Table of Contents

Lab story

This lab is one of many in a series of labs that was designed as a complete story from data preparation to publication as reports and dashboards. You can complete the labs in any order. However, if you intend to work through multiple labs, for the first 10 labs, we suggest you do them in the following order:

1. Prepare Data in Power BI Desktop
2. Load Data in Power BI Desktop
3. Model Data in Power BI Desktop, Part 1
4. Model Data in Power BI Desktop, Part 2
5. Create DAX Calculations in Power BI Desktop, Part 1
6. Create DAX Calculations in Power BI Desktop, Part 2
7. Design a Report in Power BI Desktop, Part 1
8. Design a Report in Power BI Desktop, Part 2
9. Create a Power BI Dashboard
10. Create a Power BI Paginated Report
11. Perform Data Analysis in Power BI Desktop

Prepare Data in Power BI Desktop

In this lab you commence the development of a Power BI Desktop solution for the Adventure Works company. It involves connecting to source data, previewing the data, and using data preview techniques to understand the characteristics and quality of the source data.

In this lab you learn how to:

- Open Power BI Desktop
- Set Power BI Desktop options
- Connect to source data
- Preview source data
- Use data preview techniques to better understand the data

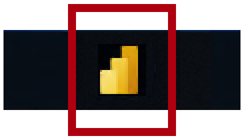
Exercise 1: Prepare Data

In this exercise you will create eight Power BI Desktop queries. Six queries will source data from SQL Server, and two from CSV files.

Task 1: Save the Power BI Desktop file

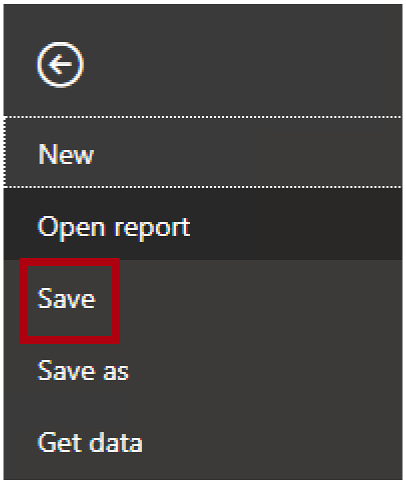
In this task you will first save the Power BI Desktop file.

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

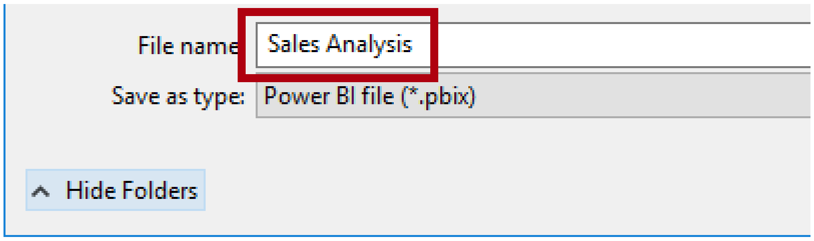


2. □ To close the getting started window, at the top-right of the window, click **X**.

3. □ To save the file, click the **File** ribbon tab to open the backstage view.
4. □ Select **Save**.

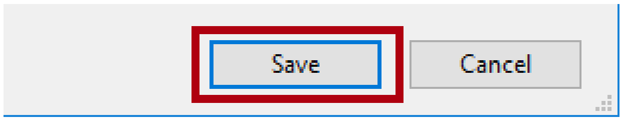


5. □ In the **Save As** window, navigate to the **D:\DA100\MySolution** folder.
6. □ In the **File Name** box, enter **Sales Analysis**.

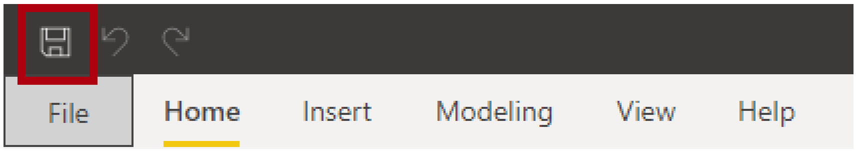

File name: Sales Analysis

Save as type: Power BI file (*.pbix)

7. □ Click **Save**.



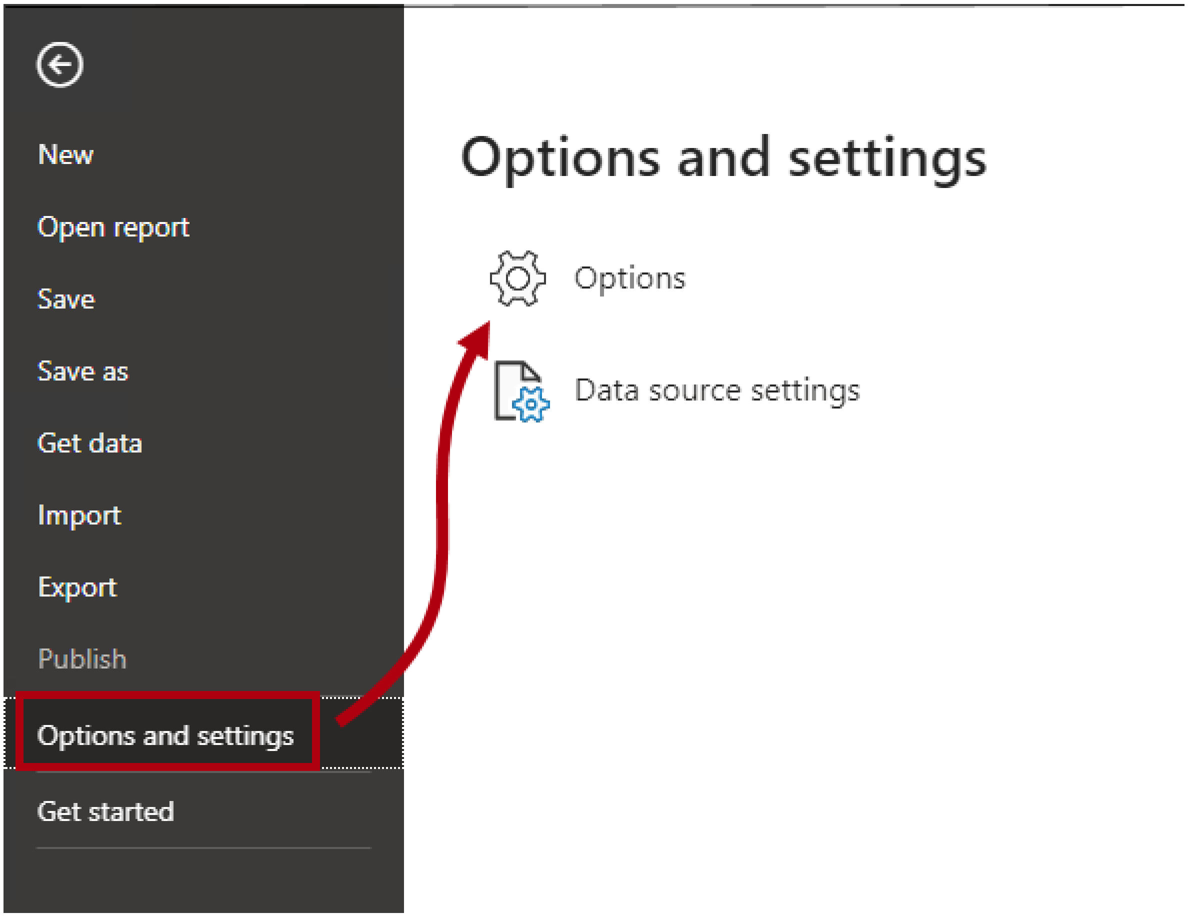
Tip: You can also save the file by click the **Save** icon located at the top-left.



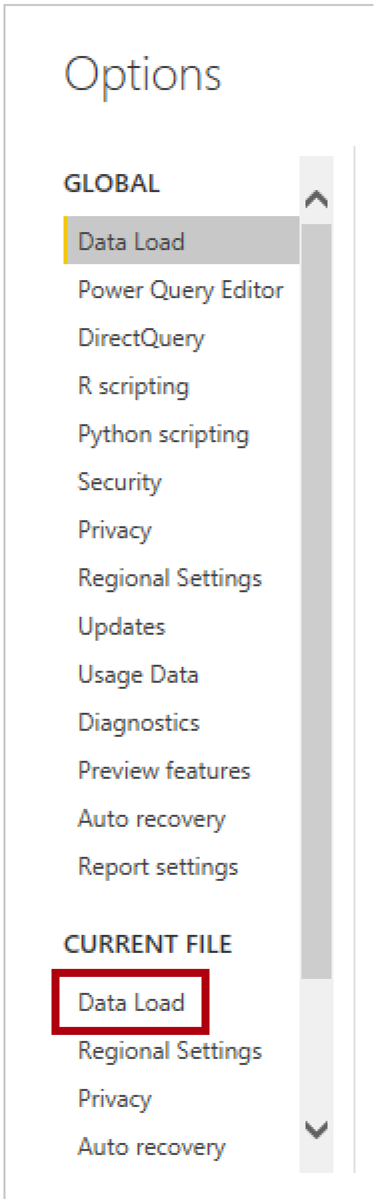
Task 2: Set Power BI Desktop options

In this task you will set Power BI Desktop options.

1. □ In Power BI Desktop, click the **File** ribbon tab to open the backstage view.
2. □ At the left, select **Options and Settings**, and then select **Options**.

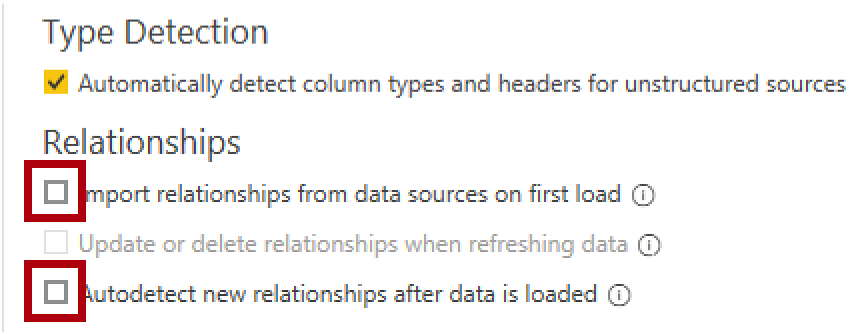


3. □ In the **Options** window, at the left, in the **Current File** group, select **Data Load**.



The **Data Load** settings for the current file allow setting options that determine default behaviors when modeling.

4. □ In the **Relationships** group, uncheck the two options that are already checked.



Type Detection

☑ Automatically detect column types and headers for unstructured sources

Relationships

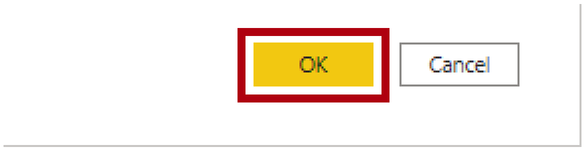
☐ Import relationships from data sources on first load ⓘ

☐ Update or delete relationships when refreshing data ⓘ

☐ Autodetect new relationships after data is loaded ⓘ

While having these two options enabled can be helpful when developing a data model, you disabled them earlier to support the lab experience. When you create relationships in the **Load Data in Power BI Desktop** lab, you'll learn why you are adding each one.

5. ☐ Click **OK**.

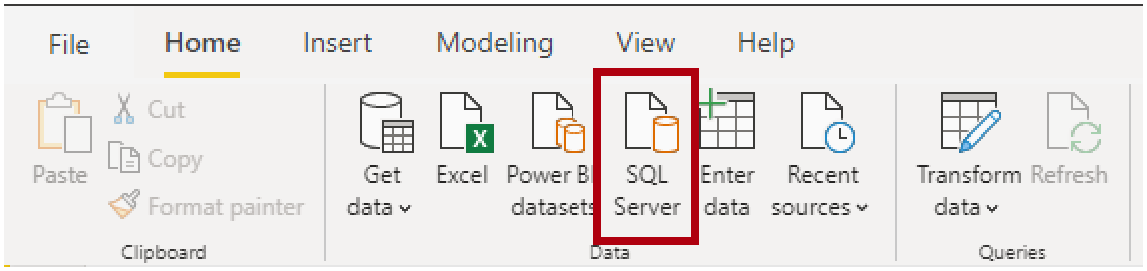


6. ☐ Save the Power BI Desktop file.

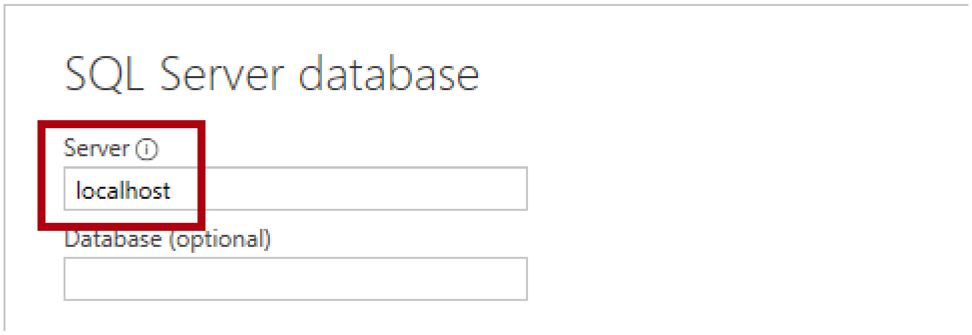
Task 3: Get data from SQL Server

In this task you will create queries based on SQL Server tables.

1. ☐ On the **Home** ribbon tab, from inside the **Data** group, click **SQL Server**.



2. ☐ In the **SQL Server Database** window, in the **Server** box, enter **localhost**.


SQL Server database

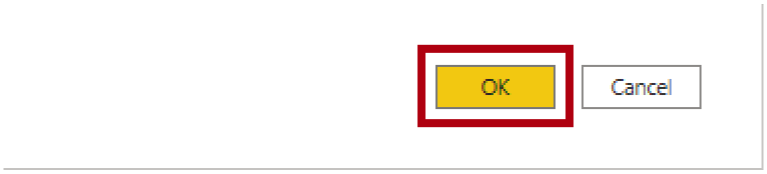
Server ⓘ

localhost

Database (optional)

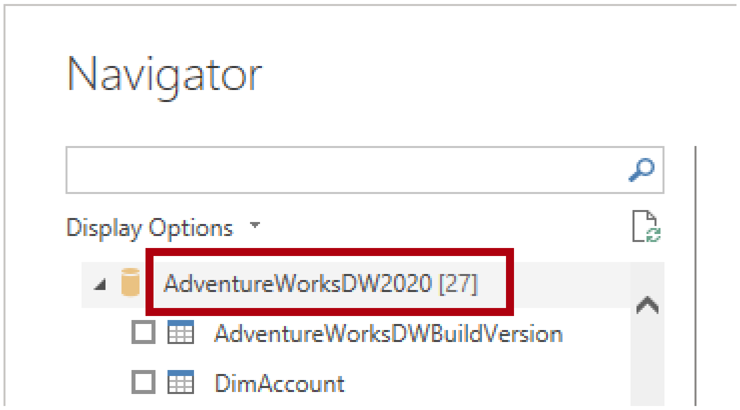
In this lab you'll connect to the SQL Server database by using **localhost**. This isn't a recommended practice when creating your own solutions. It's because gateway data sources cannot resolve **localhost**.

3. □ Click **OK**.

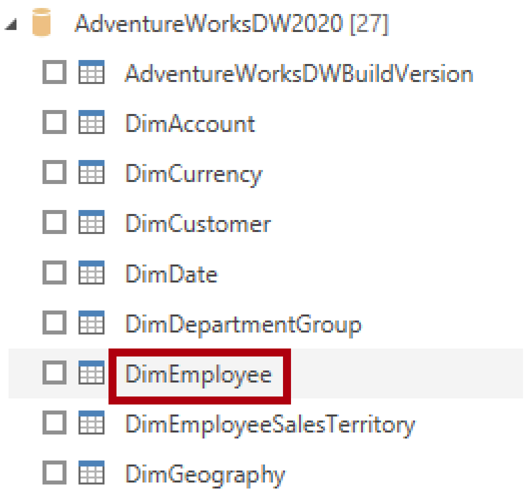


4. □ In the **Navigator** window, at the left, expand the **AdventureWorksDW2020** database.

The **AdventureWorksDW2020** database is based on the **AdventureWorksDW2017** sample database. It has been modified to support the learning objectives of the course labs.



5. □ Select—but don't check—the **DimEmployee** table.



AdventureWorksDW2020 [27]

□ AdventureWorksDWBuildVersion

□ DimAccount

□ DimCurrency

□ DimCustomer

□ DimDate

□ DimDepartmentGroup

□ DimEmployee

□ DimEmployeeSalesTerritory

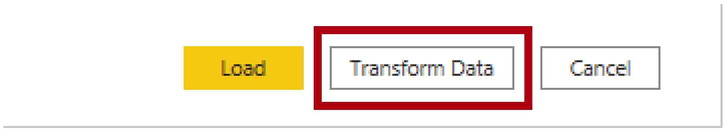
□ DimGeography

6. □ In the right pane, notice a preview of the table data.

   The preview data allows you to determine the columns and a sample of rows.

7. □ To create queries, select the checkbox next to the following six tables:
   - DimEmployee
   - DimEmployeeSalesTerritory
   - DimProduct
   - DimReseller
   - DimSalesTerritory
   - FactResellerSales
8. □ To apply transformations to the data of the selected tables, click **Transform Data**.

