen Carrington

Analyzing sales statistics by sales manager will be achieved by a many-to-many relationship that you will configure in the next exercise.

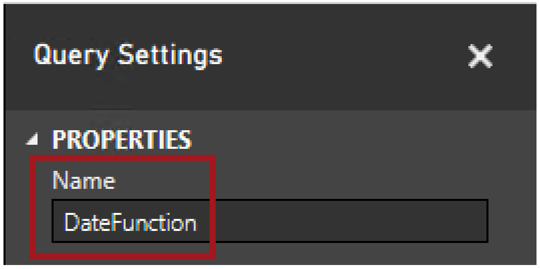
Creating Data with the Advanced Editor

In this task, you will create a query to generate date data by using a pre-developed script.

1. In the **Query Editor** window, on the **Home** ribbon, from inside the **New Query** group, click the **New Source** dropdown, and then select **Blank Query**.



2. In the **Query Editor** window, in the **Query Settings** pane, in the **Name** box, replace the text with **DateFunction**, and then press **Enter**.

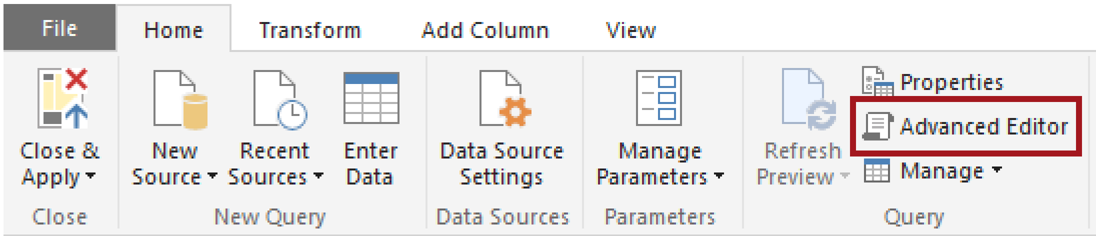

Query Settings

PROPERTIES

Name

DateFunction

3. To define the query, on the **Home** ribbon, from inside the **Query** group, click **Advanced Editor**.



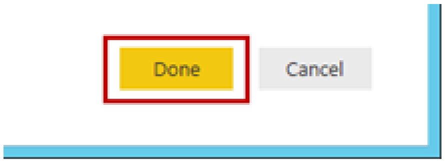
4. In the **Advanced Editor** window, to remove all text from inside the query box, click inside the box, press **Control+A**, and then press the **Delete** key.

5. Use Notepad to open the **D:\PowerBI\Lab02\Assets\DateFunctionScript.txt** file, and then copy-and-paste the entire content inside the query box.

   The script defines a function that accepts start and end dates, and an optional culture. When invoked, this function will produce a query result defining date data.

   The script has been adapted from a blog post published by Matt Masson titled [Creating a Date Dimension with a Power Query Script.]()

6. Click **Done**.

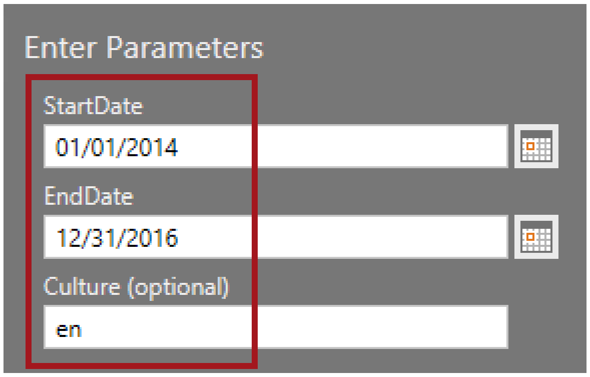


7. In the **Enter Parameters** region, enter the following values.

   The date values must be entered in US format (mm/dd/yyyy) as the lab virtual machine regional settings are set to United States.

   The culture value can be set to any valid culture (e.g. fr, fr-FR, es-ES, etc.) and is used to produce localized month names. You may enter the culture code for your region.

   Culture names are documented at MSDN within the [National Language Support (NLS) API Reference]().

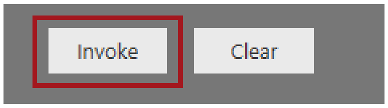

Enter Parameters

StartDate: 01/01/2014

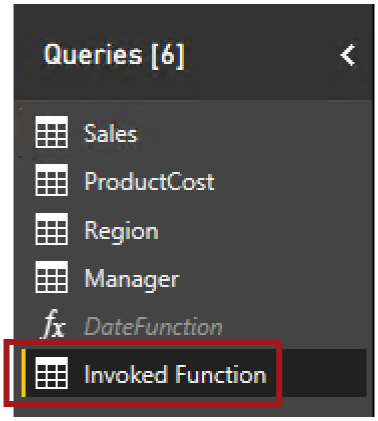
EndDate: 12/31/2016

Culture (optional): en

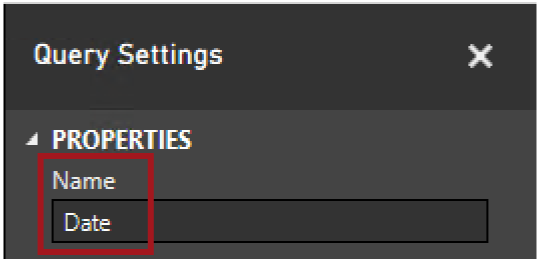
8. To invoke the function, click **Invoke**.



9. In the **Queries** pane, notice that invoking the function created a new query.

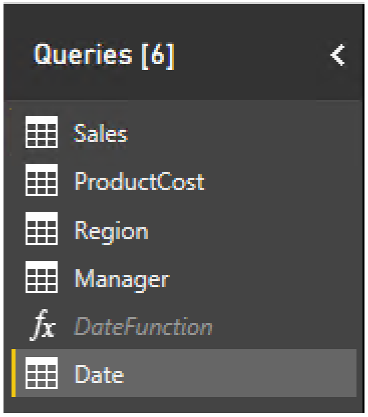


10. Rename the new query as **Date**.

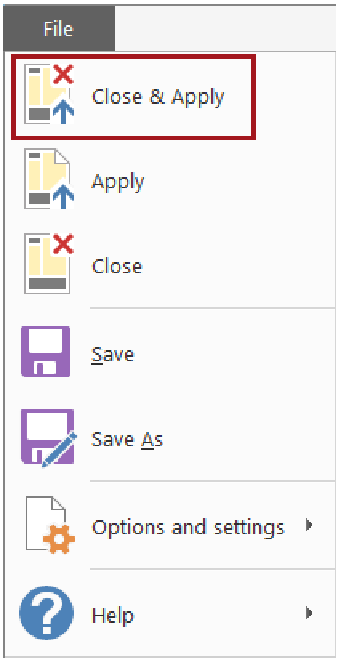


Name
Date

11. Verify that you now have defined six queries.



12. To close the **Query Editor** window, on the **File** menu, select **Close & Apply**.



Applying changes loads data and detect relationships. Note that data will only be loaded if the data has not already been loaded, or query definitions have been changed.

Preparing the Model for Reporting

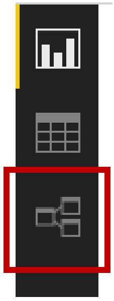
In this exercise, you will configure the model relationships to connect all five queries, hide tables and columns that are not appropriate to use in reports. In addition, you will sort columns, and create a calculated column and two measures to enable profit analysis.

Note that once loaded to the report, queries are then referred to as tables.

Managing Relationships

In this task, you will manage the model relationships to connect all queries.

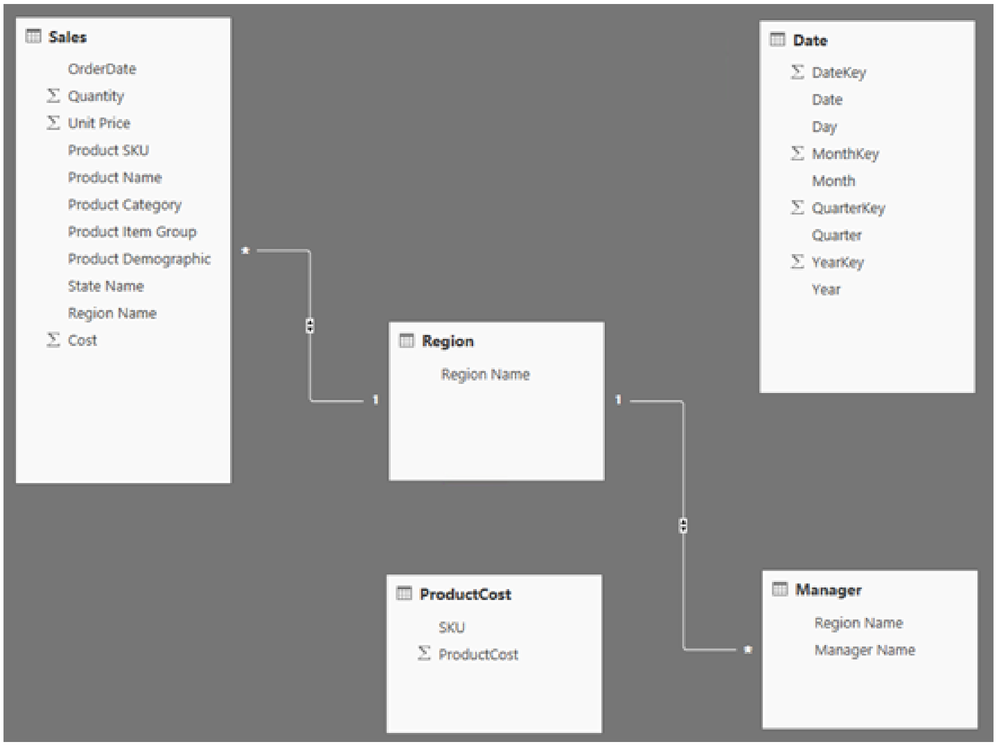
1. To switch to Relationships view, at the left side, click **Relationships**.