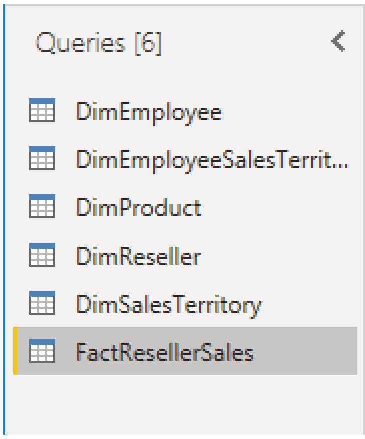
   You won’t be transforming the data in this lab. The objectives of this lab focus on exploring and profiling the data in the **Power Query Editor** window.



Task 4: Preview SQL Server queries

In this task you will preview the data of the SQL Server queries. First, you will learn relevant information about the data. You will also use column quality, column distribution, and column profile tools to understand the data and to assess data quality.

1. □ In the **Power Query Editor** window, at the left, notice the **Queries** pane.

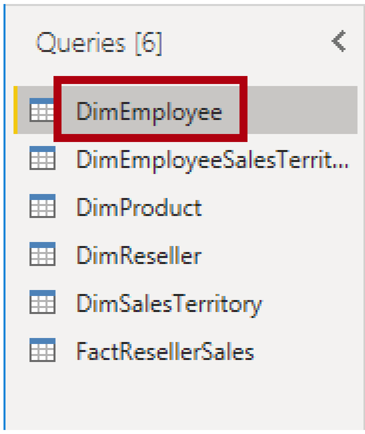


Queries [6]

DimEmployee
DimEmployeeSalesTerrit...
DimProduct
DimReseller
DimSalesTerritory
FactResellerSales

The **Queries** pane contains one query for each table you checked.

2. □ Select the first query—**DimEmployee**.

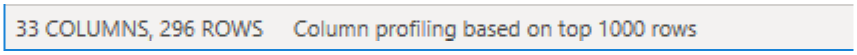


Queries [6]

DimEmployee
DimEmployeeSalesTerrit...
DimProduct
DimReseller
DimSalesTerritory
FactResellerSales

The **DimEmployee** table in the SQL Server database stores one row for each employee. A subset of the rows from this table represents the salespeople, which will be relevant to the model you'll develop.

3. □ At the bottom left, in the status bar, notice the table statistics—the table has 33 columns, and 296 rows.

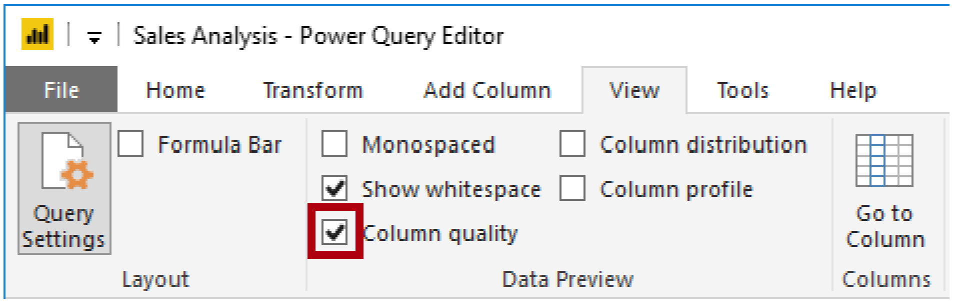


33 COLUMNS, 296 ROWS   Column profiling based on top 1000 rows

4. □ In the data preview pane, scroll horizontally to review all columns.
5. □ Notice that the last five columns contain **Table** or **Value** links.

These five columns represent relationships to other tables in the database. They can be used to join tables together. You'll join tables in the **Load Data in Power BI Desktop** lab.

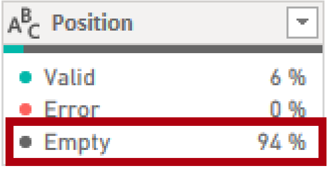
6. □ To assess column quality, on the **View** ribbon tab, from inside the **Data Preview** group, check **Column Quality**.



☐ Formula Bar
☐ Monospaced
☑ Show whitespace
☑ Column quality
☐ Column distribution
☐ Column profile

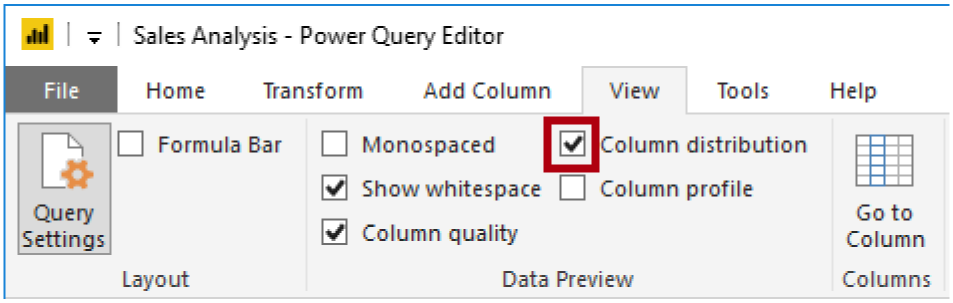
The column quality feature allows you to easily determine the percentage of valid, error, or empty values found in columns.

7. □ For the **Position** column (sixth last column), notice that 94% of rows are empty (null).



● Valid	6 %
● Error	0 %
● Empty	94 %

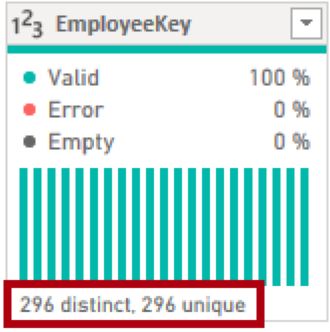
8. □ To assess column distribution, on the **View** ribbon tab, from inside the **Data Preview** group, check **Column Distribution**.



☐ Formula Bar
☐ Monospaced
☑ Show whitespace
☑ Column quality
☑ Column distribution
☐ Column profile

9. □ Review the **Position** column again, and notice that there are four distinct values, and one unique value.

10. □ Review the column distribution for the **EmployeeKey** (first) column—there are 296 distinct values, and 296 unique values.

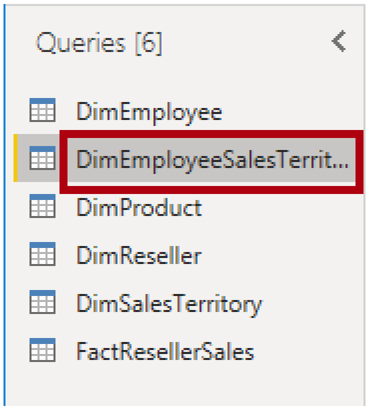


Valid 100 %
Error 0 %
Empty 0 %

296 distinct, 296 unique

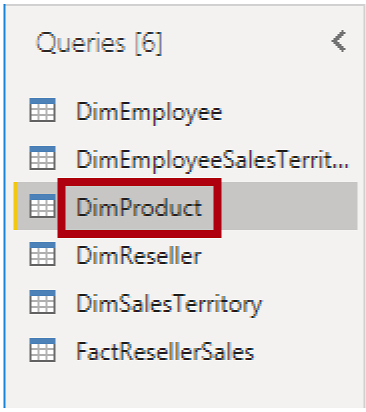
When the distinct and unique counts are the same, it means the column contains unique values. When modeling, it's important that some model tables have unique columns. These unique columns can be used to create one-to-many relationships, which you will do in the **Model Data in Power BI Desktop, Part 1** lab.

11. □ In the **Queries** pane, select the **DimEmployeeSalesTerritory** query.



The **DimEmployeeSalesTerritory** table stores one row for each employee and the sales territory regions they manage. The table supports relating many regions to a single employee. Some employees manage one, two, or possibly more regions. When you model this data, you'll need to define a many-to-many relationship, which you'll do in the **Model Data in Power BI Desktop, Part 2** lab.

12. □ In the **Queries** pane, select the **DimProduct** query.



Queries [6]

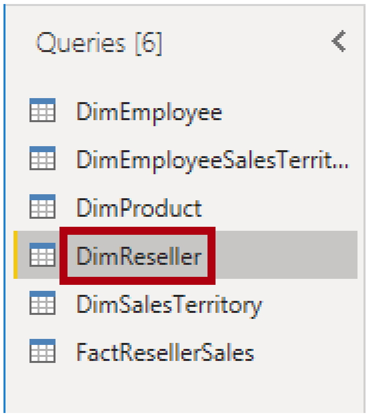
- DimEmployee
- DimEmployeeSalesTerrit...
- DimProduct
- DimReseller
- DimSalesTerritory
- FactResellerSales

The **DimProduct** table contains one row per product sold by the company.

13. □ Horizontally scroll to reveal the last columns.
14. □ Notice the **DimProductSubcategory** column.

When you add transformations to this query in the **Load Data in Power BI Desktop** lab, you'll use the **DimProductSubcategory** column to join tables.

15. □ In the **Queries** pane, select the **DimReseller** query.

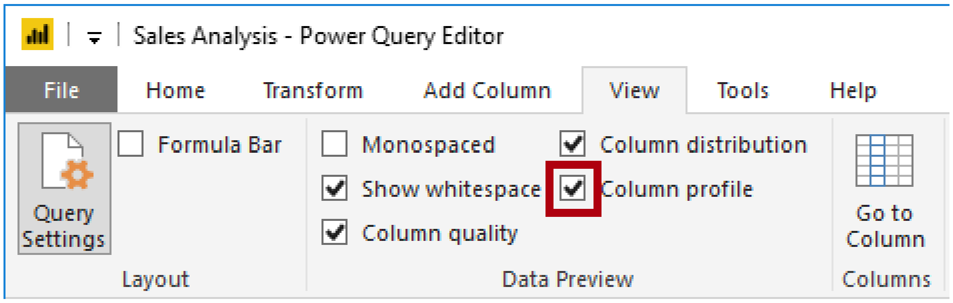


Queries [6]

- DimEmployee
- DimEmployeeSalesTerrit...
- DimProduct
- DimReseller
- DimSalesTerritory
- FactResellerSales

The **DimReseller** table contains one row per reseller. Resellers sell, distribute, or value add to the Adventure Works products.

16. □ To view column values, on the **View** ribbon tab, from inside the **Data Preview** group, check **Column Profile**.



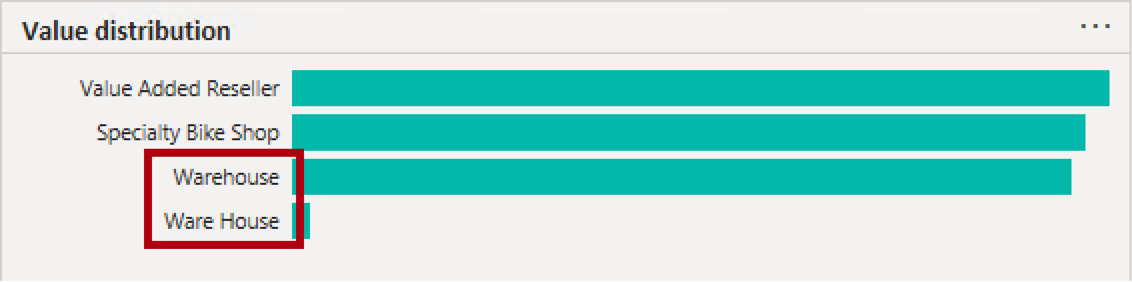
□ Formula Bar
□ Monospaced
☑ Show whitespace
☑ Column quality
☑ Column distribution
☑ Column profile

17. □ Select the **BusinessType** column header.

18. □ Notice the new pane beneath the data preview pane.

19. □ Review the column statistics and value distribution in the data preview pane.

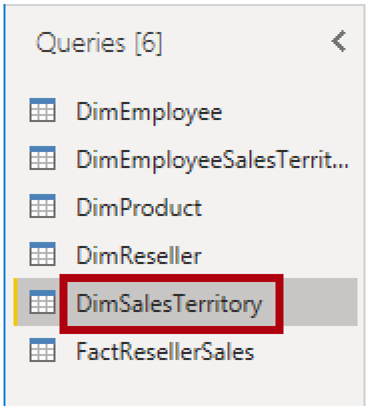
20. □ Notice the data quality issue: there are two labels for warehouse (**Warehouse**, and the misspelled **Ware House**).



21. □ Hover the cursor over the **Ware House** bar, and notice that there are five rows with this value.

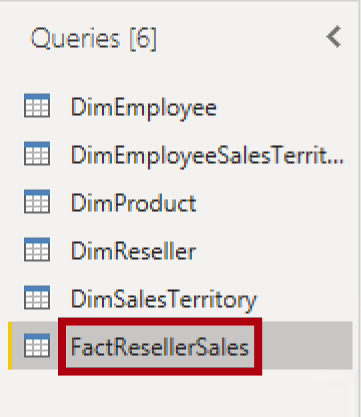
You'll apply a transformation to relabel these five rows in the **Load Data in Power BI Desktop** lab.

22. □ In the **Queries** pane, select the **DimSalesTerritory** query.



The **DimSalesTerritory** table contains one row per sales region, including **Corporate HQ** (headquarters). Regions are assigned to a country, and countries are assigned to groups. In the **Model Data in Power BI Desktop, Part 1** lab, you'll create a hierarchy to support analysis at region, country, or group level.

23. □ In the **Queries** pane, select the **FactResellerSales** query.

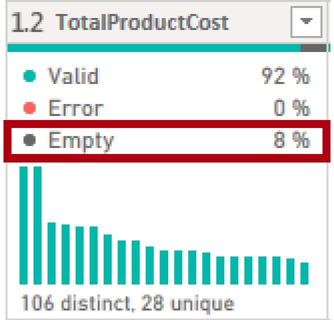


Queries [6]

- DimEmployee
- DimEmployeeSalesTerrit...
- DimProduct
- DimReseller
- DimSalesTerritory
- FactResellerSales

The **FactResellerSales** table contains one row per sales order line—a sales order contains one or more line items.

24. □ Review the column quality for the **TotalProductCost** column, and notice that 8% of the rows are empty.



1.2 TotalProductCost

● Valid	92 %
● Error	0 %
● Empty	8 %

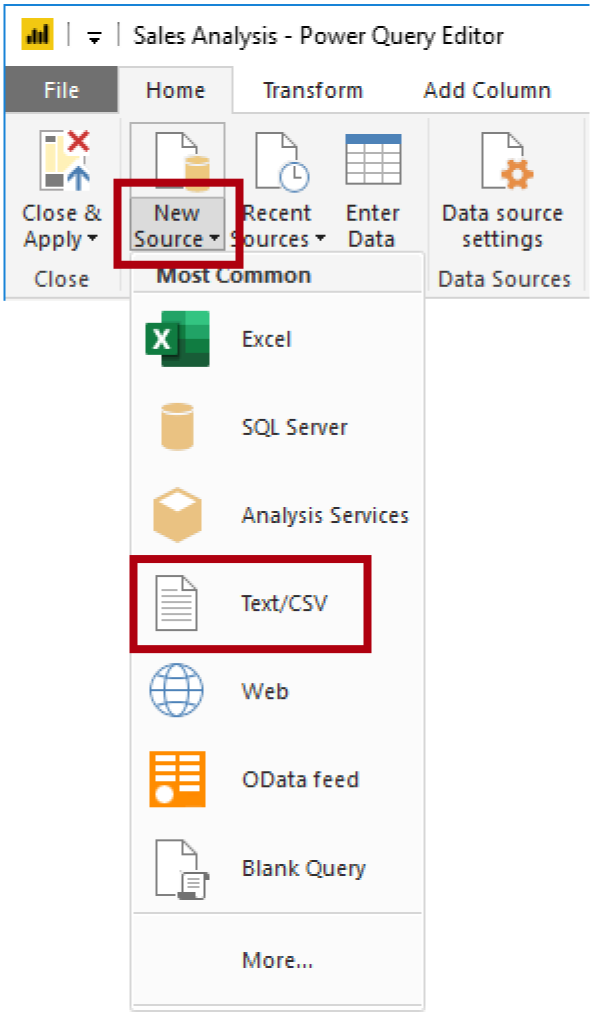
106 distinct, 28 unique

Missing **TotalProductCost** column values is a data quality issue. To address the issue, in the **Load Data in Power BI Desktop** lab, you'll apply transformations to fill in missing values by using the product standard cost, which is stored in the related **DimProduct** table.

Task 5: Get data from a CSV file

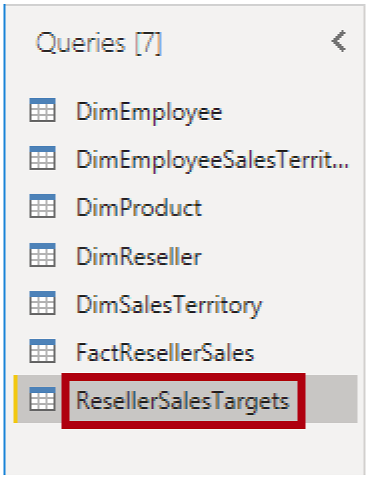
In this task you will create a query based on a CSV file.

1. □ To add a new query, in the **Power Query Editor** window, on the **Home** ribbon tab, from inside the **New Query** group, click the **New Source** down-arrow, and then select **Text/CSV**.



2. □ In the **Open** window, navigate to the **D:\DA100\Resources** folder, and select the **ResellerSalesTargets.csv** file.
3. □ Click **Open**.
4. □ In the **ResellerSalesTargets.csv** window, review the preview data.
5. □ Click **OK**.

1. □ In the **Queries** pane, notice the addition of the **ResellerSalesTargets** query.