2. Notice that two relationships were automatically detected and created.

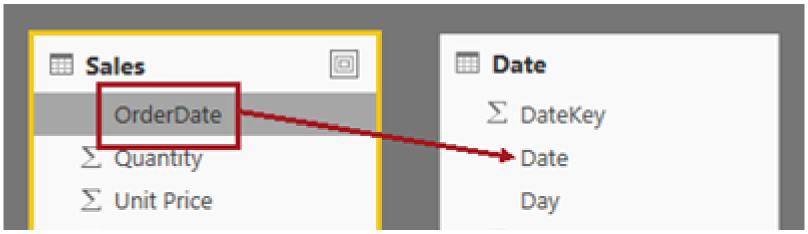
   Relationships can be automatically detected when column names match, and also their data types. The two relationships enable many-to-many analysis between sales managers and sales.

3. Reposition and resize the tables to enable a clearer understanding of the model.



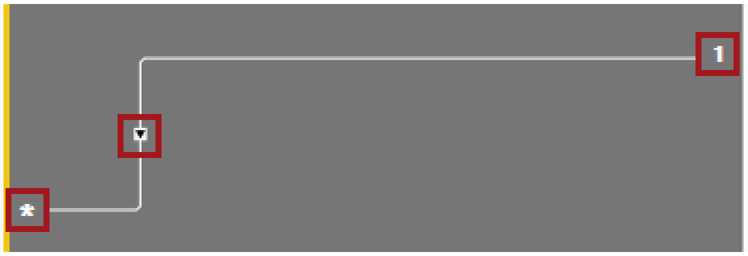
4. To create a relationship between the **Sales** and **Date** tables, from inside the **Sales** table, drag the **OrderDate** column, and then drop it onto the **Date** column of the **Date** table.

   It is a good practice to drag from the many side to the one side.



5. Notice the addition of the relationship line, and the indicators describing the many side (*), one side (1), and the cross filter direction.

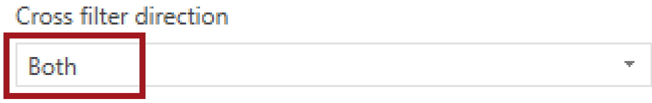
   *The cross filter direction describes how filters are propagated. In this case, a filter applied to the **Date** table, for example for a particular year, would filter the rows in the **Sales** table with an order date in that year.*



6. Notice then, that the automatically detected relationships between the **Sales**, **Region** and **Manager** tables, define a many-to-many relationship, with the **Region** table as the intermediary table, and with cross-filters defined for a single direction.

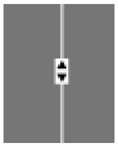
7. To edit the relationship, double-click the relationship between the **Region** and the **Manager** tables.

8. In the **Edit Relationship** window, in the **Cross Filter Direction** dropdown list, select **Both**.


Cross filter direction
Both

9. Click **OK**.

10. Verify that the relationship has a double-headed arrow, indicating the cross-filter directions.



11. In the **Date** table, notice that four columns are adorned with the sigma (∑) symbol.

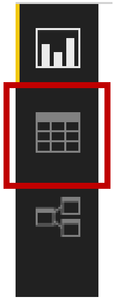
    These four columns contain numeric data, and the symbol means that the column values may be summarized in reports.

    As these four columns contain key values used to configure the chronological sorting of date label columns, it is not appropriate for these columns to be available for users to access. These columns will be hidden in the next task.

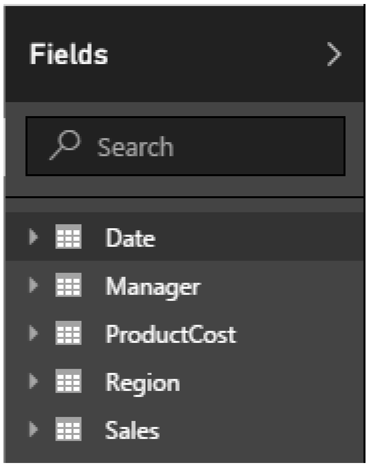
Hiding Tables and Fields

In this task, you will hide tables and fields that are not appropriate for use in reports.

1. Switch to Data view.



2. Notice the **Fields** pane (located at the right), which allows the selection of a table to view its data.

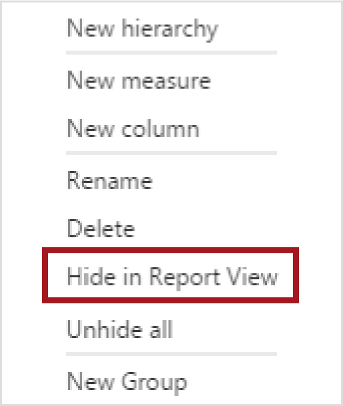


3. In the **Fields** pane, expand the **Date** table.

4. In the **Date** table, notice the four “key” fields.

   The “key” fields enable sorting related fields in the table. It is not appropriate that these fields are used in reports, and so each will be hidden.

5. From inside the **Date** table, right-click the **DateKey** field, and then select **Hide in Report View**.



6. In the **Fields** pane, notice that the hidden field uses a dark gray color.

7. Repeat the last step to hide the remaining three visible “key” fields in the **Date** table.

8. In the **Fields** pane and the data grid, notice that the header text of the hidden columns is now displayed differently.

9. Expand the **Sales** table, and then hide the **OrderDate** and **Region Name** fields.

   These fields are available in related tables.

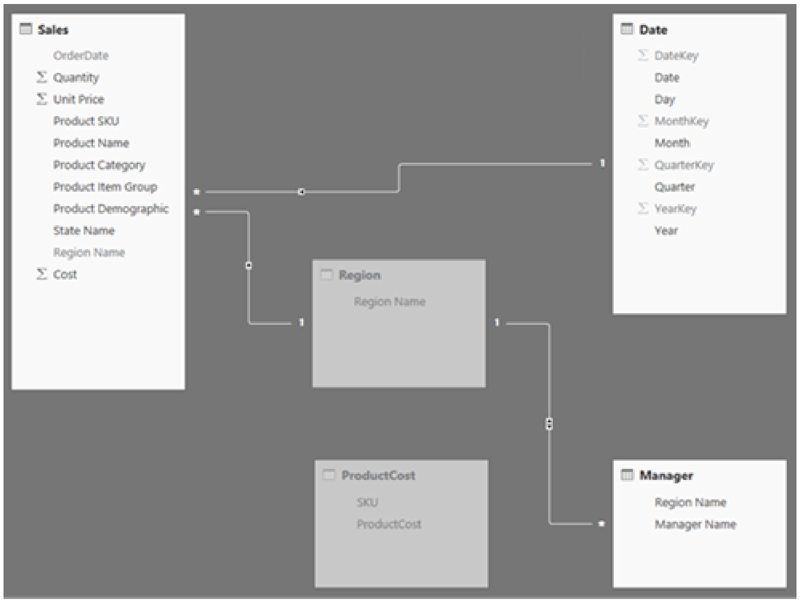
10. To hide the **ProductCost** table, right-click the table, and then select **Hide in Report View**.

11. Hide also the **Region** table.

12. Switch to Relationships view, and notice that hidden tables and fields are grayed out.

    Tip: It is also possible to hide tables and columns in Relationships view by right-clicking a table or column.

13. Verify that the relationship diagram matches the following.



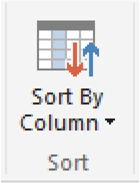
Sorting Columns

In this task, you will sort two **Date** table columns.

1. Switch back to Data view.

2. In the **Fields** pane, from inside the **Date** table, select the **Day** field.

3. On the **Modeling** ribbon, from inside the **Sort** group, click the **Sort by Column,** and then select **DateKey**.



4. Sort the **Month** column by the **MonthKey** column.

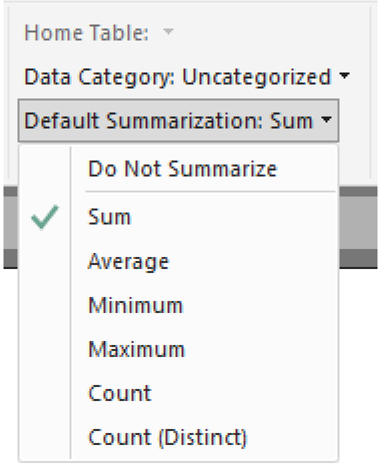
   *It is not necessary to configure the sort column for the **Quarter** and **Year** columns, as the text sort order is the same as the chronological sort order.*

Configuring Default Summarization

In this task, you will configure column default summarization.

1. In the **Fields** pane, from inside the **Sales** table, select the **Quantity** field.

2. On the **Modeling** ribbon, from inside the **Properties** group, in the **Default Summarization** dropdown list, notice that **Sum** is selected.


Home Table:

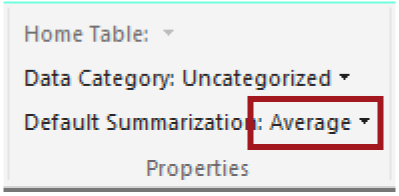
Data Category: Uncategorized

Default Summarization: Sum

   *When the query was imported into the model, numeric columns were configured to summarize by default. The **Sum** function is the most commonly used function to summarize numeric values. This property is the default, and it is possible in the report design for a user to switch to a different summarization function.*

3. In the **Fields** pane, from inside the **Sales** table, select the **Unit Price** field.

4. Set the **Default Summarization** property to **Average**.



Home Table:

Data Category: Uncategorized

Default Summarization: Average

It does not make sense to sum rates like unit prices, and so the updated default summarization will now provide a meaningful aggregation, i.e. the average of unit prices.

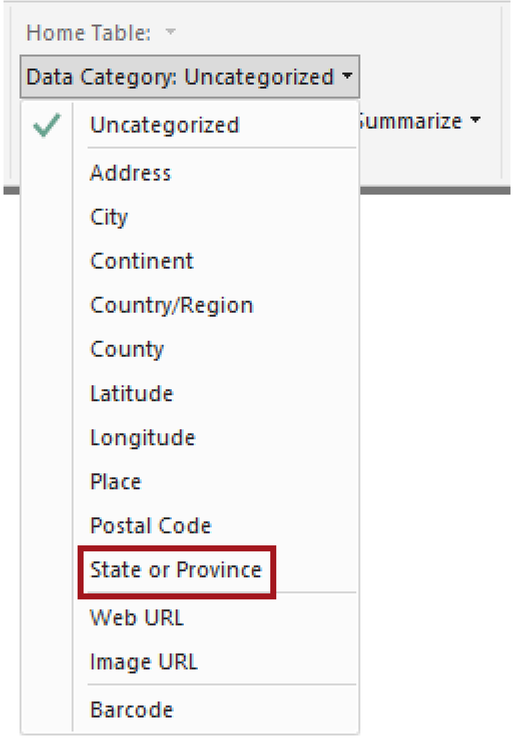
Default summarization can be set for numeric, date and text fields, with the latter two types only allowing count summarization.