Queries [7]

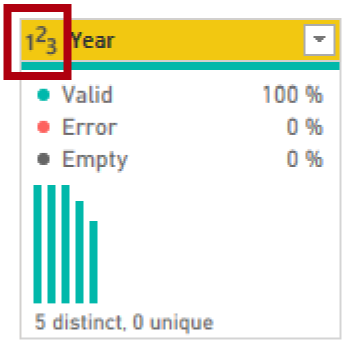
- DimEmployee
- DimEmployeeSalesTerrit...
- DimProduct
- DimReseller
- DimSalesTerritory
- FactResellerSales
- ResellerSalesTargets

The **ResellerSalesTargets** CSV file contains one row per salesperson, per year. Each row records 12 monthly sales targets (expressed in thousands). Note that the business year for the Adventure Works company commences on July 1.

2. □ Notice that no column contains empty values.

When there isn't a monthly sales target, a hyphen character is stored instead.

3. □ Review the icons in each column header, to the left of the column name.



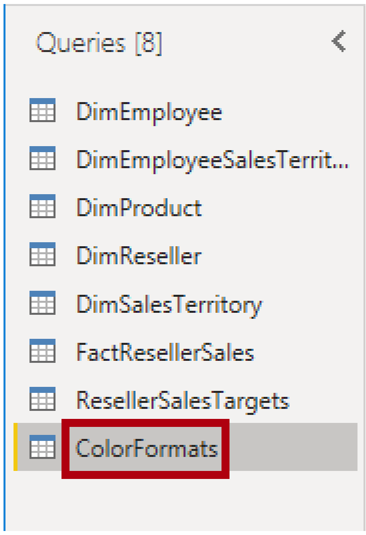
The icons represent the column data type. **123** is whole number, and **ABC** is text.

You'll apply many transformations to achieve a different shaped result consisting of only three columns: **Date**, **EmployeeKey**, and **TargetAmount** in the **Load Data in Power BI Desktop** lab.

Task 6: Get additional data from a CSV file

In this task you will create an additional query based on a different CSV file.

1. □ Use the steps in the previous task to create a query based on the **D:\DA100\Resources\ColorFormats.csv** file.



Queries [8]

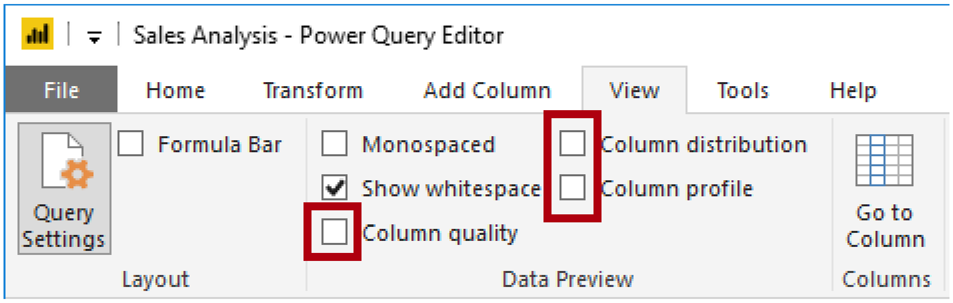
- DimEmployee
- DimEmployeeSalesTerrit...
- DimProduct
- DimReseller
- DimSalesTerritory
- FactResellerSales
- ResellerSalesTargets
- ColorFormats

The **ColorFormats** CSV file contains one row per product color. Each row records the HEX codes to format background and font colors. You'll integrate this data with the **DimProduct** query data in the **Load Data in Power BI Desktop** lab.

Task 7: Finish up

In this task you will complete the lab.

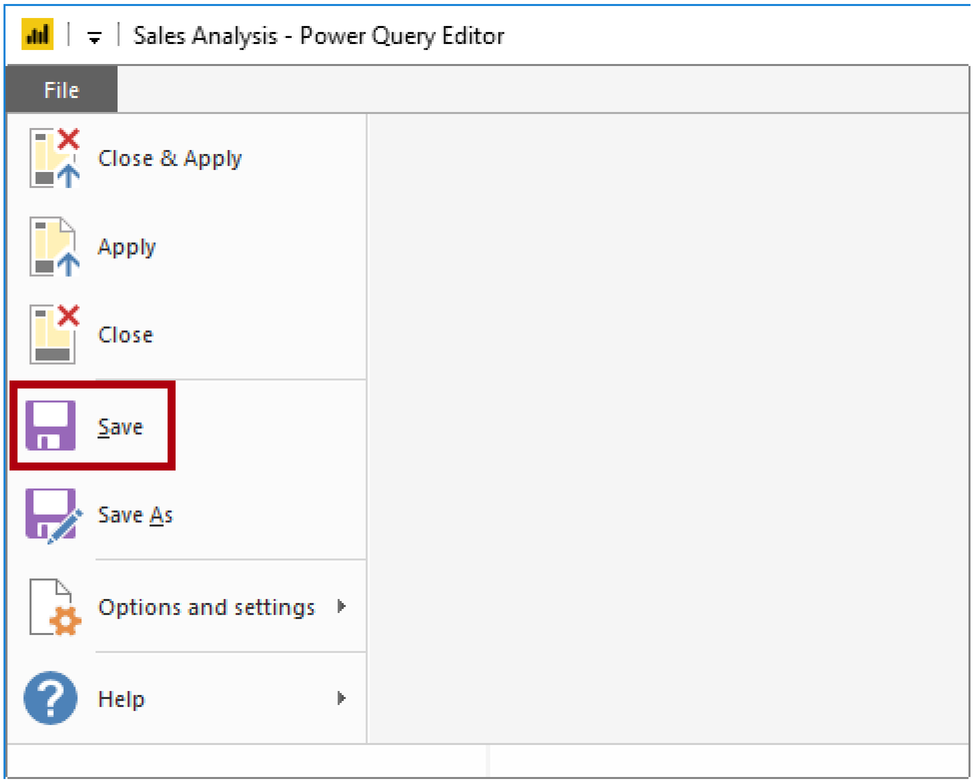
1. □ On the **View** ribbon tab, from inside the **Data Preview** group, uncheck the three data preview options that were previously enabled in this lab:
   - Column quality
   - Column distribution
   - Column profile



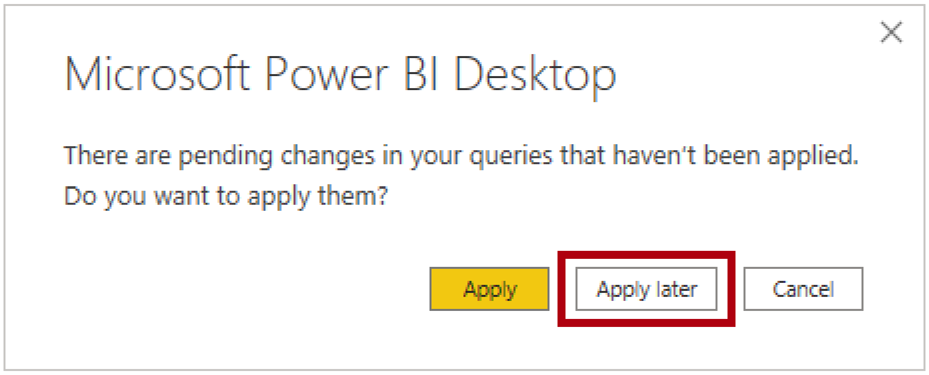
Sales Analysis - Power Query Editor

□ Formula Bar
□ Monospaced
☑ Show whitespace
□ Column quality
□ Column distribution
□ Column profile

2. □ To save the Power BI Desktop file, in the **Power Query Editor** window, on the **File** backstage view, select **Save**.



3. □ When prompted to apply the queries, click **Apply Later**.


Microsoft Power BI Desktop

There are pending changes in your queries that haven't been applied. Do you want to apply them?

Applying the queries will load their data to the data model. You're not ready to do that, as there are many transformations that must be applied first.

4. □ If you intend to start the next lab, leave Power BI Desktop open.

You'll apply various transformations to the queries and then apply the queries to load them to the data model in the **Load Data in Power BI Desktop** lab.

Click **here** to return to the **Table of Contents**.

Load Data in Power BI Desktop

In this lab you will commence apply transformations to each of the queries created in the previous lab. You will then apply the queries to load each as a table to the data model.

In this lab you learn how to:

- Apply various transformations
- Apply queries to load them to the data model

Exercise 1: Load Data

In this exercise you will apply transformations to each of the queries created in the previous lab.

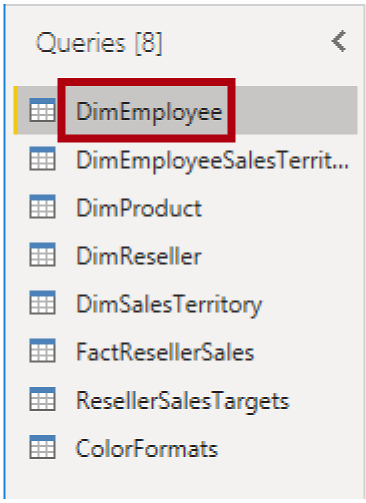
Task 1: Get started

1. □ In the **Open** window, navigate to the **D:\DA100\Labs\02-load-data-with-power-query-in-power-bi-desktop\Starter** folder.
2. □ Select the **Sales Analysis** file.

Task 2: Configure the Salesperson query

In this task you will configure the **Salesperson** query.

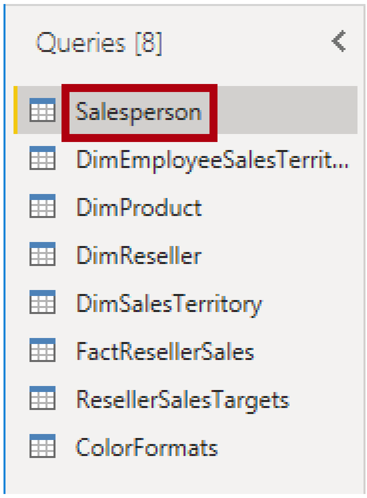
1. □ In the **Power Query Editor** window, in the **Queries** pane, select the **DimEmployee** query.



2. □ To rename the query, in the **Query Settings** pane (located at the right), in the **Name** box, replace the text with **Salesperson**, and then press **Enter**.

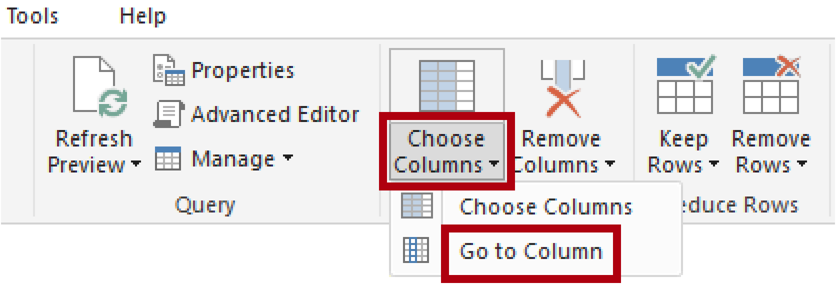
The query name will determine the model table name. It's recommended to define concise, yet friendly, names.

3. □ In the **Queries** pane, verify that the query name has updated.



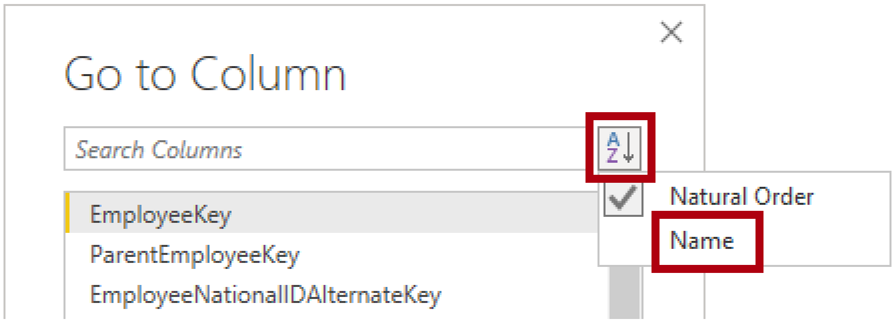
You'll now filter the query rows to retrieve only employees who are salespeople.

4. □ To locate a specific column, on the **Home** ribbon tab, from inside the **Manage Columns** group, click the **Choose Columns** down-arrow, and then select **Go to Column**.



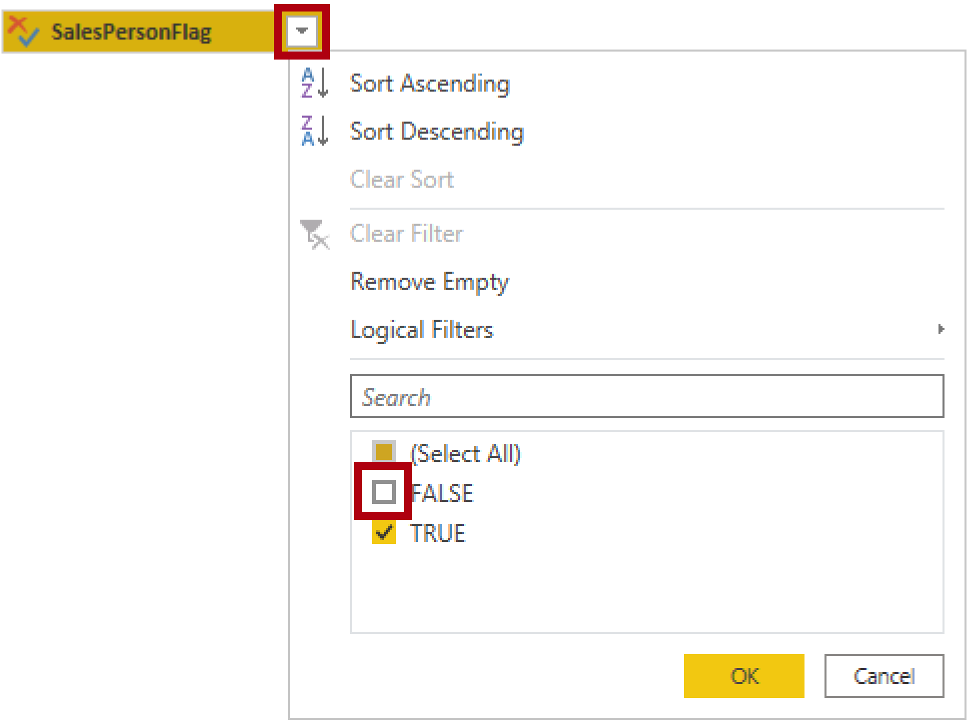
Tip: This technique is useful when a query contains many columns. If there's not too many columns, you can simply horizontally scroll to locate the column of interest.

5. □ In the **Go to Column** window, to order the list by column name, click the **AZ** sort button, and then select **Name**.



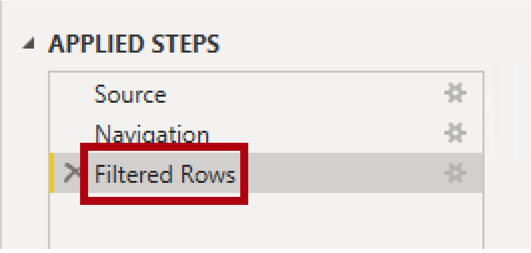
6. □ Select the **SalesPersonFlag** column, and then click **OK**.

7. □ To filter the query, in the **SalesPersonFlag** column header, click the down-arrow, and then uncheck **FALSE**.



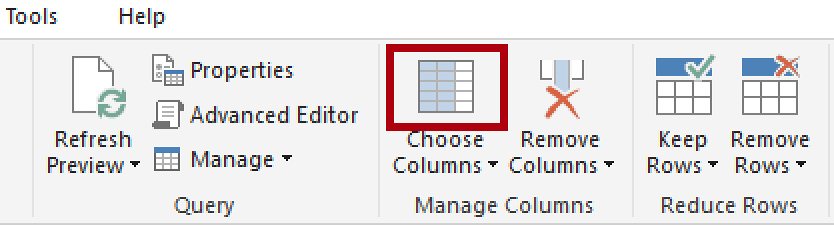
8. □ Click **OK**.

9. □ In the **Query Settings** pane, in the **Applied Steps** list, notice the addition of the **Filtered Rows** step.

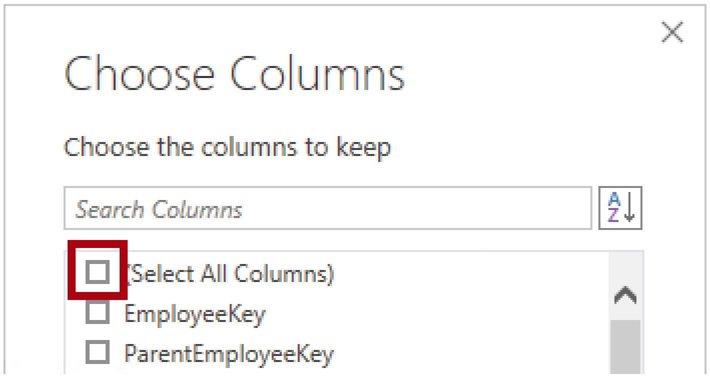


Each transformation you create results in additional step logic. It's possible to edit or delete steps. It's also possible to select a step to preview the query results at that stage of the query transformation.

10. □ To remove columns, on the **Home** ribbon tab, from inside the **Manage Columns** group, click the **Choose Columns** icon.

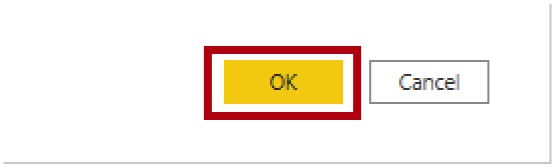


11. □ In the **Choose Columns** window, to uncheck all columns, uncheck the **(Select All Columns)** item.