Configuring Column Categorization

In this task, you will categorize the **State Name** column.

1. In the **Fields** pane, from inside the **Sales** table, select the **State Name** field.

2. On the **Modeling** ribbon, from inside the **Properties** group, in the **Data Category** dropdown list, select **State or Province**.

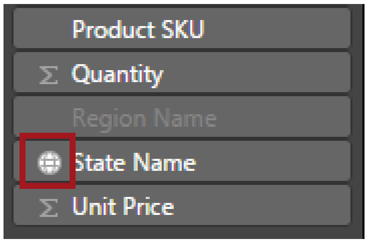


Home Table:

Data Category: Uncategorized

This will enable default spatial reporting by using maps.

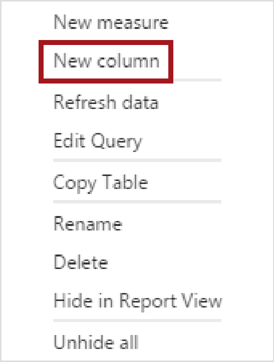
3. In the **Fields** pane, notice that the **State Name** field is now adorned with the spatial icon.



Creating a Calculated Column

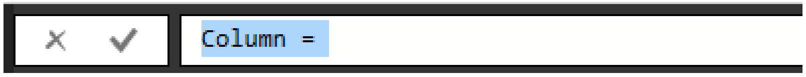
In this task, you will create a calculated column to enable profit reporting.

1. In the **Fields** pane, right-click the **Sales** table, and then select **New Column**.



Tip: You can also create new calculations by using the ribbon commands.

2. Notice that the focus is set to the formula bar, and that the column name defaults to **Column**, and is selected.



```
Column =
```

3. To rename the column, replace the text **Column** with **Revenue**.

4. On the right side of the equals sign, use the assistance of IntelliSense to enter the following formula.

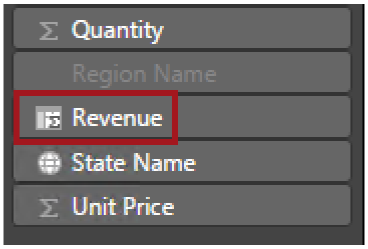
DAX

```
[Quantity] * [Unit Price]
```

This column could also have been implemented as a new column in a query, as either approach produces the same outcome. The appropriate approach will often be determined by the skillset of the analyst, and whether the M language (used by queries) or DAX (used for model calculations) best achieves the requirement.

5. Press **Enter**.

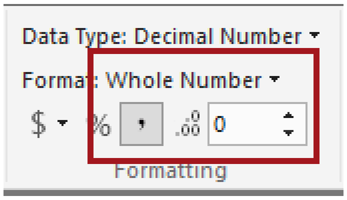
6. In the **Fields** pane, notice the addition of the new field.



7. To format the column, in the **Fields** pane, ensure that the **Revenue** field is selected.

8. On the **Modeling** ribbon, from inside the **Formatting** group, configure the following format properties.

   *If the formatting controls are not enabled, in the **Field** list, select another field, and then select the **Revenue** field.*



Data Type: Decimal Number

Format: Whole Number

$ % , .00 0

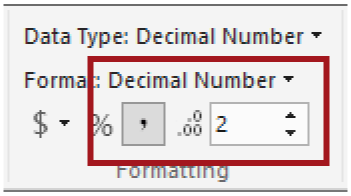
   For consistency in this lab, you will configure all monetary amounts with the same formatting settings.

Formatting Fields

In this task, you will format fields for presentation in reports.

1. Format the **Quantity** column with the thousands separator.

2. Set the format for the **Unit Price** column to **Decimal Number**, with the thousands separator and two decimal places.



Data Type: Decimal Number

Format: Decimal Number

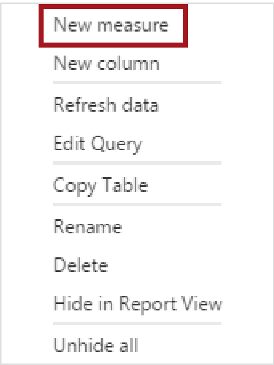
$ % , .00 2

3. Format the **Cost** column as a whole number and with the thousands separator.

Creating Measures

In this task, you will create two measures to report on profitability.

1. In the **Fields** pane, right-click the **Sales** table, and then select **New Measure**.



2. In the formula bar, replace the word **Measure** with **Profit**.

3. On the right side of the equals sign, enter the following formula.

DAX

```
SUM(Sales[Revenue]) - SUM(Sales[Cost])
```

Unlike calculated columns, measures involve the aggregation of values in columns. Also, unlike calculated columns, measures do not store their results in the model.

4. Format the **Profit** measure as a whole number and with the thousands separator.

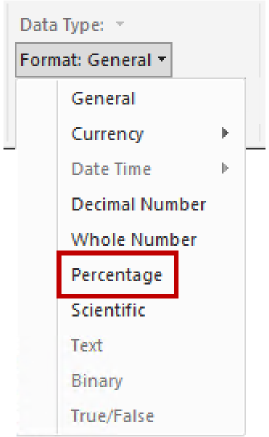
5. Repeat the previous steps in this task to create an additional measure named **Profitability**, by using the following formula.

DAX

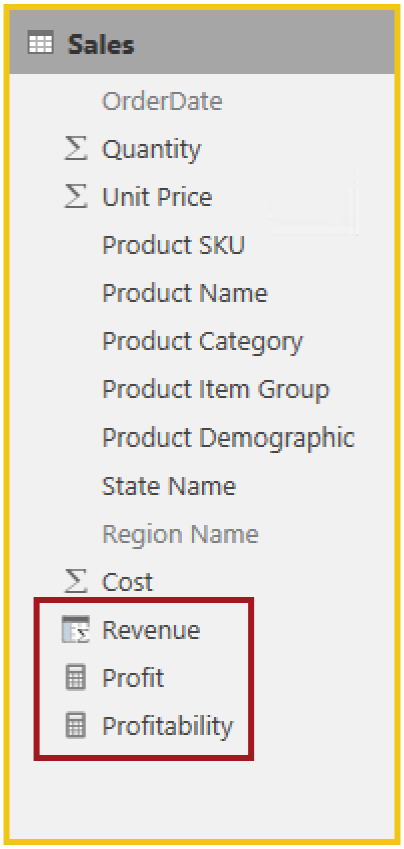
```
DIVIDE([Profit], SUM(Sales[Revenue]))
```

*The **DIVIDE** function divides two expressions, providing that the second argument results in a non-zero number. If the second argument results in zero or blank (missing), then the function will return blank.*

6. Format the new measure as **Percentage** and two decimal places.



7. Switch to Relationships view, and in the **Sales** table, notice that the three calculations are adorned with different icons.



Sales

OrderDate

∑ Quantity

∑ Unit Price

Product SKU

Product Name

Product Category

Product Item Group

Product Demographic

State Name

Region Name

∑ Cost

Revenue

Profit

Profitability

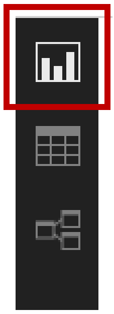
Creating Power BI Reports

In this exercise, you will create two report pages. The first report page will consist of five visualizations to report on and analyze regional sales and profitability. The second report page will import a custom visual, and use it to analyze revenue by manager and region.

Adding Report Decoration

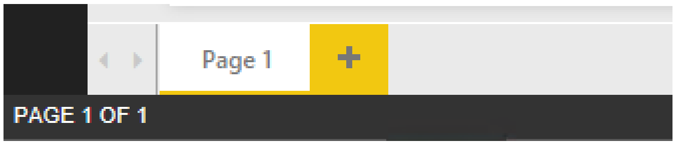
In this task, you will add a text box and an image to the default report page.

1. Switch to Report view.



   The blank area is the report page canvas.

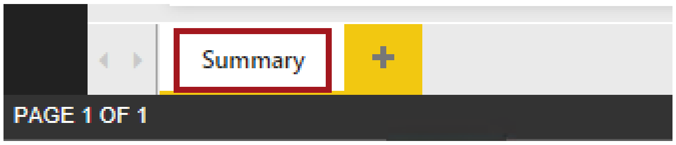
2. At the bottom left corner, notice the page navigation control.



   A report can consist of multiple pages. Only a single report page containing multiple data visualizations will be developed in this lab.

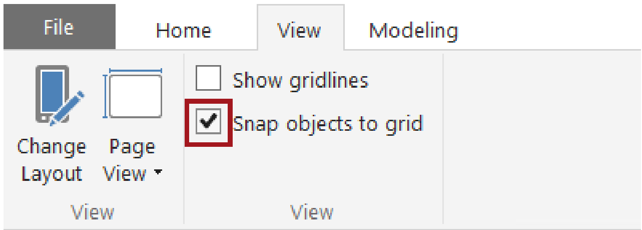
3. To name the page, double-click **Page 1**.

4. Replace the text with **Summary**, and then press **Enter**.



Summary

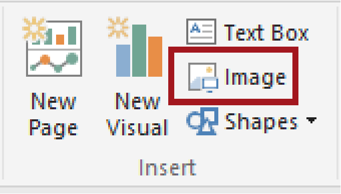
5. To enable snap-to-grid, on the **View** ribbon, check **Snap Objects to Grid**.