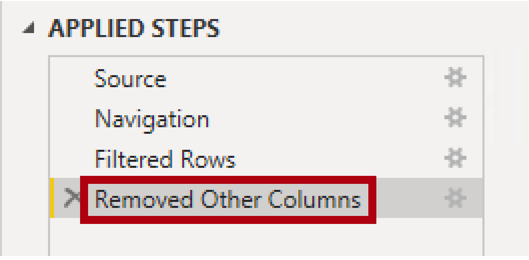


12. □ To include columns, check the following six columns:
    - EmployeeKey
    - EmployeeNationalIDAlternateKey
    - FirstName
    - LastName
    - Title
    - EmailAddress

13. □ Click **OK**.



14. ☐ In the **Applied Steps** list, notice the addition of another query step.

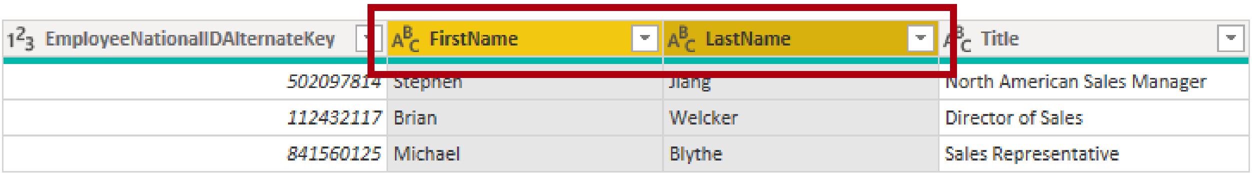


APPLIED STEPS

- Source
- Navigation
- Filtered Rows
- Removed Other Columns

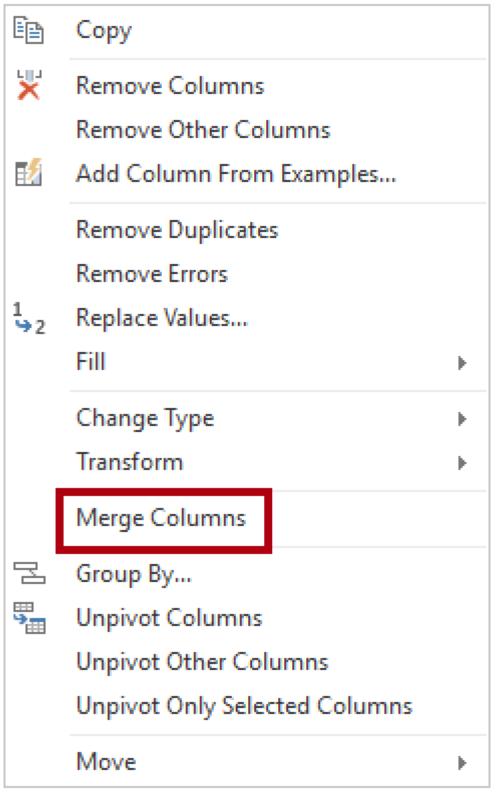
15. ☐ To create a single name column, first select the **FirstName** column header.

16. ☐ While pressing the **Ctrl** key, select the **LastName** column.



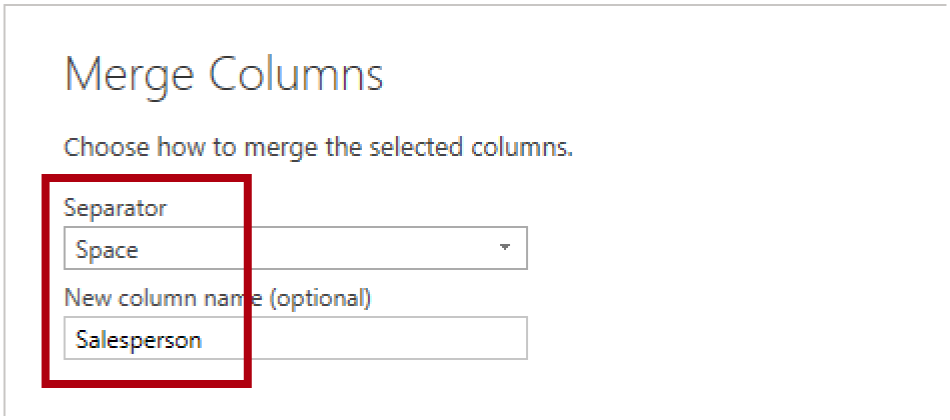
EmployeeNationalIDAlternateKey	FirstName	LastName	Title
502097814	Stephen	Jiang	North American Sales Manager
112432117	Brian	Welcker	Director of Sales
841560125	Michael	Blythe	Sales Representative

17. ☐ Right-click either of the select column headers, and then in the context menu, select **Merge Columns**.



Many common transformations can be applied by right-clicking the column header, and then choosing them from the context menu. Note, however, more transformations are available in the ribbon.

18. □ In the **Merge Columns** window, in the **Separator** dropdown list, select **Space**.
19. □ In the **New Column Name** box, replace the text with **Salesperson**.



Merge Columns

Choose how to merge the selected columns.

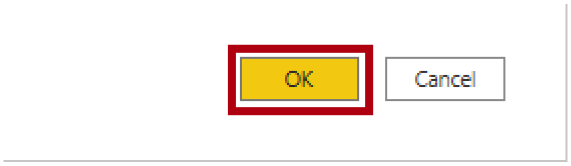
Separator

Space

New column name (optional)

Salesperson

20. □ Click **OK**.



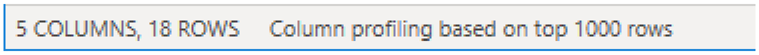
21. □ To rename the **EmployeeNationalIDAlternateKey** column, double-click the **EmployeeNationalIDAlternateKey** column header.
22. □ Replace the text with **EmployeeID**, and then press **Enter**.

    Important: When instructed to rename columns, it's important that you rename them exactly as described.

23. □ Use the previous steps to rename the **EmailAddress** column to **UPN**.

    UPN is an acronym for ***User Principal Name****. The values in this column will be used when you configure row-level security in the* ***Model Data in Power BI Desktop, Part 2*** *lab.*

24. □ At the bottom-left, in the status bar, verify that the query has five columns and 18 rows.



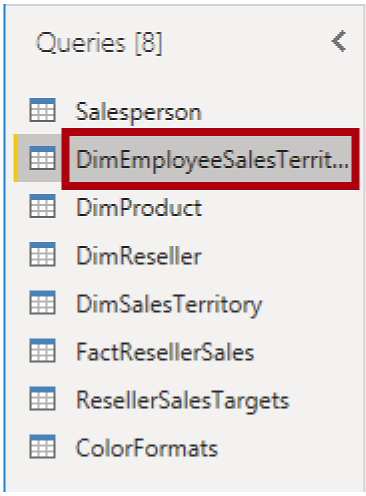
5 COLUMNS, 18 ROWS   Column profiling based on top 1000 rows

Important: It's important that you do not proceed if your query does not produce the correct result—it won't be possible to complete later labs. If the query columns or rows don't match, refer back to the steps in this task to fix any problems.

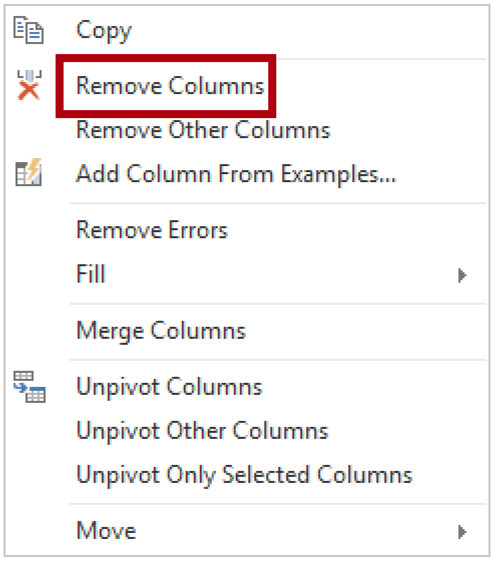
Task 3: Configure the SalespersonRegion query

In this task you will configure the **SalespersonRegion** query.

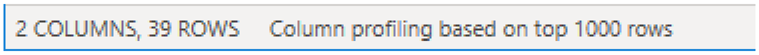
1. □ In the **Queries** pane, select the **DimEmployeeSalesTerritory** query.



2. ☐ In the **Query Settings** pane, rename the query to **SalespersonRegion**.
3. ☐ To remove the last two columns, first select the **DimEmployee** column header.
4. ☐ While pressing the **Ctrl** key, select the **DimSalesTerritory** column header.
5. ☐ Right-click either of the select column headers, and then in the context menu, select **Remove Columns**.



6. ☐ In the status bar, verify that the query has two columns and 39 rows.



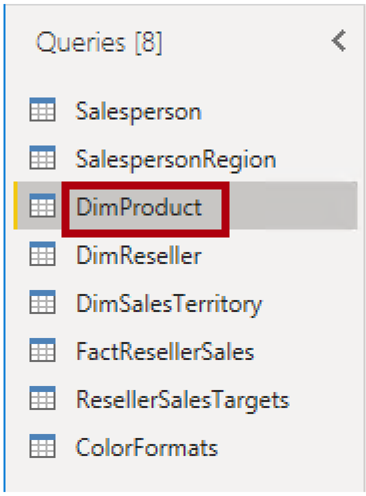
2 COLUMNS, 39 ROWS   Column profiling based on top 1000 rows

Task 4: Configure the Product query

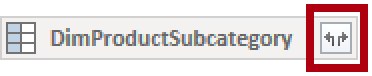
In this task you will configure the **Product** query.

Important: When detailed instructions have already been provided, the lab steps will now provide more concise instructions. If you need the detailed instructions, you can refer back to the steps of previous tasks.

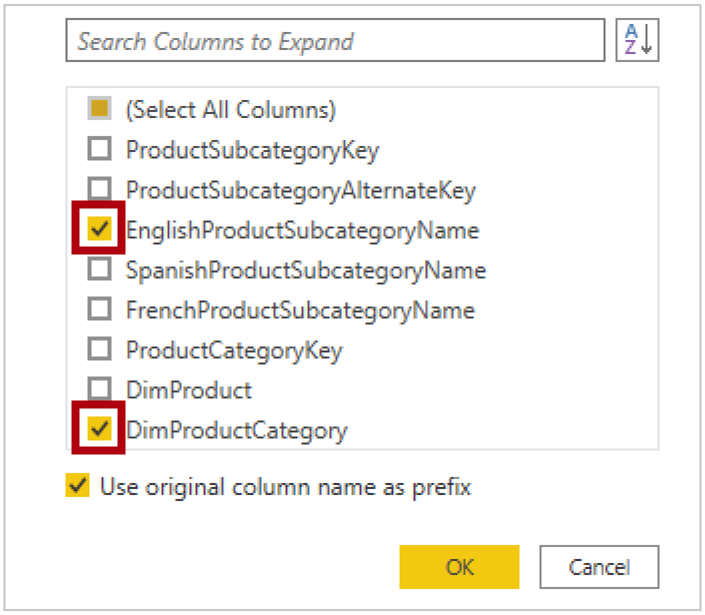
1. □ Select the **DimProduct** query.



2. □ Rename the query to **Product**.
3. □ Locate the **FinishedGoodsFlag** column, and then filter the column to retrieve products that are finished goods (i.e. TRUE).
4. □ Remove all columns, except the following:
   - ProductKey
   - EnglishProductName
   - StandardCost
   - Color
   - DimProductSubcategory
5. □ Notice that the **DimProductSubcategory** column represents a related table (it contains **Value** links).
6. □ In the **DimProductSubcategory** column header, at the right of the column name, click the expand button.



7. □ To uncheck all columns, uncheck the **(Select All Columns)** item.
8. □ Check the **EnglishProductSubcategoryName** and **DimProductCategory** columns.



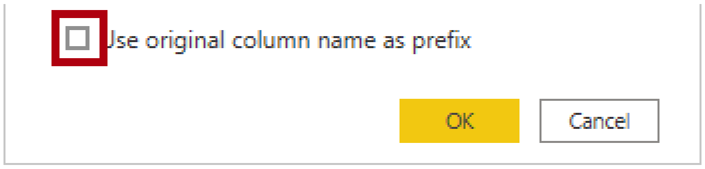
Search Columns to Expand

■ (Select All Columns)
□ ProductSubcategoryKey
□ ProductSubcategoryAlternateKey
☑ EnglishProductSubcategoryName
□ SpanishProductSubcategoryName
□ FrenchProductSubcategoryName
□ ProductCategoryKey
□ DimProduct
☑ DimProductCategory

☑ Use original column name as prefix

By selecting these two columns, a transformation will be applied to join to the ***DimProductSubcategory*** *table, and then include these columns. The* ***DimProductCategory*** *column is, in fact, another related table in the data source.*

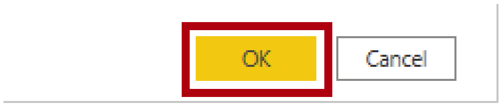
9. □ Uncheck the **Use Original Column Name as Prefix** checkbox.



□ Use original column name as prefix

Query column names must always be unique. If left checked, this checkbox would prefix each column with the expanded column name (in this case ***DimProductSubcategory****). Because it's known that the selected column names don't collide with column names in the* ***Product*** *query, the option is deselected.*

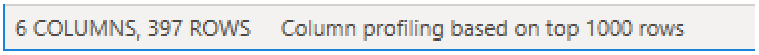
10. □ Click **OK**.



11. □ Notice that the transformation resulted in the addition of two columns, and that the **DimProductSubcategory** column has been removed.

12. □ Expand the **DimProductCategory** column, and then introduce only the **EnglishProductCategoryName** column.

13. □ Rename the following four columns:
    - **EnglishProductName** to **Product**
    - **StandardCost** to **Standard Cost** (include a space)
    - **EnglishProductSubcategoryName** to **Subcategory**
    - **EnglishProductCategoryName** to **Category**
14. □ In the status bar, verify that the query has six columns and 397 rows.

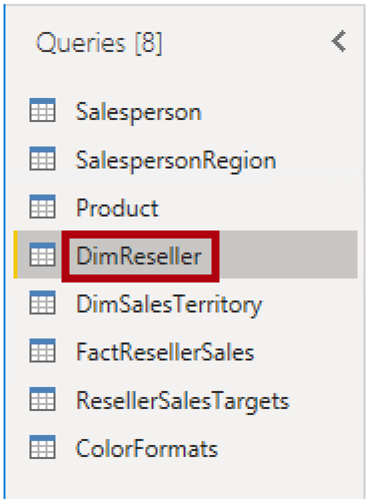


```
6 COLUMNS, 397 ROWS    Column profiling based on top 1000 rows
```

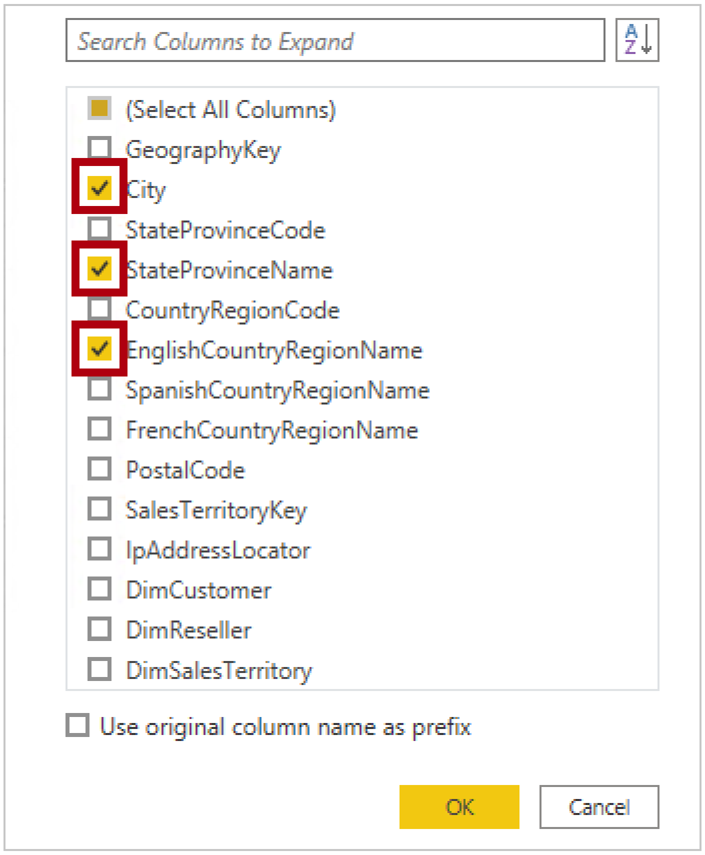
Task 5: Configure the Reseller query

In this task you will configure the **Reseller** query.

1. □ Select the **DimReseller** query.



2. □ Rename the query to **Reseller**.
3. □ Remove all columns, except the following:
    - ResellerKey
    - BusinessType
    - ResellerName
    - DimGeography
4. □ Expand the **DimGeography** column, to include only the following three columns:
    - City
    - StateProvinceName
    - EnglishCountryRegionName

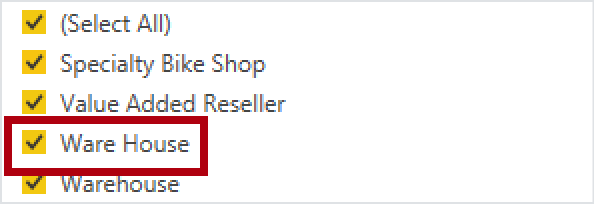


Search Columns to Expand

■ (Select All Columns)
□ GeographyKey
☑ City
□ StateProvinceCode
☑ StateProvinceName
□ CountryRegionCode
☑ EnglishCountryRegionName
□ SpanishCountryRegionName
□ FrenchCountryRegionName
□ PostalCode
□ SalesTerritoryKey
□ IpAddressLocator
□ DimCustomer
□ DimReseller
□ DimSalesTerritory

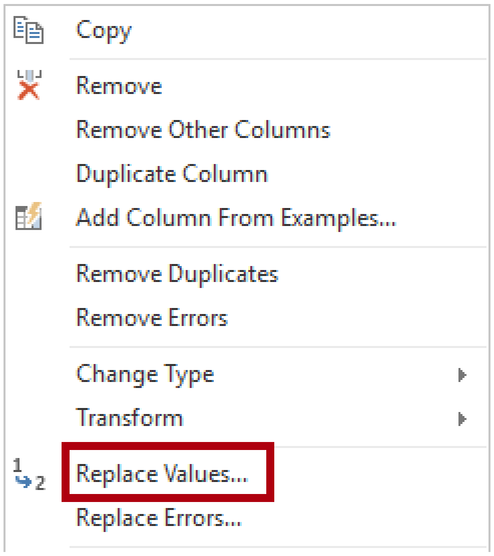
□ Use original column name as prefix

5. □ In the **Business Type** column header, click the down-arrow, and then review the distinct column values, and notice the incorrect spelling of warehouse.