□ Show gridlines

☑ Snap objects to grid

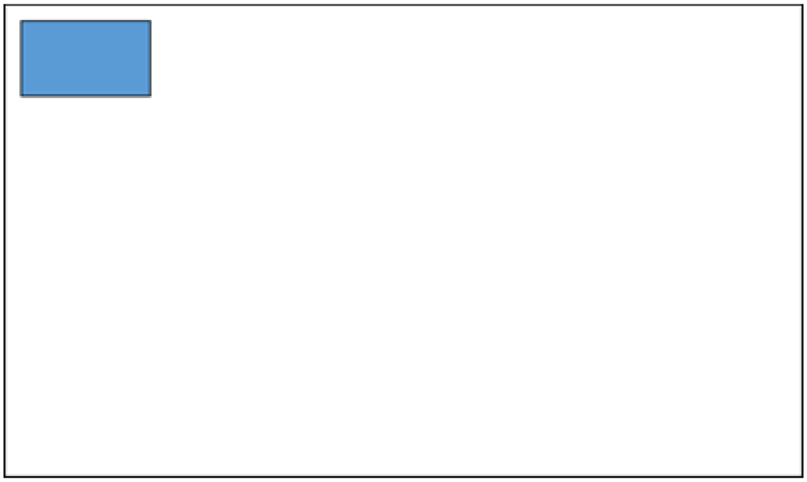
6. To insert an image, on the **Home** ribbon, from inside the **Insert** group, click **Image**.



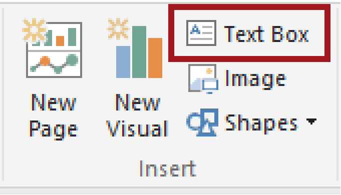
7. In the **Open** window, navigate to the **D:\PowerBI\Lab02\Assets** folder.
8. Select the **TailspinToysLogo.png** file, and then click **Open**.

   Tip: If your intention is to pin this image to a Power BI dashboard, for best results use a rectangular image with a 13:9 aspect ratio, or a white background. If the image is not exactly of a 13:9 ratio, Power BI will add white padding around the edges as needed.

9. To resize the image, ensure that it is selected to reveal the border guides.
10. Drag the border guides to resize the image to create a smaller sized tile, and then reposition it as follows.

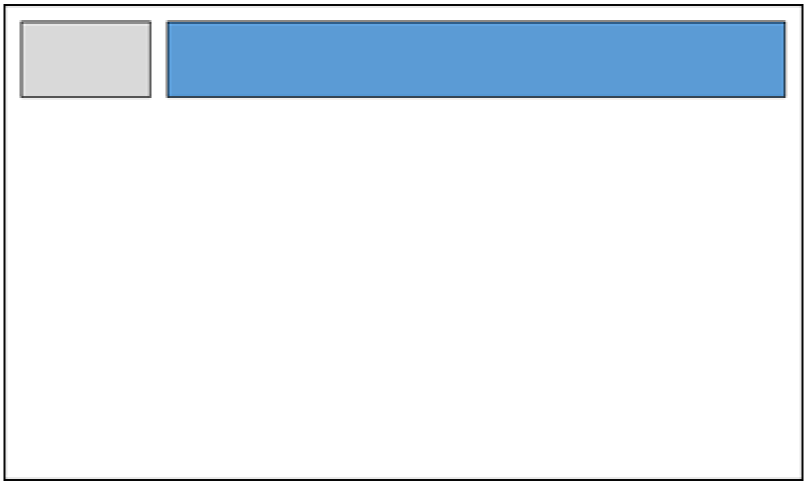


11. To insert a text box, on the **Home** ribbon, from inside the **Insert** group, click **Text Box**.



12. Inside the text box, enter **US Sales Analysis**.
13. Select the entire text, and then use the text format bar to increase the font size to **60**.

14. Resize the text box to a smaller size, and then reposition it as follows.



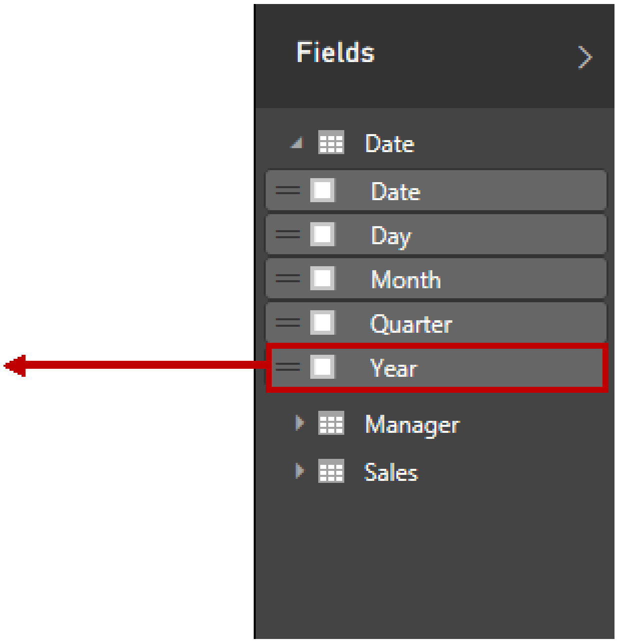
Creating a Slicer

In this task, you will create a slicer to enable the report user to interact and filter by year.

1. In the **Fields** pane, notice that only three tables are available.

   In Report view, only visible tables and fields are available for use in reports.

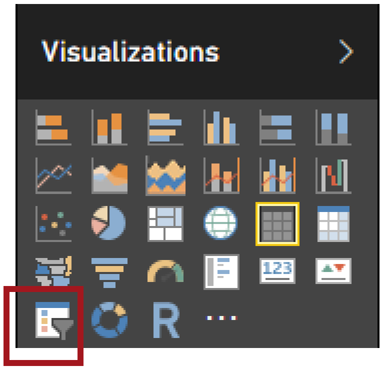
2. To create a visualization based on a field, in the **Fields** pane, expand the **Date** table, and then drag the **Year** field and drop it on a blank area of the canvas.



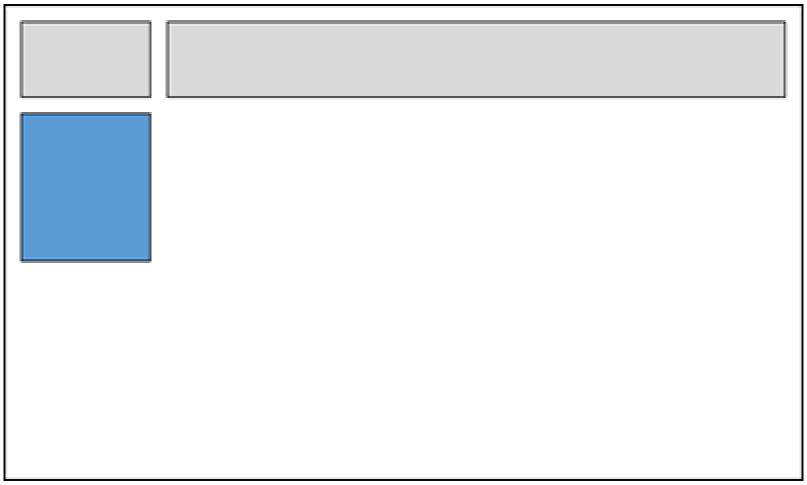
3. Notice that the field was used to create a table visualization.

4. To switch the visualization to a slicer, in the **Visualizations** pane, click the **Slicer** icon.

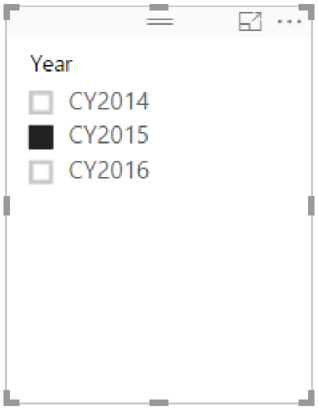
   Tip: Hovering over a visualization type icon will reveal a tooltip that describes the visualization type.



5. Resize the slicer, and then reposition it as follows.



6. In the slicer, select **CY2015**.



Year

□ CY2014

■ CY2015

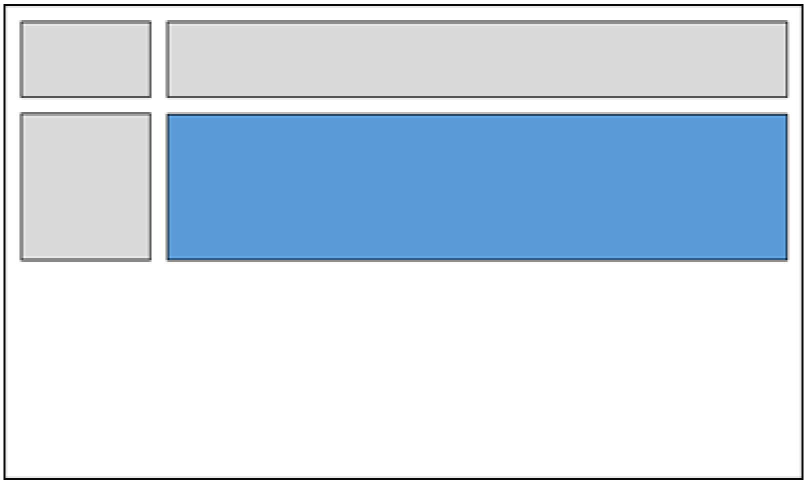
□ CY2016

All data visualizations on the page will now be filtered by ***CY2015****.*

Creating a Combo Chart

In this task, you will create a combo chart visualization to display monthly revenue, cost and profitability.

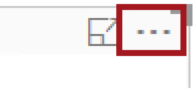
1. In the **Fields** pane, from inside the **Sales** table, drag the **Revenue** field and drop it on a blank area of the canvas.
2. Resize, and reposition, the visualization as follows.



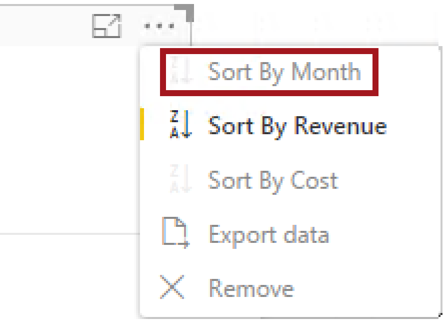
3. Drag the following fields and drop them inside the chart visualization.

Table	Field
Sales	Cost
Date	Month

4. On the horizontal axis, notice that the columns are sorted in descending revenue order.
5. To modify the sort order, first click the ellipsis located at the top-right corner of the visualization.

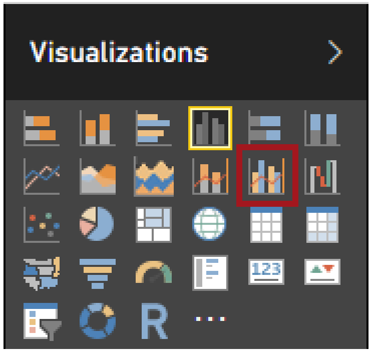


6. Select **Sort by Month**.

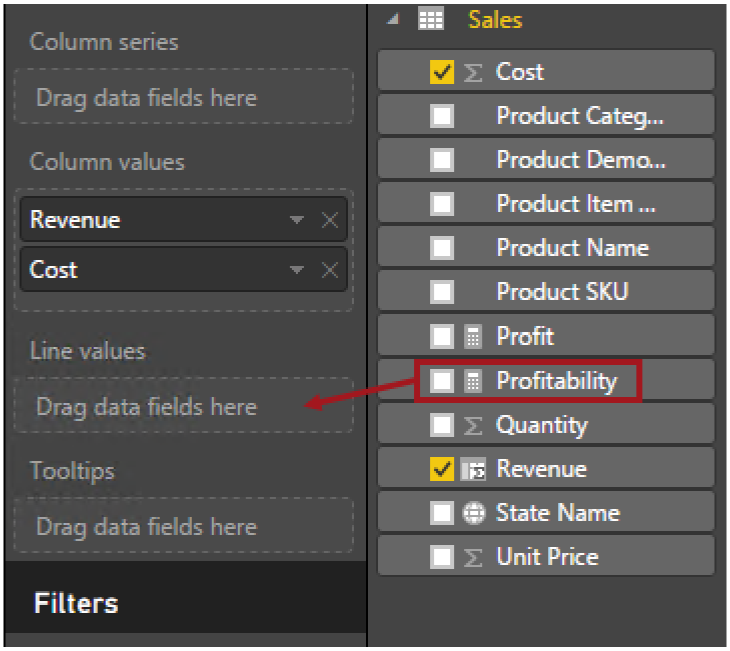


7. To sort by month ascending, click the ellipsis again, and then click **Sort by Month** again.

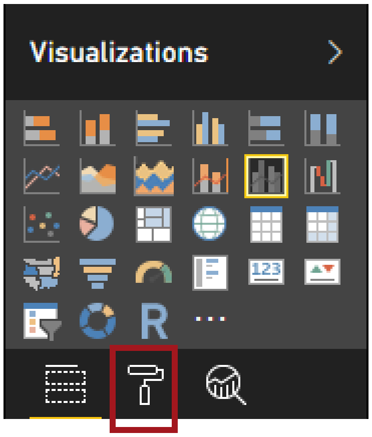
8. Modify the visualization to **Line and Clustered Column Chart**.



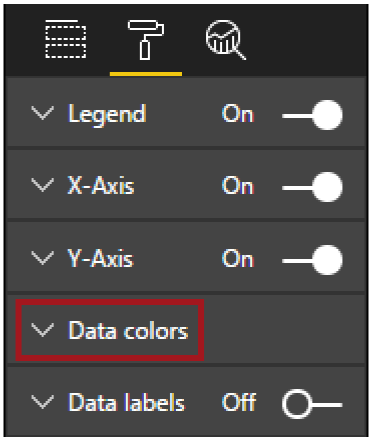
9. In the **Fields** pane, from inside the **Sales** table, drag the **Profitability** field and drop it in the **Visualizations** pane, inside the **Line Values** well.



10. To modify the visualization style, switch to Format view.



11. Expand **Data Colors**.



Legend On

X-Axis On

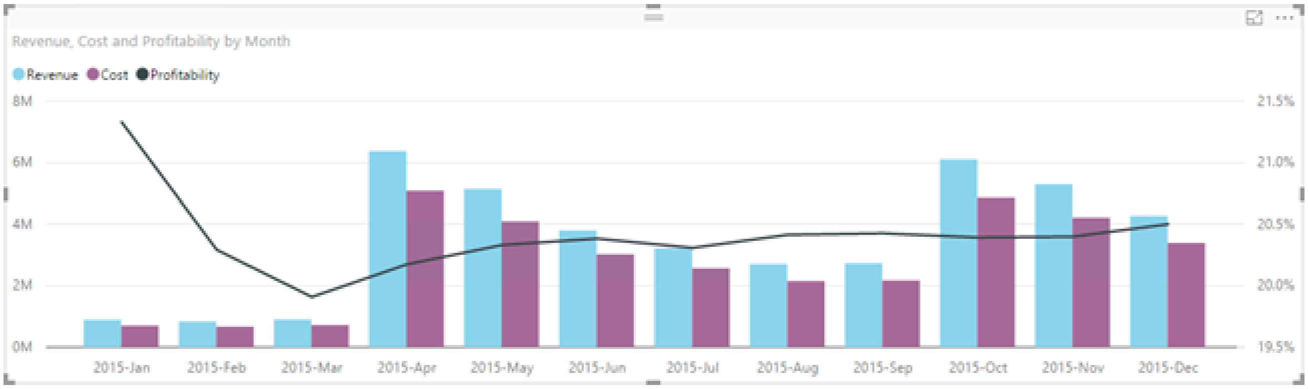
Y-Axis On

Data colors

Data labels Off

12. Select appropriate colors for the three fields (suggesting dark gray for **Profitability**, blue for **Revenue**, and purple for **Cost**).

13. Verify that the visualization matches the following.



14. Notice the different values and scales on the left and right sides of the chart.

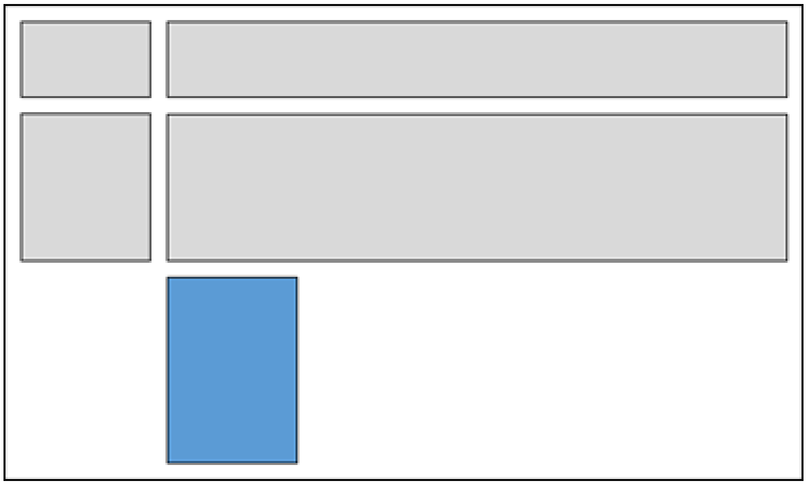
Creating Tables

In this task, you will create two tables. The first will display regional sales, and the second will display manager sales. The tables will help you to understand the many-to-many relationship defined between managers and sales.

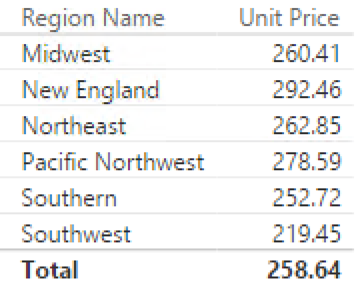
1. In the **Fields** pane, from inside the **Manager** table, drag the **Region Name** field and drop it on a blank area of the canvas.

2. Add the **Unit Price** field to the new visualization.

3. Resize, and reposition, the visualization as follows.



4. Verify that the table matches the following.

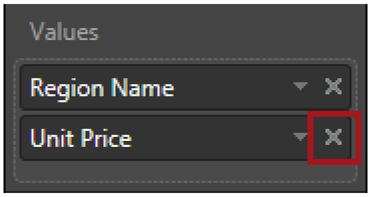


Region Name	Unit Price
Midwest	260.41
New England	292.46
Northeast	262.85
Pacific Northwest	278.59
Southern	252.72
Southwest	219.45
Total	**258.64**

5. Notice that the **Unit Price** values, including the total, are average prices.

   The unit price values are average summaries because you configured the default summarization in the previous exercise.

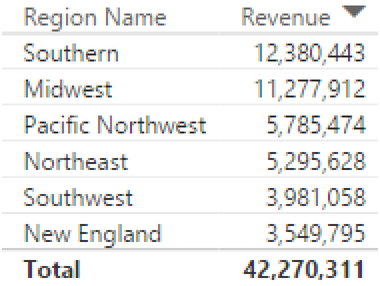
6. To remove the **Unit Price** column, in the **Values** well, for the **Unit Price** field, click **X**.



7. Add the **Revenue** field to the table.

8. To sort the table rows by descending revenue, click the **Revenue** column header.

9. Verify that the visualization matches the following.

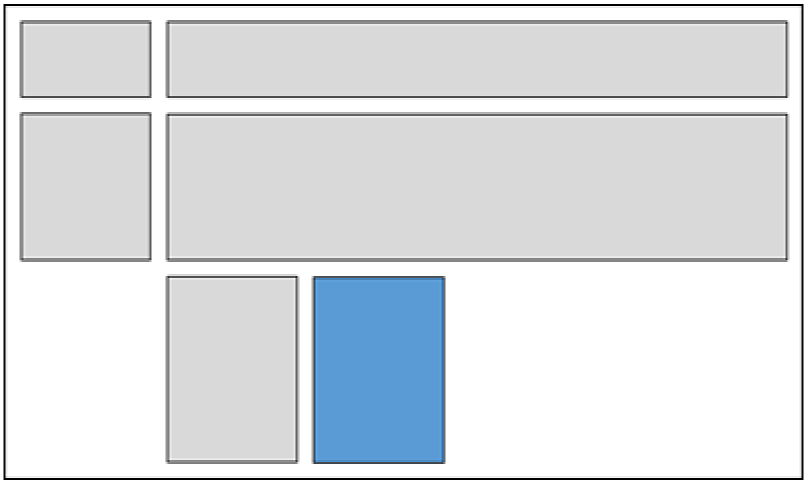


Region Name	Revenue ▼
Southern	12,380,443
Midwest	11,277,912
Pacific Northwest	5,785,474
Northeast	5,295,628
Southwest	3,981,058
New England	3,549,795
Total	**42,270,311**

10. Repeat the steps in this task to create a second table to display revenue, but this time by using the **Manager Name** field.