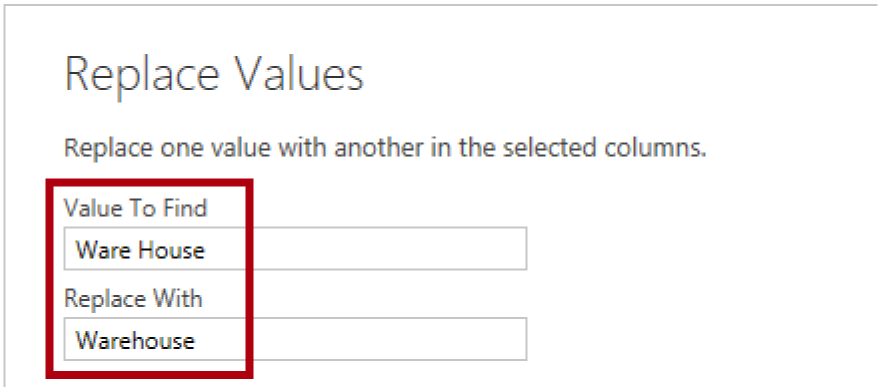


☑ (Select All)
☑ Specialty Bike Shop
☑ Value Added Reseller
☑ Ware House
☑ Warehouse

1. □ Right-click the **Business Type** column header, and then select **Replace Values**.



2. □ In the **Replace Values** window, configure the following values:
   - In the **Value to Find** box, enter **Ware House**
   - In the **Replace With** box, enter **Warehouse**



Replace Values

Replace one value with another in the selected columns.

Value To Find

Ware House

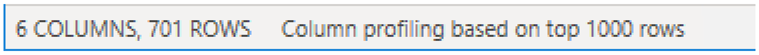
Replace With

Warehouse

3. □ Click **OK**.

4. □ Rename the following four columns:
   - **BusinessType** to **Business Type** (include a space)
   - **ResellerName** to **Reseller**
   - **StateProvinceName** to **State-Province**
   - **EnglishCountryRegionName** to **Country-Region**

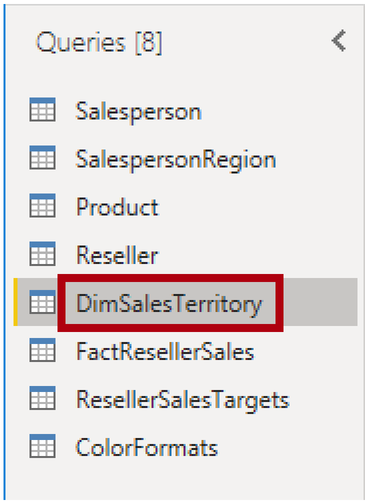
5. □ In the status bar, verify that the query has six columns and 701 rows.


6 COLUMNS, 701 ROWS Column profiling based on top 1000 rows

Task 6: Configure the Region query

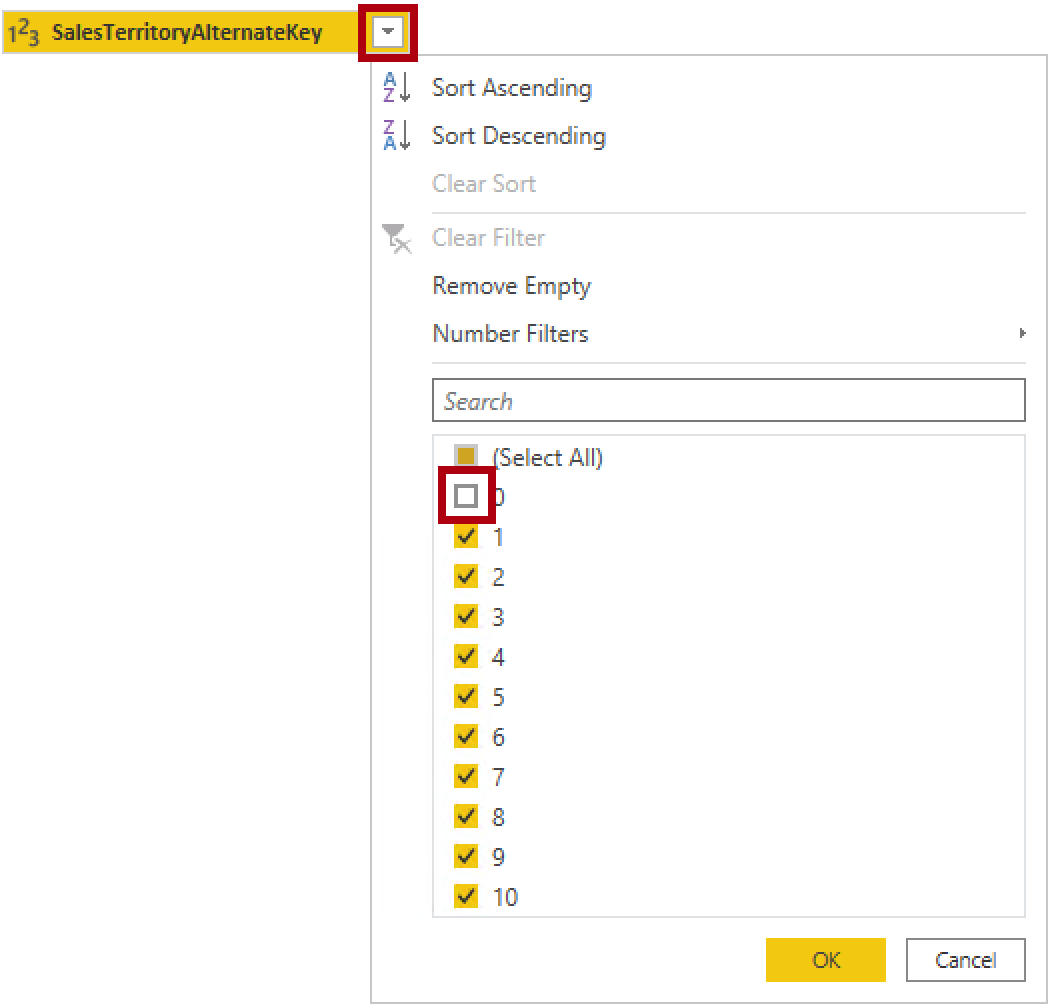
In this task you will configure the **Region** query.

1. □ Select the **DimSalesTerritory** query.

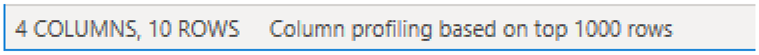

Queries [8]

- Salesperson
- SalespersonRegion
- Product
- Reseller
- DimSalesTerritory
- FactResellerSales
- ResellerSalesTargets
- ColorFormats

2. □ Rename the query to **Region**.
3. □ Apply a filter to the **SalesTerritoryAlternateKey** column to remove the value 0 (zero).



4. □ Remove all columns, except the following:
   - SalesTerritoryKey
   - SalesTerritoryRegion
   - SalesTerritoryCountry
   - SalesTerritoryGroup
5. □ Rename the following three columns:
   - **SalesTerritoryRegion** to **Region**
   - **SalesTerritoryCountry** to **Country**
   - **SalesTerritoryGroup** to **Group**
6. □ In the status bar, verify that the query has four columns and 10 rows.

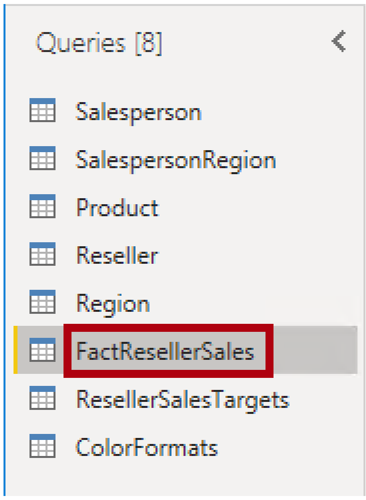


4 COLUMNS, 10 ROWS   Column profiling based on top 1000 rows

Task 7: Configure the Sales query

In this task you will configure the **Sales** query.

1. □ Select the **FactResellerSales** query.



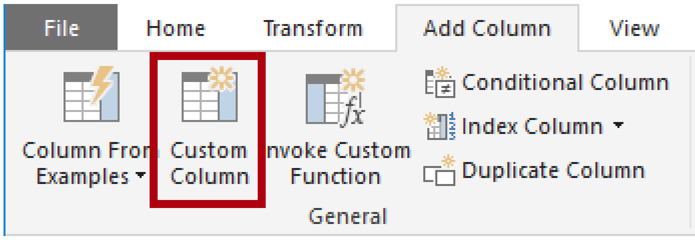
Queries [8]

- Salesperson
- SalespersonRegion
- Product
- Reseller
- Region
- FactResellerSales
- ResellerSalesTargets
- ColorFormats

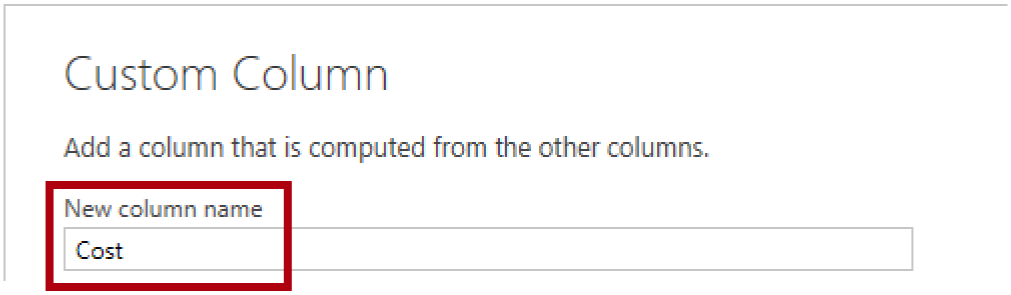
2. □ Rename the query to **Sales**.
3. □ Remove all columns, except the following:
    - SalesOrderNumber
    - OrderDate
    - ProductKey
    - ResellerKey
    - EmployeeKey
    - SalesTerritoryKey
    - OrderQuantity
    - UnitPrice
    - TotalProductCost
    - SalesAmount
    - DimProduct

*You may recall in the **Prepare Data in Power BI Desktop** lab that a small percentage of **FactResellerSales** rows had missing **TotalProductCost** values. The **DimProduct** column has been included to retrieve the product standard cost column to assist fixing the missing values.*

4. □ Expand the **DimProduct** column, uncheck all columns, and then include only the **StandardCost** column.
5. □ To create a custom column, on the **Add Column** ribbon tab, from inside the **General** group, click **Custom Column**.



6. □ In the **Custom Column** window, in the **New Column Name** box, replace the text with **Cost**.



Custom Column

Add a column that is computed from the other columns.

New column name

Cost

7. □ In the **Custom Column Formula** box, enter the following expression (after the equals symbol):

8. □ For your convenience, you can copy the expression from the **D:\DA100\Labs\load-data-with-power-query-in-power-bi-desktop\Assets\Snippets.txt** file.

Power Query

```
   if [TotalProductCost] = null then [OrderQuantity] * [StandardCost] else
[TotalProductCost]
```

This expression tests if the ***TotalProductCost*** *value is missing. If it is, produces a value by multiplying the* ***OrderQuantity*** *value by the* ***StandardCost*** *value; otherwise, it uses the existing* ***TotalProductCost*** *value.*

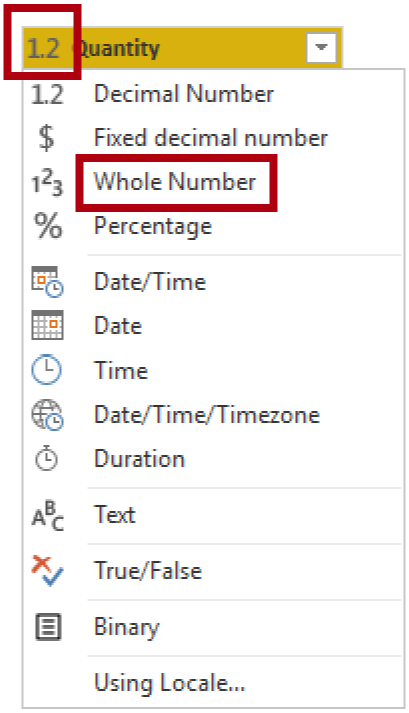
1. □ Click **OK**.

2. □ Remove the following two columns:
   - TotalProductCost
   - StandardCost

3. □ Rename the following three columns:

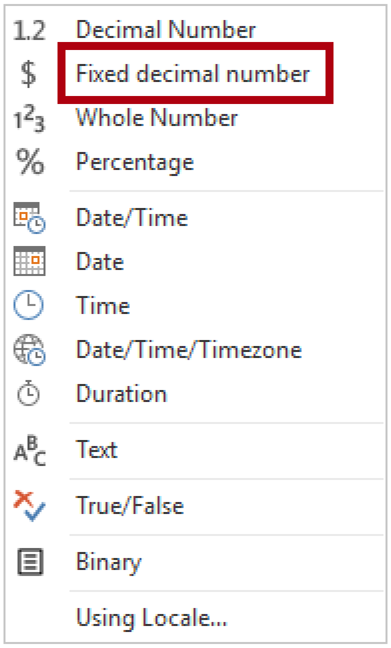
- **OrderQuantity** to **Quantity**
- **UnitPrice** to **Unit Price** (include a space)
- **SalesAmount** to **Sales**

4. □ To modify the column data type, in the **Quantity** column header, at the left of the column name, click the **1.2** icon, and then select **Whole Number**.



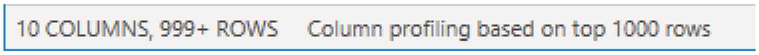
Configuring the correct data type is important. When the column contains numeric value, it's also important to choose the correct type if you expect to perform mathematic calculations.

5. □ Modify the following three column data types to **Fixed Decimal Number**.
   - Unit Price
   - Sales
   - Cost



*The **fixed decimal** number data type stores values with **full precision**, and so requires more storage space that decimal number. It's important to use the fixed decimal number type for financial values, or rates (like exchange rates).*

6. □ In the status bar, verify that the query has 10 columns and 999+ rows.



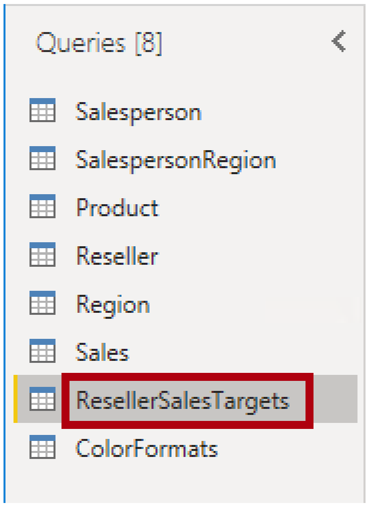
10 COLUMNS, 999+ ROWS   Column profiling based on top 1000 rows

A maximum of 1000 rows will be loaded as preview data for each query.

Task 8: Configure the Targets query

In this task you will configure the **Targets** query.

1. □ Select the **ResellerSalesTargets** query.

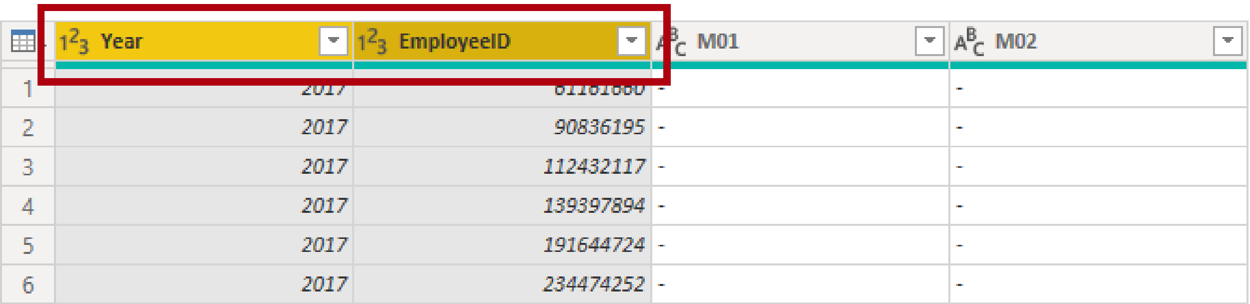


Queries [8]

- Salesperson
- SalespersonRegion
- Product
- Reseller
- Region
- Sales
- ResellerSalesTargets
- ColorFormats

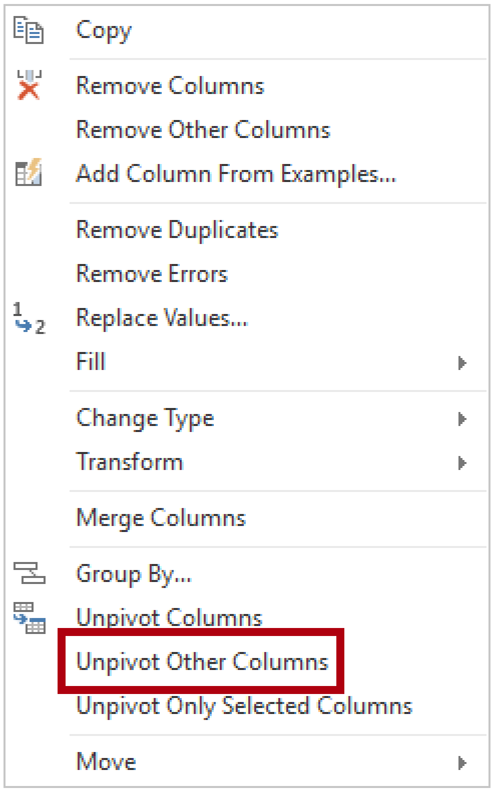
2. ☐ Rename the query to **Targets**.

3. ☐ To unpivot the 12 month columns (**M01-M12**), first multi-select the **Year** and **EmployeeID** column headers.



	Year	EmployeeID	M01	M02
1	2017	81161880	-	-
2	2017	90836195	-	-
3	2017	112432117	-	-
4	2017	139397894	-	-
5	2017	191644724	-	-
6	2017	234474252	-	-

4. ☐ Right-click either of the select column headers, and then in the context menu, select **Unpivot Other Columns**.



5. □ Notice that the column names now appear in the **Attribute** column, and the values appear in the **Value** column.

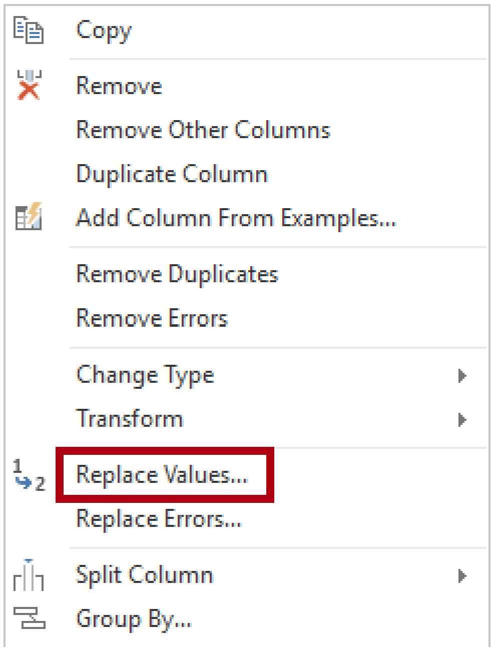
6. □ Apply a filter to the **Value** column to remove hyphen (-) values.

   You may recall that the hyphen character was used in the source CSV file to represent zero (0).

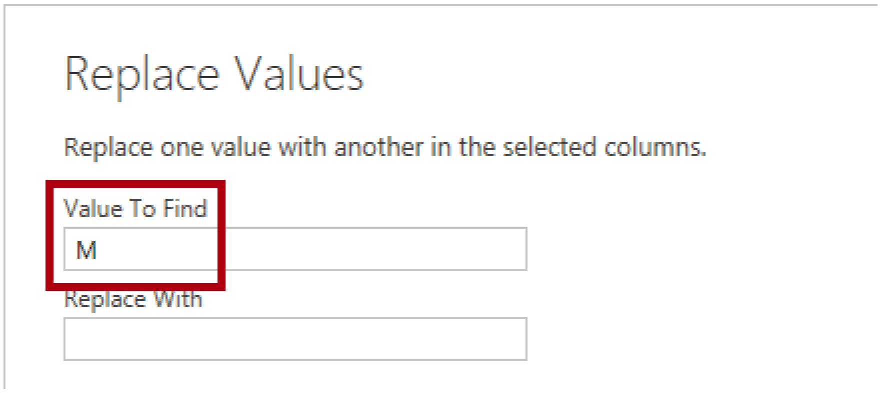
7. □ Rename the following two columns:
   - **Attribute** to **MonthNumber** (there is no space between the two words—it will be removed later)
   - **Value** to **Target**

   You'll now apply transformations to produce a date column. The date will be derived from the ***Year*** *and* ***MonthNumber*** *columns. You'll create the column by using the* ***Columns From Examples*** *feature.*

8. □ To prepare the **MonthNumber** column values, right-click the **MonthNumber** column header, and then select **Replace Values**.



9. □ In the **Replace Values** window, in the **Value To Find** box, enter **M**.



Replace Values

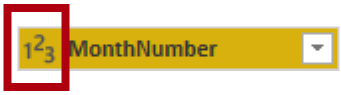
Replace one value with another in the selected columns.

Value To Find
M

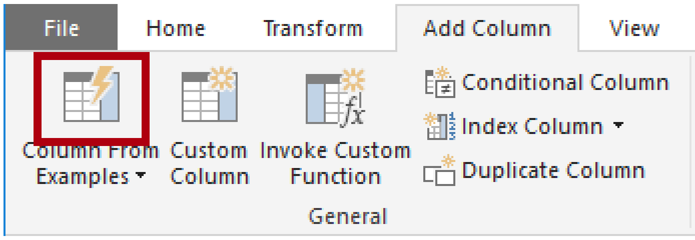
Replace With

10. □ Click **OK**.

11. □ Modify the **MonthNumber** column data type to **Whole Number**.



12. □ On the **Add Column** ribbon tab, from inside the **General** group, click The **Column From Examples** icon.



13. □ Notice that the first row is for year **2017** and month number **7**.

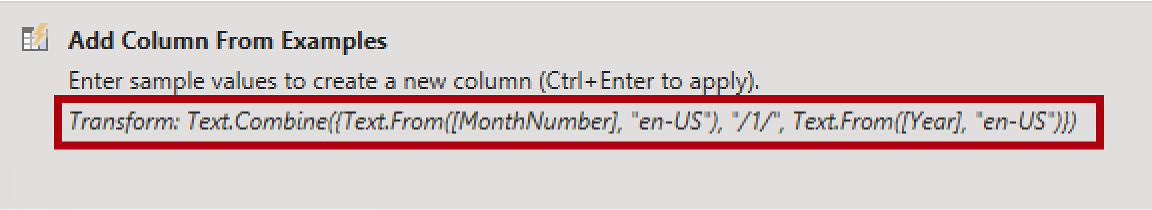
14. □ In the **Column1** column, in the first grid cell, commence entering **7/1/2017**, and then press **Enter**.

The virtual machine uses US regional settings, so this date is in fact July 1, 2017.

15. □ Notice that the grid cells update with predicted values.

*The feature has accurately predicted that you are combining values from the **Year** and **MonthNumber** columns.*

16. □ Notice also the formula presented above the query grid.



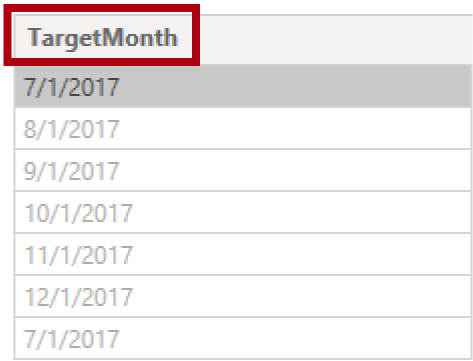
Add Column From Examples

Enter sample values to create a new column (Ctrl+Enter to apply).

Transform: Text.Combine({Text.From([MonthNumber], "en-US"), "/1/", Text.From([Year], "en-US")})

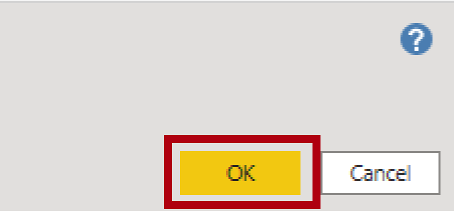
17. □ To rename the new column, double-click the **Merged** column header.

18. □ Rename the column as **TargetMonth**.



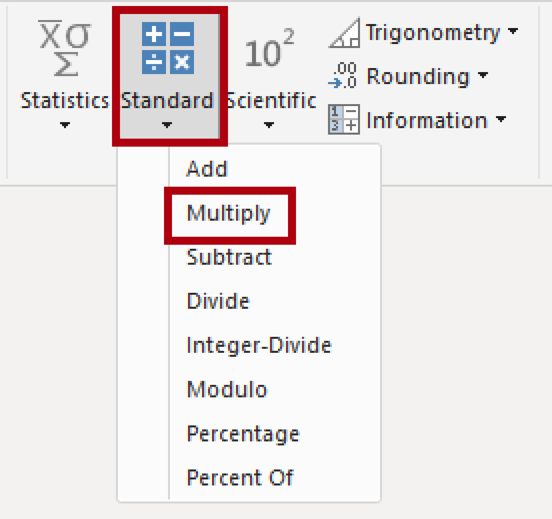
TargetMonth
7/1/2017
8/1/2017
9/1/2017
10/1/2017
11/1/2017
12/1/2017
7/1/2017

19. □ To add the new column, click **OK**.

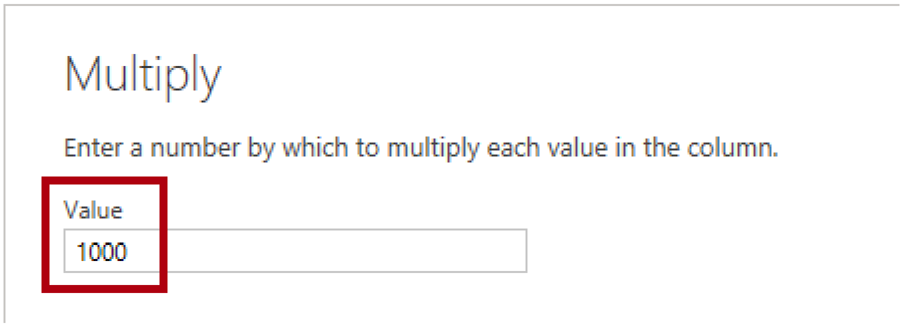


20. □ Remove the following columns:
    - Year
    - MonthNumber
21. □ Modify the following column data types:
    - **Target** as fixed decimal number
    - **TargetMonth** as date
22. □ To multiply the **Target** values by 1000, select the **Target** column header, and then on the **Transform** ribbon tab, from inside the **Number Column** group, click **Standard**, and then select **Multiply**.

    You may recall that the target values were stored as thousands.



23. □ In the **Multiply** window, in the **Value** box, enter **1000**.



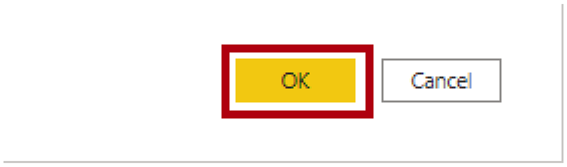
Multiply

Enter a number by which to multiply each value in the column.

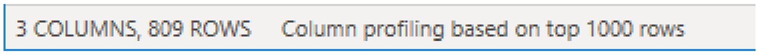
Value

1000

24. □ Click **OK**.



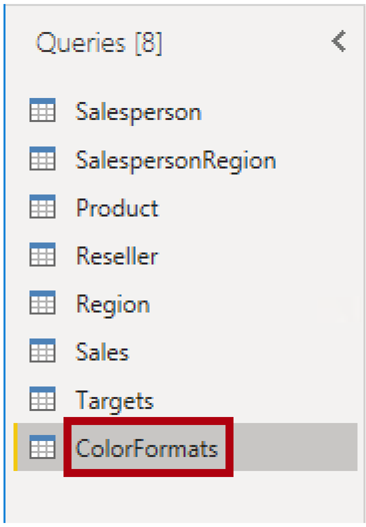
25. □ In the status bar, verify that the query has three columns and 809 rows.


3 COLUMNS, 809 ROWS   Column profiling based on top 1000 rows

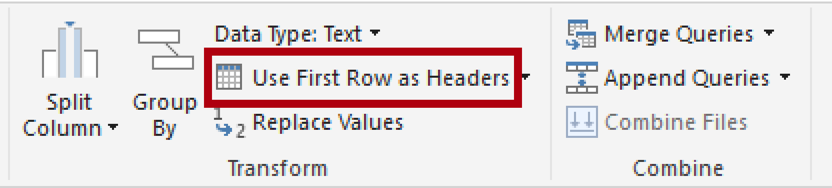
Task 9: Configure the ColorFormats query

In this task you will configure the **ColorFormats** query.

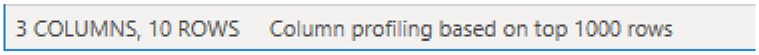
1. □ Select the **ColorFormats** query.



2. □ Notice that the first row contains the column names.
3. □ On the **Home** ribbon tab, from inside the **Transform** group, click **Use First Row as Headers**.



4. □ In the status bar, verify that the query has three columns and 10 rows.

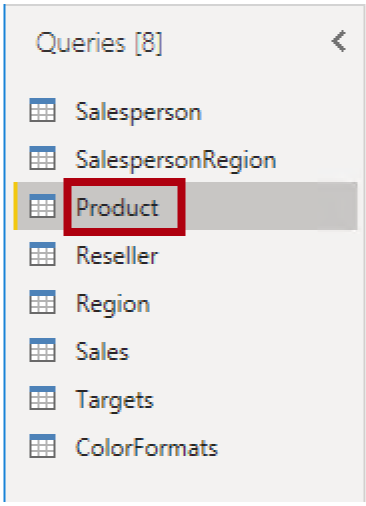


3 COLUMNS, 10 ROWS   Column profiling based on top 1000 rows

Task 10: Update the Product query

In this task you will update the **Product** query by merging the **ColorFormats** query.

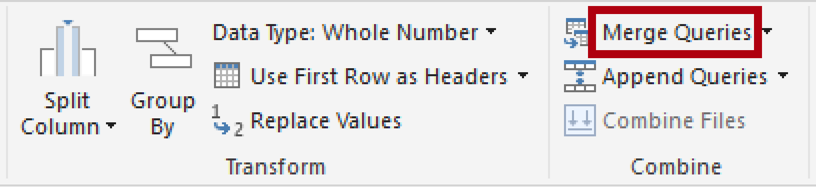
1. □ Select the **Product** query.



Queries [8]

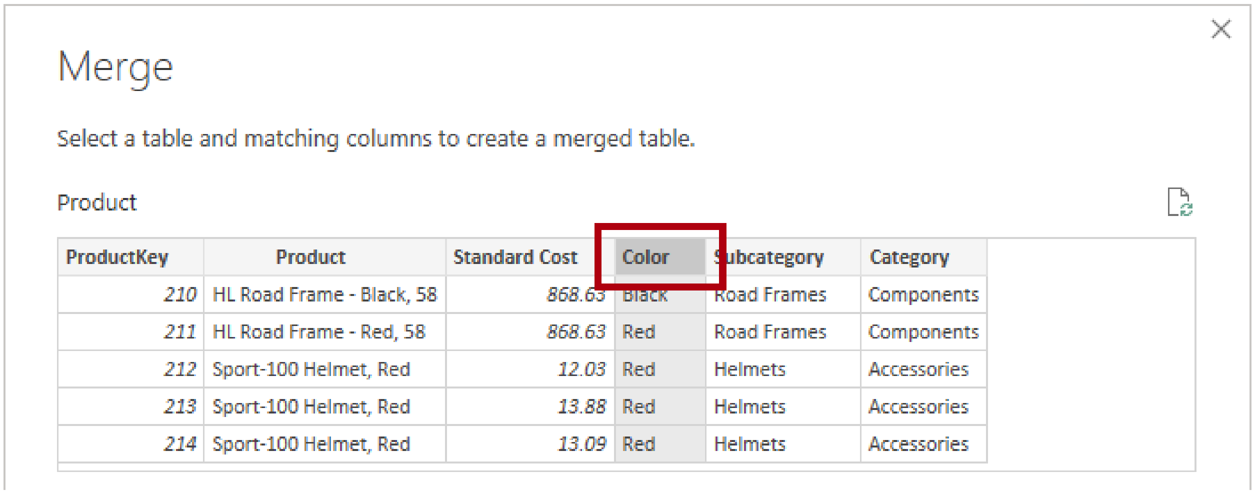
- Salesperson
- SalespersonRegion
- Product
- Reseller
- Region
- Sales
- Targets
- ColorFormats

2. □ To merge the **ColorFormats** query, on the **Home** ribbon tab, from inside the **Combine** group, click **Merge Queries**.



Merging queries allows integrating data, in this case from different data sources (SQL Server and a CSV file).

3. □ In the **Merge** window, in the **Product** query grid, select the **Color** column header.