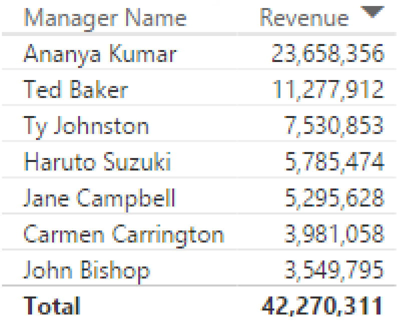
    Tip: It is possible to copy and paste a visualization, and then modify the settings of the copy. These commands are available on the ***Home*** *ribbon.*

11. Resize, and reposition, the visualization as follows.



12. Sort the table by descending revenue.

13. Verify that the visualization matches the following.



Manager Name	Revenue ▼
Ananya Kumar	23,658,356
Ted Baker	11,277,912
Ty Johnston	7,530,853
Haruto Suzuki	5,785,474
Jane Campbell	5,295,628
Carmen Carrington	3,981,058
John Bishop	3,549,795
Total	**42,270,311**

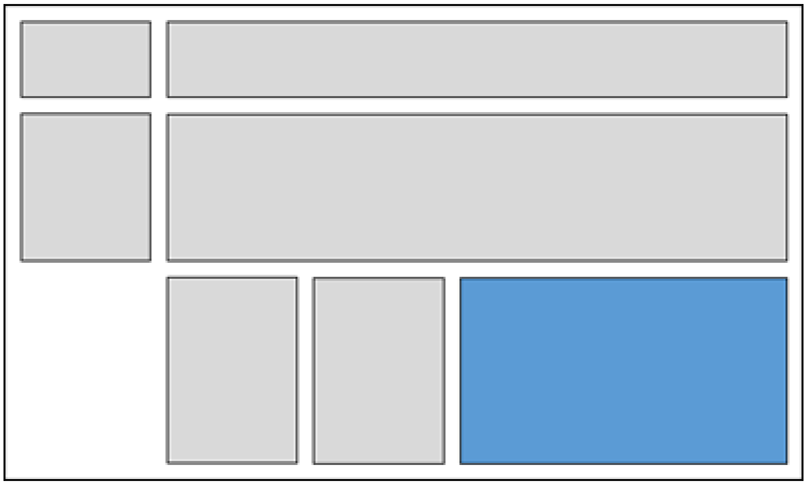
14. At a glance, notice that the sum of the individual manager revenue values easily exceeds the $42.27M table total.

    The many-to-many relationship results in duplication of revenue values, as multiple managers are assigned to a single region. The total of second table is correct, as can be verified against the total of the first table.

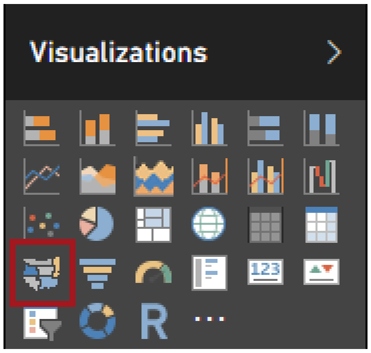
Creating a Fill Map

In this task, you will create a map to display US state profitability.

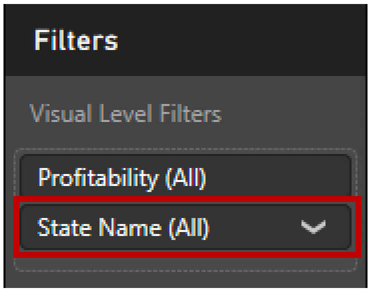
1. In the **Fields** pane, from inside the **Sales** table, create a new visualization based on the **State Name** field.
2. Resize, and reposition, the visualization as follows.



3. Drag the **Profitability** field and drop it into the map, and notice that it is added to **Values** well.
4. Modify the visualization to **Fill Map**.



5. To display only the contiguous US states, in the **Filters** section, click **State Name (All)** to expand the filter list.



Filters

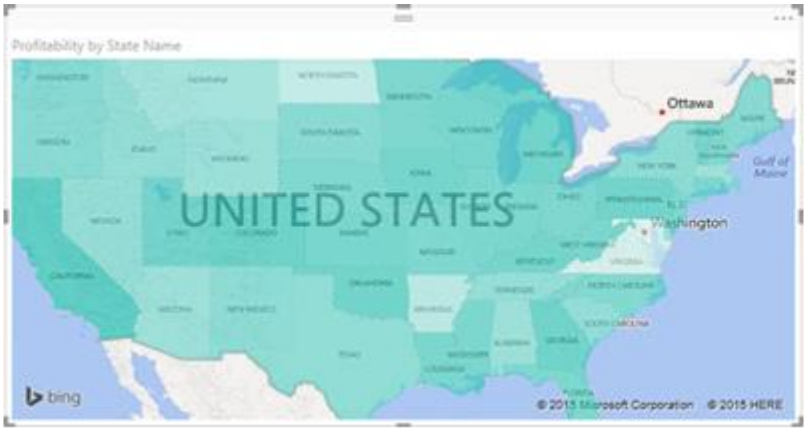
Visual Level Filters

Profitability (All)

State Name (All)

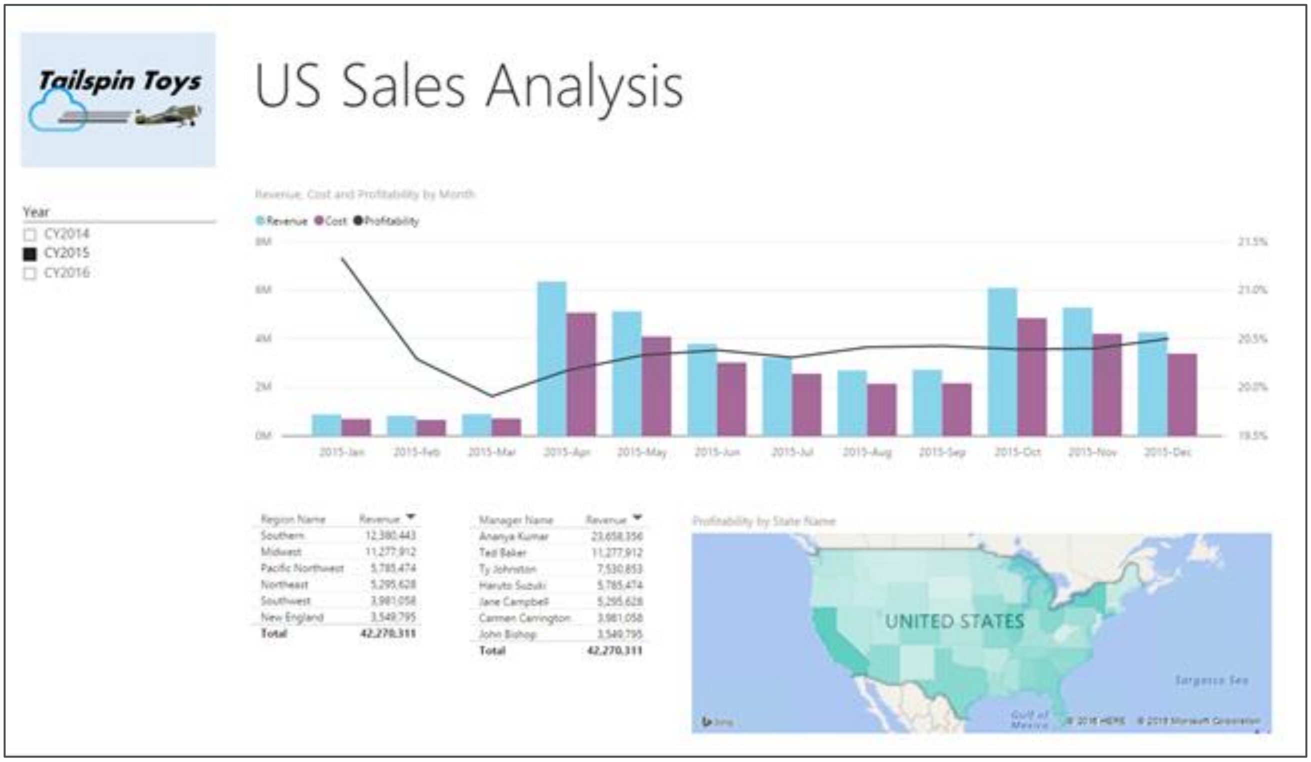
6. Check **(Select All)**, and then uncheck **Alaska** and **Hawaii**.

7. Verify that the visualization matches the following.



Profitability by State Name
UNITED STATES
Ottawa
Washington
Gulf of Maine
bing
© 2015 Microsoft Corporation © 2015 HERE

8. Verify that the report page matches the following.



US Sales Analysis

Year

☐ CY2014
■ CY2015
☐ CY2016

Region Name	Revenue ▼
Southern	12,380,443
Midwest	11,277,912
Pacific Northwest	5,785,474
Northeast	5,295,628
Southwest	3,981,058
New England	3,549,795
Total	**42,270,311**

Manager Name	Revenue ▼
Ananya Kumar	23,658,356
Ted Baker	11,277,912
Ty Johnston	7,530,853
Haruto Suzuki	5,785,474
Jane Campbell	5,295,628
Carmen Carrington	3,981,058
John Bishop	3,549,795
Total	**42,270,311**

Interacting with the Report Page

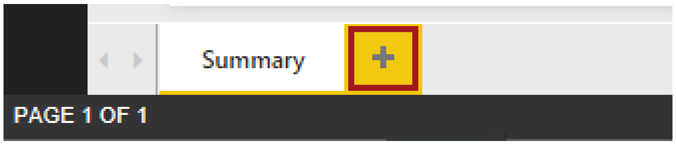
In this task, you will interact with the report page.