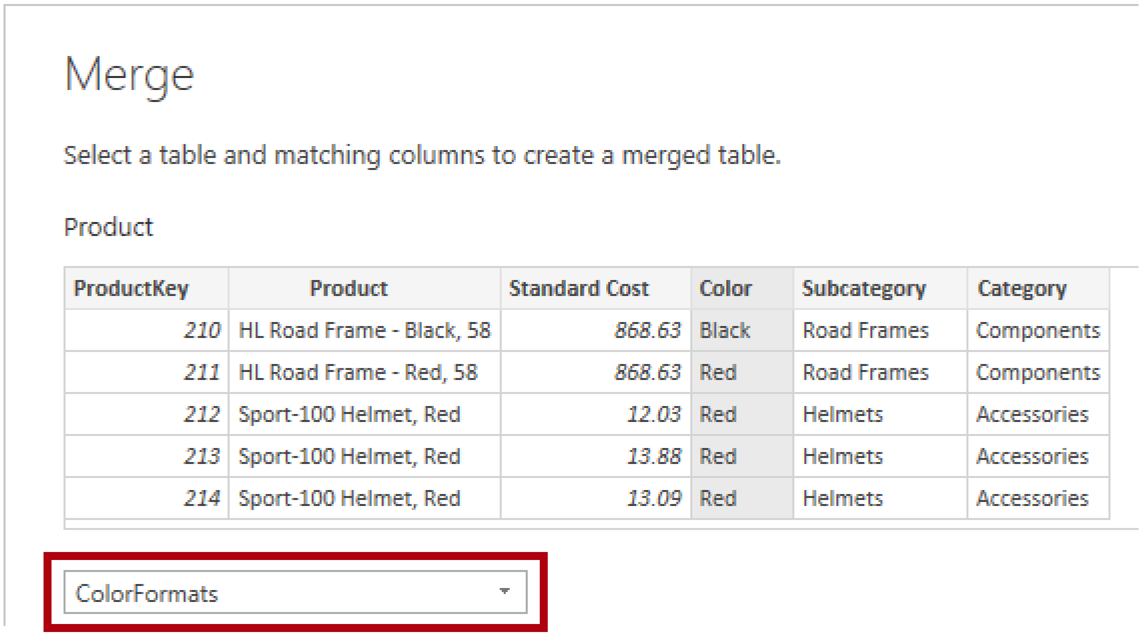
Merge

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

Product

ProductKey	Product	Standard Cost	Color	Subcategory	Category
210	HL Road Frame - Black, 58	868.63	Black	Road Frames	Components
211	HL Road Frame - Red, 58	868.63	Red	Road Frames	Components
212	Sport-100 Helmet, Red	12.03	Red	Helmets	Accessories
213	Sport-100 Helmet, Red	13.88	Red	Helmets	Accessories
214	Sport-100 Helmet, Red	13.09	Red	Helmets	Accessories

4. □ Beneath the **Product** query grid, in the dropdown list, select the **ColorFormats** query.



Merge

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

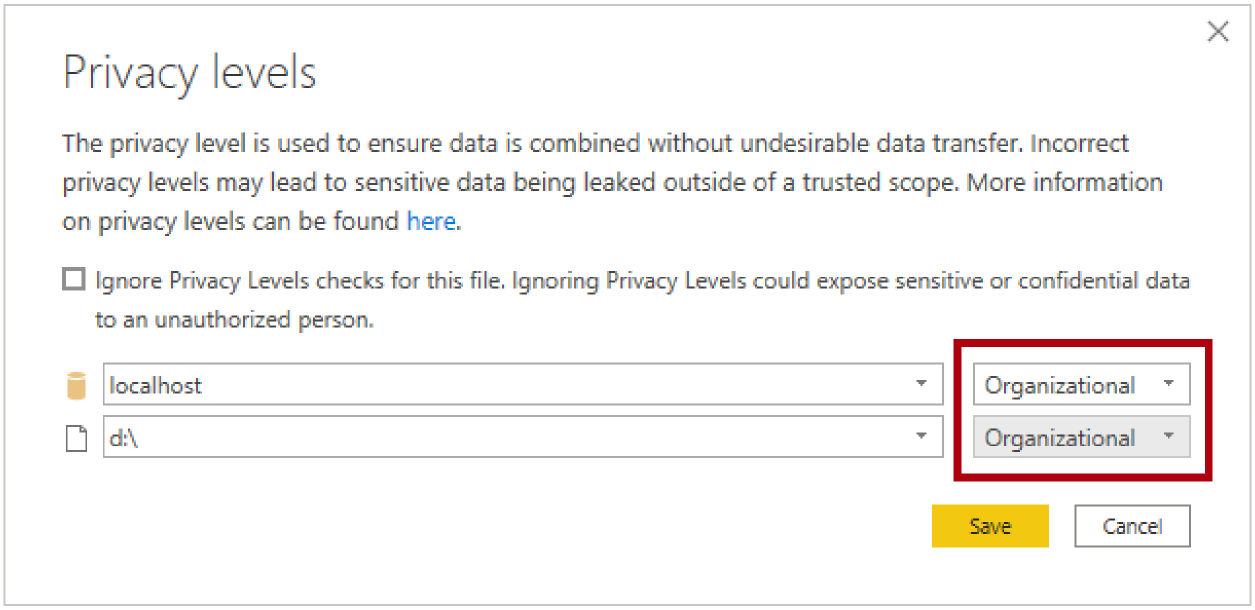
Product

ProductKey	Product	Standard Cost	Color	Subcategory	Category
210	HL Road Frame - Black, 58	868.63	Black	Road Frames	Components
211	HL Road Frame - Red, 58	868.63	Red	Road Frames	Components
212	Sport-100 Helmet, Red	12.03	Red	Helmets	Accessories
213	Sport-100 Helmet, Red	13.88	Red	Helmets	Accessories
214	Sport-100 Helmet, Red	13.09	Red	Helmets	Accessories

ColorFormats

5. □ In the **ColorFormats** query grid, select the **Color** column header.

6. □ When the **Privacy Levels** window opens, for each of the two data sources, in the corresponding dropdown list, select **Organizational**.



Privacy levels

The privacy level is used to ensure data is combined without undesirable data transfer. Incorrect privacy levels may lead to sensitive data being leaked outside of a trusted scope. More information on privacy levels can be found here.

□ Ignore Privacy Levels checks for this file. Ignoring Privacy Levels could expose sensitive or confidential data to an unauthorized person.

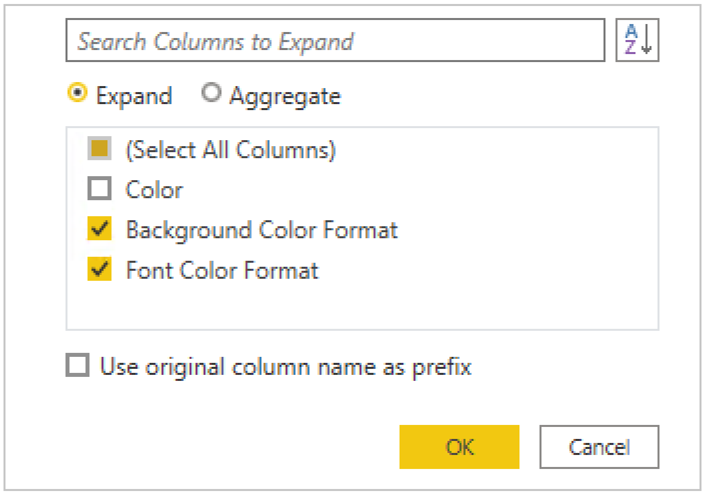
localhost	Organizational
d:\	Organizational

Privacy levels can be configured for data source to determine whether data can be shared between sources. Setting each data source as ***Organizational*** *allows them to share data, if necessary. Note that Private data sources can never be shared with other data sources. It doesn't mean that Private data cannot be shared; it means that the Power Query engine cannot share data between the sources.*

7. □ Click **Save**.

8. □ In the **Merge** window, use the default **Join Kind** - maintaining the selection of Left Outer and click **OK**.

9. □ Expand the **ColorFormats** column to include the following two columns:
   - Background Color Format
   - Font Color Format



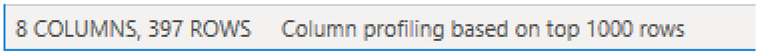
Search Columns to Expand

● Expand ○ Aggregate

■ (Select All Columns)
☐ Color
☑ Background Color Format
☑ Font Color Format

☐ Use original column name as prefix

10. □ In the status bar, verify that the query now has eight columns and 397 rows.

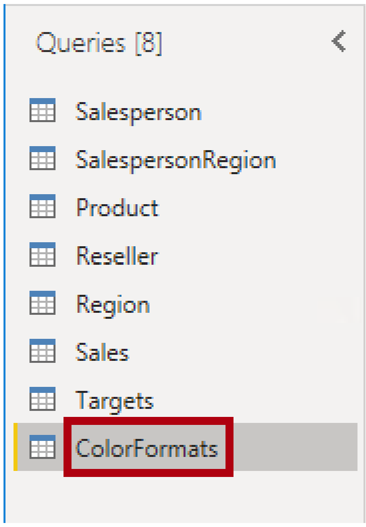


8 COLUMNS, 397 ROWS Column profiling based on top 1000 rows

Task 11: Update the ColorFormats query

In this task you will update the **ColorFormats** to disable its load.

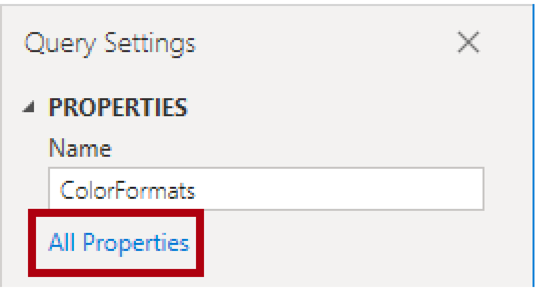
1. □ Select the **ColorFormats** query.



Queries [8]

- Salesperson
- SalespersonRegion
- Product
- Reseller
- Region
- Sales
- Targets
- ColorFormats

2. □ In the **Query Settings** pane, click the **All Properties** link.



Query Settings

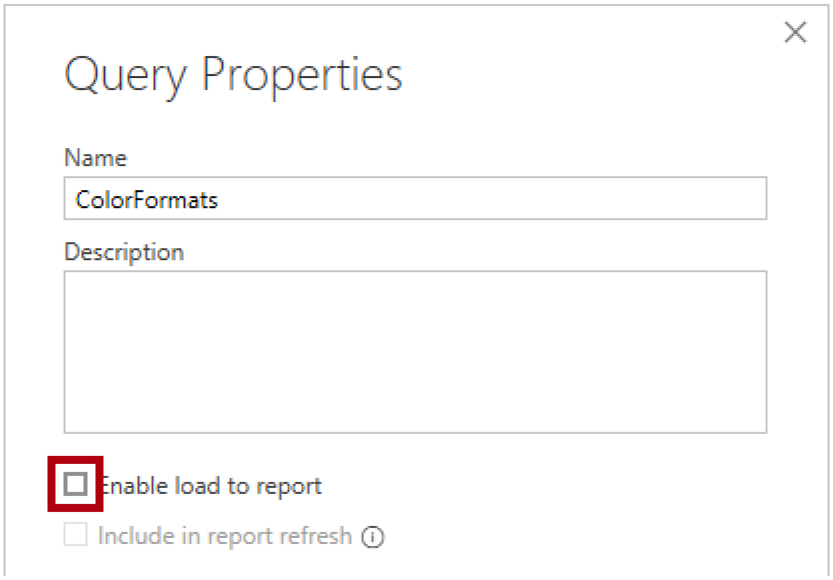
PROPERTIES

Name

ColorFormats

All Properties

3. □ In the **Query Properties** window, uncheck the **Enable Load To Report** checkbox.



Query Properties

Name

ColorFormats

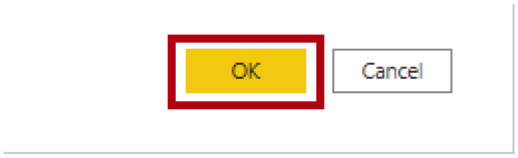
Description

□ Enable load to report

□ Include in report refresh ⓘ

Disabling the load means it will not load as a table to the data model. This is done because the query was merged with the **Product** query, which is enabled to load to the data model.

4. □ Click **OK**.



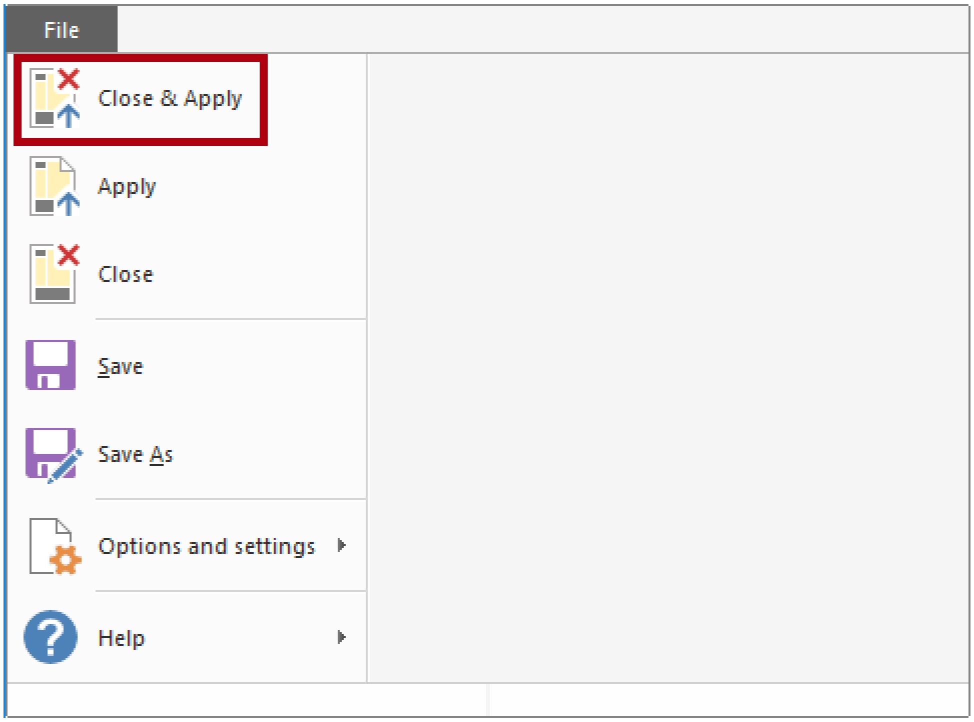
Task 12: Finish up

In this task you will complete the lab.

1. □ Verify that you have eight queries, correctly named as follows:
   - Salesperson
   - SalespersonRegion

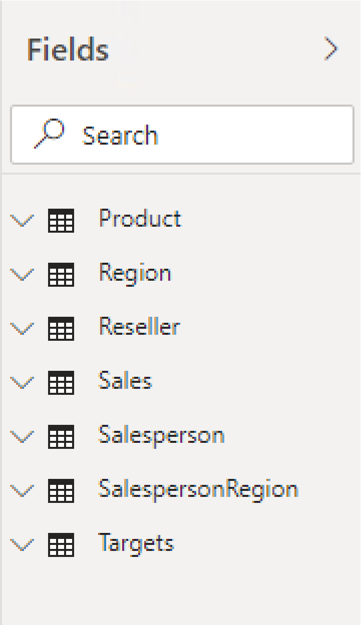
- Product
- Reseller
- Region
- Sales
- Targets
- ColorFormats (which will not load to the data model)

2. ☐ To load the data model, on the **File** backstage view, select **Close & Apply**.



All load-enabled queries are now loaded to the data model.

3. ☐ In the **Fields** pane (located at the right), notice the seven tables loaded to the data model.



4. □ Save the Power BI Desktop file.
5. □ If you intend to start the next lab, leave Power BI Desktop open.

You'll configure data model tables and relationships in the ***Model Data in Power BI Desktop, Part 1*** *lab.*

Congratulations!

Model Data in Power BI Desktop, Part 1

In this lab you will commence developing the data model. It will involve creating relationships between tables, and then configuring table and column properties to improve the friendliness and usability of the data model. You will also create hierarchies and create quick measures.

In this lab you learn how to:

- Create model relationships
- Configure table and column properties
- Create hierarchies
- Create quick measures

Exercise 1: Create Model Relationships

In this exercise you will create model relationships.

Task 1: Get started

In this task you will setup the environment for the lab.

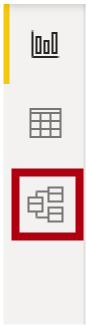
Important: If you are continuing on from the previous lab (and you completed that lab successfully), do not complete this task; instead, continue from the next task.

1. □ In the **Open** window, navigate to the **D:\DA100\Labs\03-configure-data-model-in-power-bi-desktop\Starter** folder.
2. □ Select the **Sales Analysis** file.

Task 2: Create model relationships

In this task you will create model relationships.

1. □ In Power BI Desktop, at the left, click the **Model** view icon.

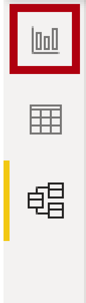


2. □ If you do not see all seven tables, scroll horizontally to the right, and then drag and arrange the tables more closely together so they can all be seen at the same time.

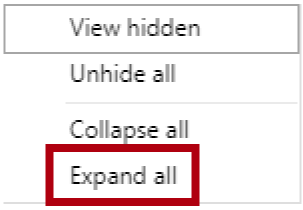
   Tip: You can also use the zoom control located at the bottom of the window.

   In Model view, it's possible to view each table and relationships (connectors between tables). Presently, there are no relationships because in the ***Prepare Data in Power BI Desktop*** *lab, you disabled the data load relationship options.*

3. □ To return to Report view, at the left, click the **Report** view icon.

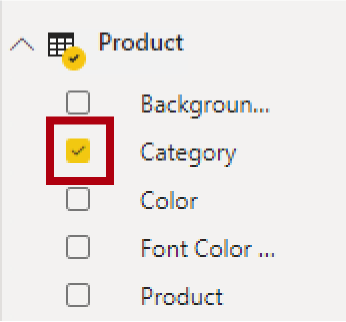


4. □ To view all table fields, in the **Fields** pane, right-click an empty area, and then select **Expand All**.



View hidden
Unhide all
Collapse all
Expand all

5. □ To create a table visual, in the **Fields** pane, from inside the **Product** table, check the **Category** field.