1. In the slicer, select **CY2016**.
2. To filter by a particular month, in the combo chart, hover the cursor over any one of the **Revenue** columns, and then take note of the value.
3. Click the column, and notice that all of the report page visualizations filter by that month. (You can verify this by comparing the table totals to the noted value.)
4. To clear the filter, click in a blank area of the combo chart.
5. To determine the profitability for **California**, hover the cursor over the state, and then take note of the value.
6. Click **California**, and notice that all of the report page visualizations filter by that state.
7. To clear the filter, either click **California** again, or click outside a state region on the map.

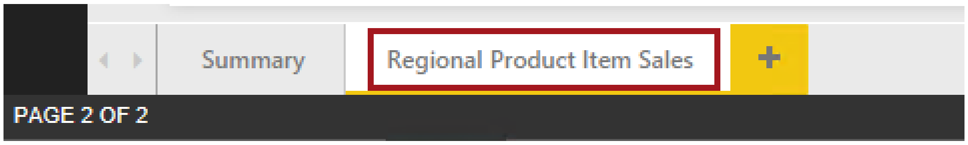
Creating a Second Report Page

In this task, you will create a second report page, and commence the design by copying and pasting in the **Year** slicer.

1. To add a report page, click the **New Page** command.

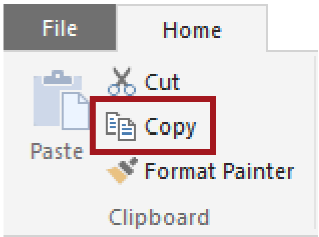


2. Rename the new report page to **Regional Product Item Sales**.

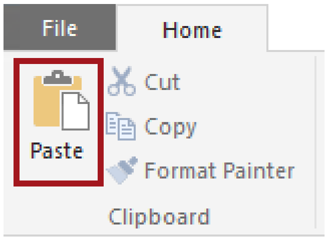


3. To add a slicer, return to the **Summary** report page, and then select the **Year** slicer.

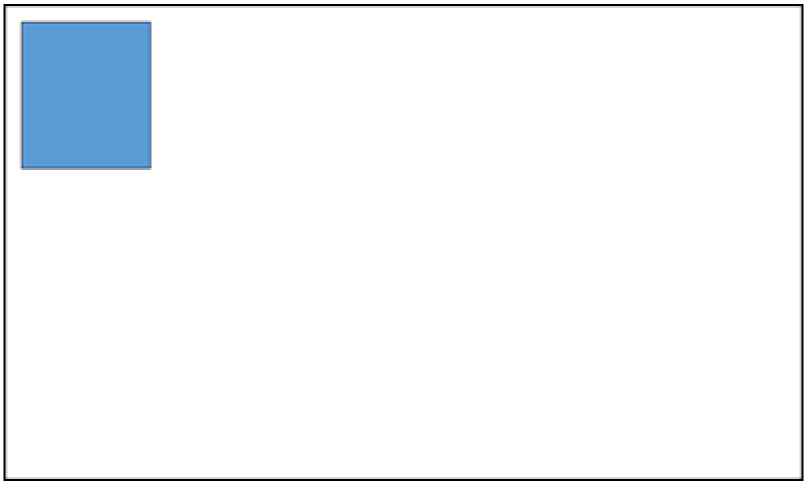
4. On the **Home** ribbon, from inside the **Clipboard** group, click **Copy**.



5. Return to the **Regional Product Item Sales** report page, and then on the **Home** ribbon, from inside the **Clipboard** group, click **Paste**.



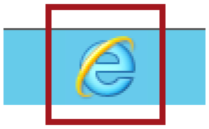
6. Reposition the visualization as follows.



Adding a Custom Visual

In this task, you will download a visual from the Office Store, and then import it into the Power BI Desktop file.

1. To open Internet Explorer, on the taskbar, right-click the **Internet Explorer** program shortcut.

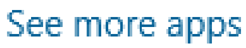


2. In Internet Explorer, navigate to https://store.office.com/en-us/appshome.aspx?productgroup=PowerBI.

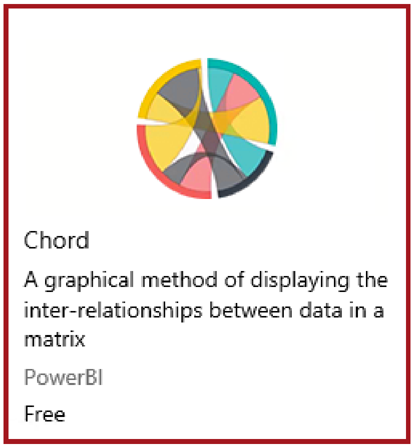
   *Tip: You can also use the **Custom Visuals for Power BI** Internet Explorer favorite.*

   The store allows discovering and downloading visuals created by Microsoft and the Power BI community. You can download and use these visuals in Power BI Desktop solutions. You can even submit one of your own for others to use.

3. Locate, and then select the **Chord** visual. If you cannot locate it, at the bottom-right click **See More Apps**.



See more apps



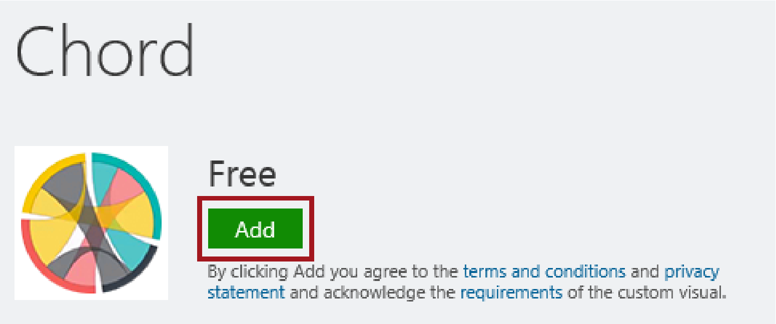
Chord

A graphical method of displaying the inter-relationships between data in a matrix

PowerBI

Free

4. Once having navigated to the **Chord** page, if you agree to the terms and conditions, click **Add**.



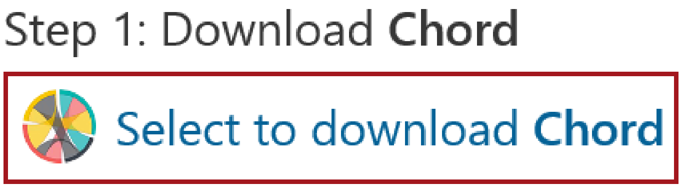
Chord

Free

Add

By clicking Add you agree to the terms and conditions and privacy statement and acknowledge the requirements of the custom visual.

5. Click **Select to Download Chord**.



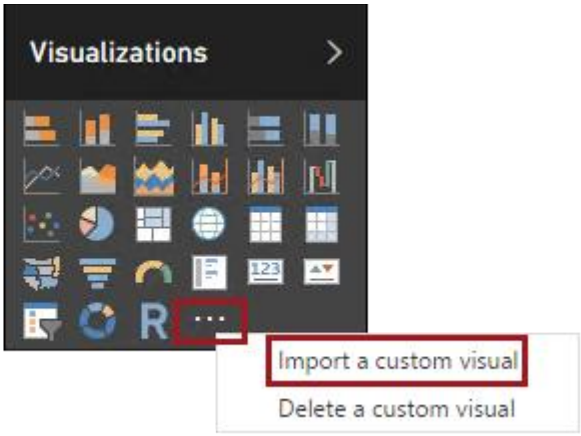
Step 1: Download Chord

Select to download Chord

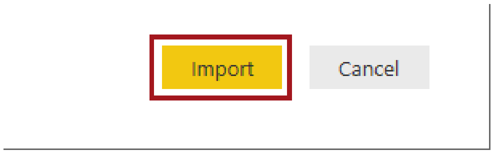
6. When prompted by Internet Explorer to download the visual, save the visual to the **D:\PowerBI\Lab02** folder.

7. Switch to Power BI Desktop.

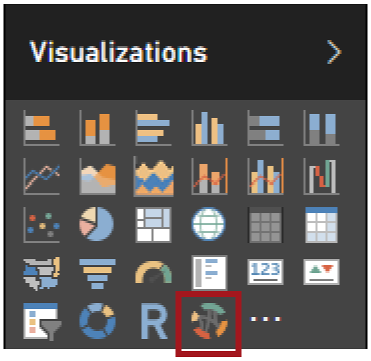
8. In the **Visualizations** pane, click the ellipsis, and then select **Import a Custom Visual**.



9. In the dialog window, click **Import**.



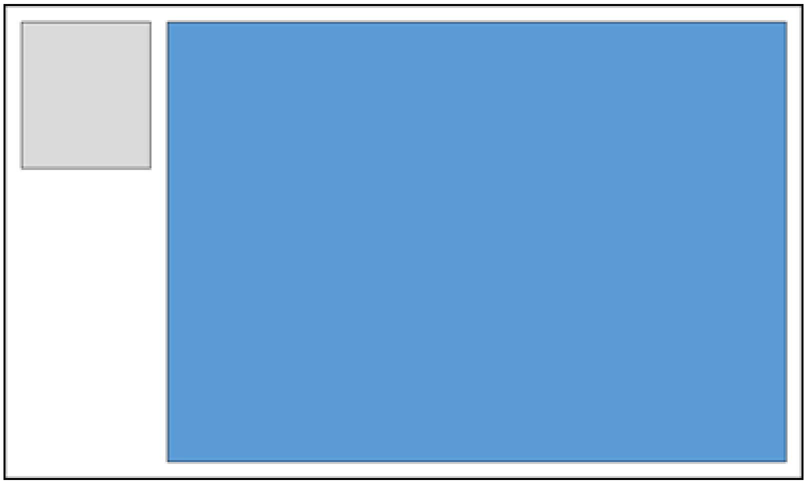
10. In the **Open** window, navigate to the **D:\PowerBI\Lab02** folder.
11. Select the downloaded visualization file, and then select **Open**.
12. When the visual is successfully imported, click **OK**.
13. Verify that the visual has been added to the **Visualizations** pane.



Configuring the Chord Visual

In this task, you will configure the chord visual to display revenue sales by region for each product item group.

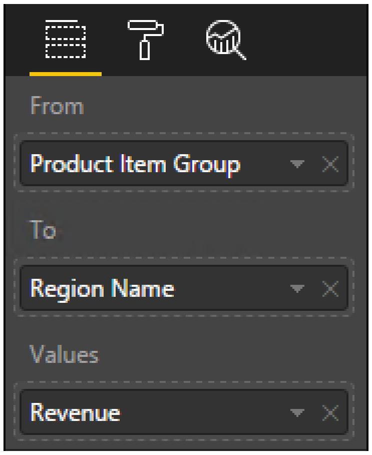
1. To add a chord visual to the report page, first ensure that the slicer is not selected, and then in the **Visualizations** pane, click the **Chord** visual.
2. Resize, and reposition, the visualization as follows.



3. From the **Fields** pane, from inside the **Sales** table, drag the **Product Item Group** field into the chord visual.
4. From inside the **Manager** table, drag the **Region Name** field into the chord visual.

   *Ignore the error which will soon be resolved when a field is added to the **Values** well.*

5. From inside the **Sales** table, drag the **Revenue** field into the chord visual.
6. Verify that the drag operations produced the following layout.



From

Product Item Group

To

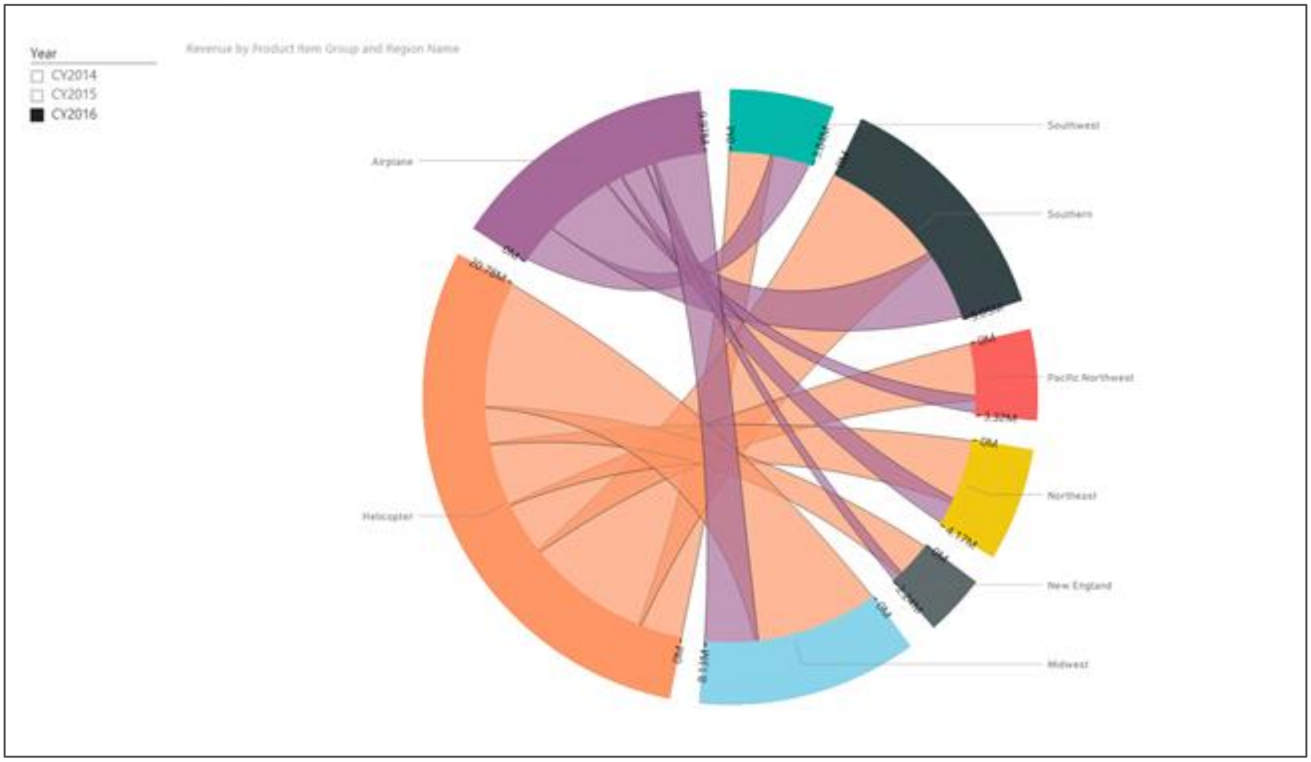
Region Name

Values

Revenue

Tip: To add fields to the in-focus visualization, you can also check the fields, or drag then directly into the wells.

7. Verify that the report page matches the following.



Interacting with the Report Page

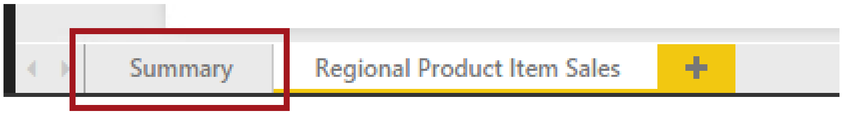
In this task, you will interact with the report page.

1. Modify the slicer values, and notice how the chord animates to reflect updated values.
2. Hover the cursor over the circle segments (representing total value) and also the chords, to reveal informative tooltips.

Finishing Up

In this task, you will finish up by saving the Power BI Desktop file, and then closing the application.

1. In the **Year** slicer, ensure that **CY2016** is selected.
2. Select the **Summary** report page.



Tip: By default, the report will display the page last selected. Be sure to select the report page you would like users to see first.

3. In the **Year** slicer, ensure that **CY2016** is selected.
4. To save the Power BI Desktop file, on the **File** menu, select **Save**.
5. To close the application, on the **File** tab, menu **Exit**.

6. Close the Internet Explorer Office Store window.

   The Power BI Desktop file is now ready to upload to the Power BI site. It will be uploaded in the lab titled ***Creating and Exploring a Microsoft Power BI Dashboard****.*

Summary

In this lab, you created a Power BI Desktop solution to enable the reporting and analysis of regional sales activity. This involved creating Power BI Desktop queries that sourced data from SQL Server, an Excel workbook and a CSV file.

You then created two report pages consisting of various data visualizations, including a visual downloaded from the Power BI Visual Gallery.

Terms of Use

By using this hands-on lab, you agree to the following terms:

The technology/functionality described in this hands-on lab is provided by Microsoft Corporation in a “sandbox” testing environment for purposes of obtaining your feedback and to provide you with a learning experience. You may only use the hands-on lab to evaluate such technology features and functionality and provide feedback to Microsoft. You may not use it for any other purpose. Without written permission, you may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this hands-on lab or any portion thereof.

COPYING OR REPRODUCTION OF THE HANDS-ON LAB (OR ANY PORTION OF IT) TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION WITHOUT WRITTEN PERMISSION IS EXPRESSLY PROHIBITED.

THIS HANDS-ON LAB PROVIDES CERTAIN SOFTWARE TECHNOLOGY/PRODUCT FEATURES AND FUNCTIONALITY, INCLUDING POTENTIAL NEW FEATURES AND CONCEPTS, IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION FOR THE PURPOSE DESCRIBED ABOVE. THE TECHNOLOGY/CONCEPTS REPRESENTED IN THIS HANDS-ON LAB MAY NOT REPRESENT FULL FEATURE FUNCTIONALITY AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. WE ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH FEATURES OR CONCEPTS. YOUR EXPERIENCE WITH USING SUCH FEATURES AND FUNCITONALITY IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

FEEDBACK If you give feedback about the technology features, functionality and/or concepts described in this hands-on lab to Microsoft, you give to Microsoft, without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your feedback in them. These rights survive this agreement.

MICROSOFT CORPORATION HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE HANDS-ON LAB, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, OUTPUT THAT DERIVES FROM USE OF THE VIRTUAL LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE VIRTUAL LAB FOR ANY PURPOSE.

DISCLAIMER This lab contains only a portion of new features and enhancements in Microsoft Power BI. Some of the features might change in future releases of the product.

Document Version

#	Date	Author	Comments
0	14-APR-2015	Peter Myers	Power BI Designer v2.21.3975.261
1	25-APR-2015	Peter Myers	Power BI Designer v2.22.4009.122
2	29-APR-2015	Peter Myers	Reformat as Power BI lab
3	06-JUN-2015	Peter Myers	Power BI Designer v2.23.4036.321
4	16-JUL-2015	Peter Myers	Power BI Desktop v2.24.4091.1
5	24-JUL-2015	Peter Myers	Power BI Desktop v2.25.4095.554 (GA)
6	06-AUG-2015	Peter Myers	Improvements made based on feedback
7	28-AUG-2015	Peter Myers	Migrate to new document template Power BI Desktop v2.26.4128.403
8	26-SEP-2015	Peter Myers	Power BI Desktop v2.27.4163.351
9	24-OCT-2015	Peter Myers	Power BI Desktop v2.28.4190.122
10	30-NOV-2015	Peter Myers	Power BI Desktop v2.29.4217.221
11	30-DEC-2015	Peter Myers	Power BI Desktop v2.30.4246.281
12	28-JAN-2016	Peter Myers	Power BI Desktop v2.30.4246.281
13	04-MAR-2016	Peter Myers	Power BI Desktop v2.32.4307.522
14	31-MAR-2016	Peter Myers	Power BI Desktop v2.33.4337.281
15	20-JUL-2016	Peter Myers	Updated Power BI logo TailspinToys-US database v1.02 Power BI Desktop v2.36.4434.381 Chord Chart v0.3.3
16	22-AUG-2016	Peter Myers	Power BI Desktop v2.37.4464.602 Chord Chart v0.3.4
17	04-SEP-2016	Peter Myers	Power BI Desktop v2.38.4491.282 Chord Chart v0.3.4
18	31-DEC-2016	Peter Myers	Power BI Desktop v2.41.4581.361 Chord Chart v0.3.4
19	01-FEB-2017	Peter Myers	Power BI Desktop v2.42.4611.901 Chord Chart v0.4.3
20	14-APR-2017	Peter Myers	Power BI Desktop v2.45.4704.442 Chord Chart v1.0.0.0 (downloaded from the Office Store)