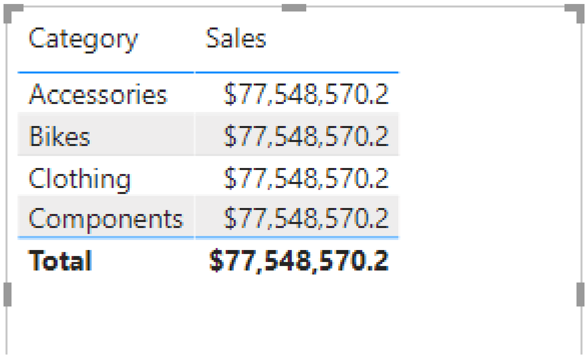


Product
□ Backgroun...
☑ Category
□ Color
□ Font Color ...
□ Product

The labs use a shorthand notation to reference a field. It will look like this: ***Product | Category****. In this example,* ***Product*** *is the table name and* ***Category*** *is the field name.*

6. □ To add an additional column to the table, in the **Fields** pane, check the **Sales | Sales** field.

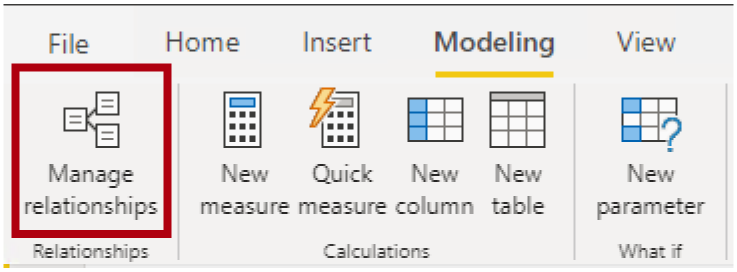
7. □ Notice that the table visual lists four product categories, and that the sales value is the same for each, and the same for the total.



Category	Sales
Accessories	$77,548,570.2
Bikes	$77,548,570.2
Clothing	$77,548,570.2
Components	$77,548,570.2
Total	**$77,548,570.2**

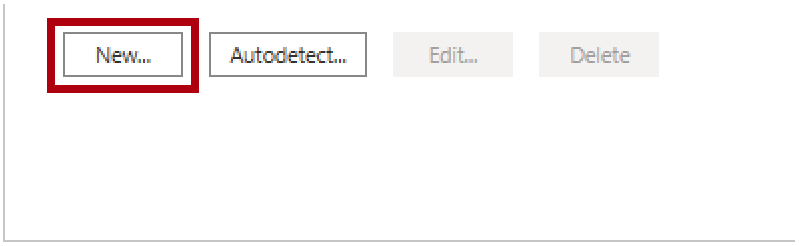
The issue is that the table is based on fields from different tables. The expectation is that each product category displays the sales for that category. However, because there isn't a model relationship between these tables, the ***Sales*** *table is not filtered. You'll now add a relationship to propagate filters between the tables.*

8. □ On the **Modeling** ribbon tab, from inside the **Relationships** group, click **Manage Relationships**.

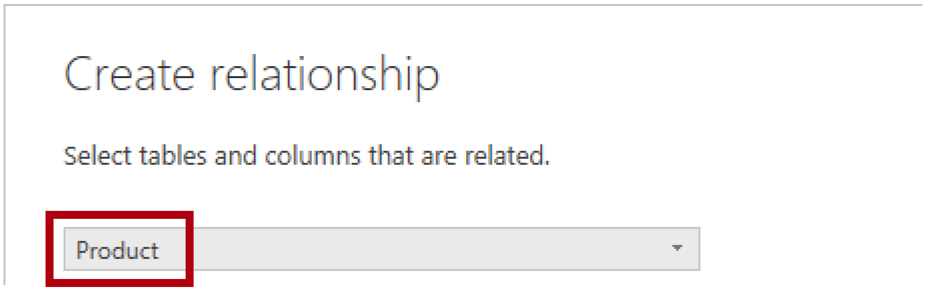


9. □ In the **Manage Relationships** window, notice that no relationships are yet defined.

10. □ To create a relationship, click **New**.



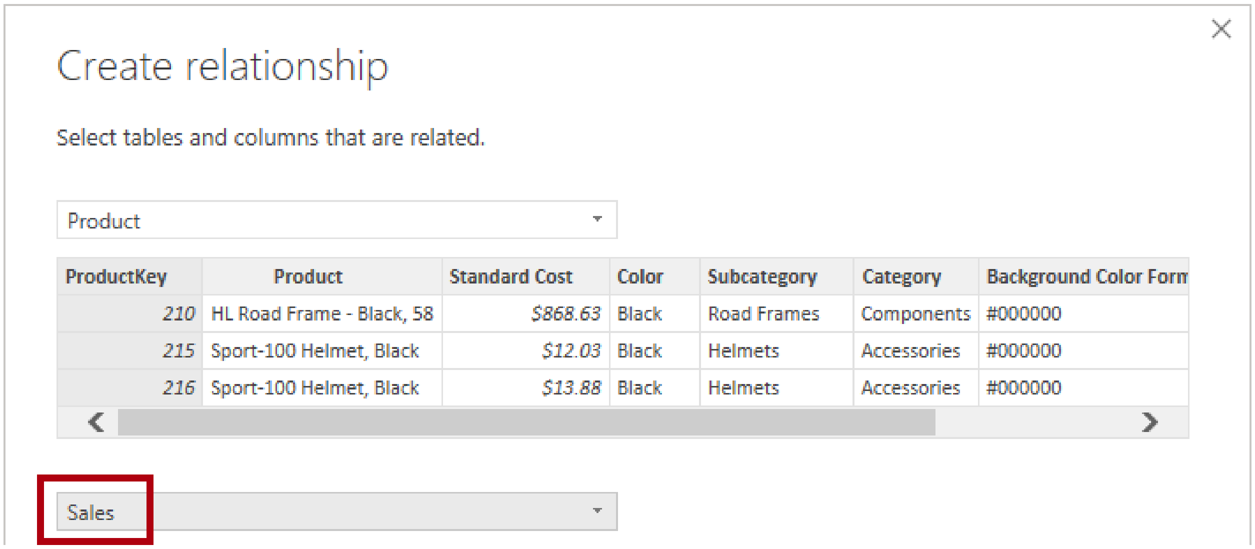
11. □ In the **Create Relationship** window, in the first dropdown list, select the **Product** table.



Create relationship

Select tables and columns that are related.

12. □ In the second dropdown list (beneath the **Product** table grid), select the **Sales** table.



Create relationship

Select tables and columns that are related.

ProductKey	Product	Standard Cost	Color	Subcategory	Category	Background Color Form
210	HL Road Frame - Black, 58	*$868.63*	Black	Road Frames	Components	#000000
215	Sport-100 Helmet, Black	*$12.03*	Black	Helmets	Accessories	#000000
216	Sport-100 Helmet, Black	*$13.88*	Black	Helmets	Accessories	#000000

13. □ Notice the **ProductKey** columns in each table have been automatically selected.

The columns were selected because they share the same name and data type.

14. □ In the **Cardinality** dropdown list, notice that **One To Many (1:*)** is selected.

*The cardinality was automatically detected, because Power BI understands that the **ProductKey** column from the **Product** table contains unique values. One-to-many relationships are the most common cardinality, and all relationship you create in this lab will be this type. You'll work with a Many-to-many cardinality in the **Model Data in Power BI Desktop, Part 2** lab.*

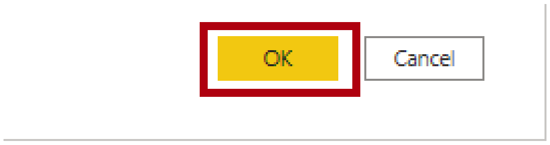
15. □ In the **Cross Filter Direction** dropdown list, notice that **Single** is selected.

*Single filter direction means that filters propagate from the "one side" to the "many side". In this case, it means filters applied to the **Product** table will propagate to the **Sales** table, but not in the opposite direction. You'll work with a bi-directional relationship in the **Model Data in Power BI Desktop, Part 2** lab.*

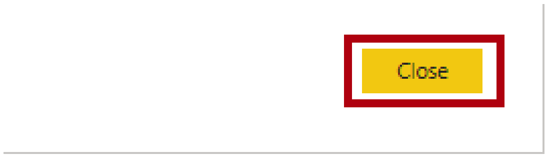
16. □ Notice that the **Mark This Relationship Active** is checked.

Active relationships propagate filters. It's possible to mark a relationship as inactive so filters don't propagate. Inactive relationships can exist when there are multiple relationship paths between tables. In this case, model calculations can use special functions to activate them. You'll work with an inactive relationship in the ***Model Data in Power BI Desktop, Part 2*** *lab.*

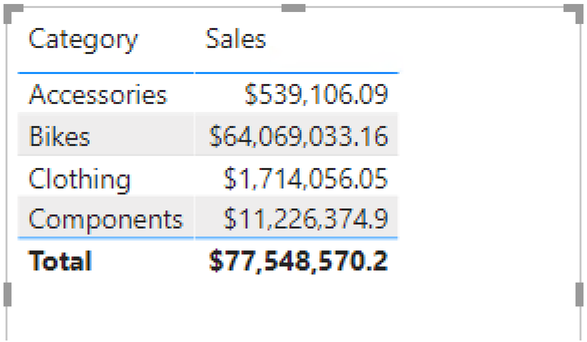
17. □ Click **OK**.



18. □ In the **Manage Relationships** window, notice that the new relationship is listed, and then click **Close**.



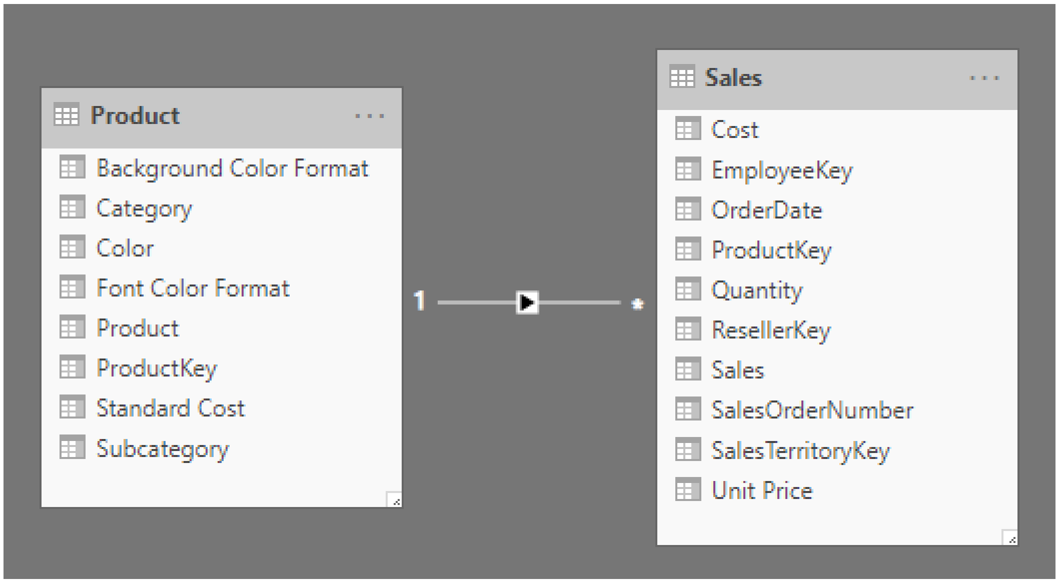
19. □ In the report, notice that the table visual updated to display different values for each product category.



Category	Sales
Accessories	$539,106.09
Bikes	$64,069,033.16
Clothing	$1,714,056.05
Components	$11,226,374.9
Total	**$77,548,570.2**

Filters applied to the ***Product*** *table now propagate to the* ***Sales*** *table.*

20. □ Switch to Model view, and then notice there is now a connector between the two tables (it does not matter if the tables are positioned next to each other).



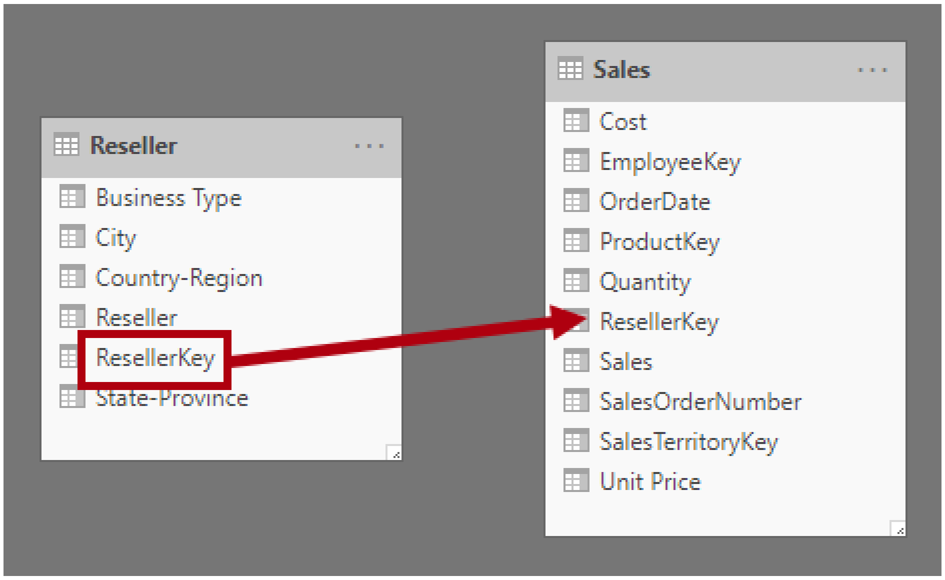
21. □ In the diagram, notice that you can interpret the cardinality which is represented by the **1** and ***** indicators.

Filter direction is represented by the arrow head. A solid line represents an active relationship; a dashed line represents an inactive relationship.

22. □ Hover the cursor over the relationship to highlight the related columns.

There's an easier way to create a relationship. In the model diagram, you can drag and drop columns to create a new relationship.

23. □ To create a new relationship using a different technique, from the **Reseller** table, drag the **ResellerKey** column on to the **ResellerKey** column of the **Sales** table.

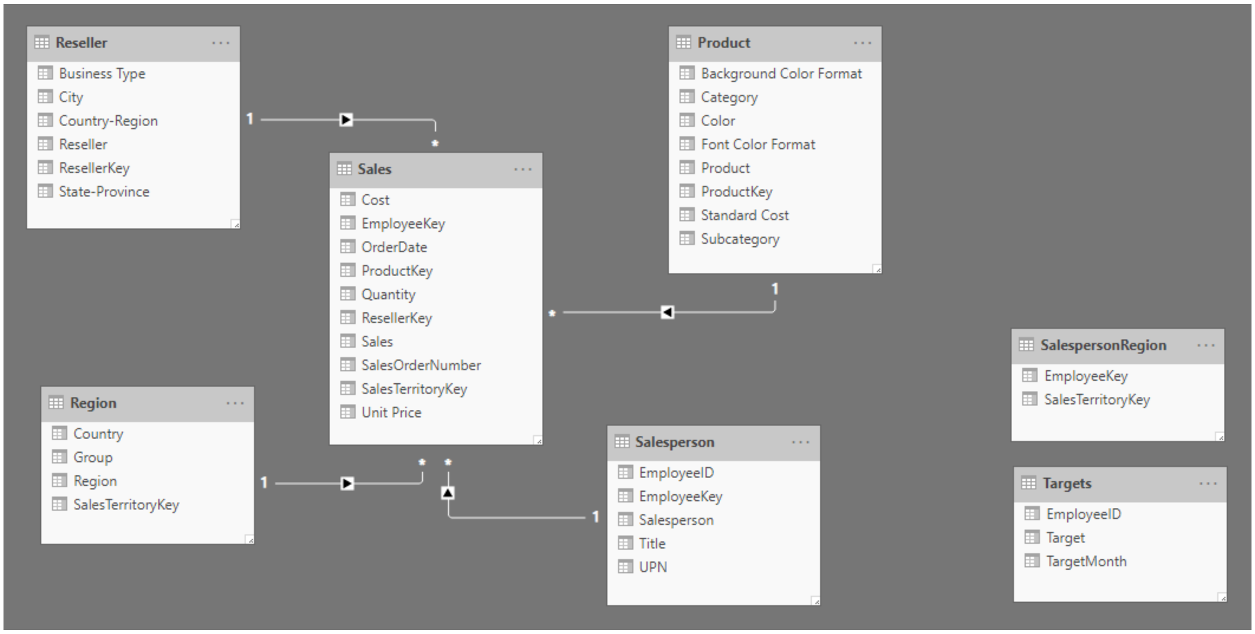


Tip: Sometimes a column doesn't want to be dragged. If this situation arises, select a different column, and then select the column you intend to drag again, and then try again. Ensure that you see the new relationship added to the diagram.

24. □ Use the new technique to create the following two model relationships:
    - **Region | SalesTerritoryKey** to **Sales | SalesTerritoryKey**
    - **Salesperson | EmployeeKey** to **Sales | EmployeeKey**

In this lab the ***SalespersonRegion*** *and* ***Targets*** *tables will remain disconnected. There's a many-to-many relationship between salespeople and regions, and you'll work with this advanced scenario in the* ***Model Data in Power BI Desktop, Part 2*** *lab.*

25. □ In the diagram, arrange the tables so that the **Sales** table is positioned in the center of the diagram, and the related tables are arranged about it. Position the disconnected tables to the side.



26. □ Save the Power BI Desktop file.

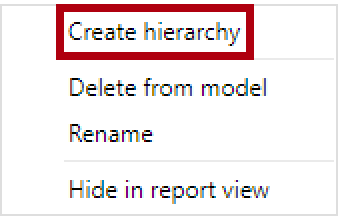
Exercise 2: Configure Tables

In this exercise you will configure each table by creating hierarchies, and hiding, formatting, and categorizing columns.

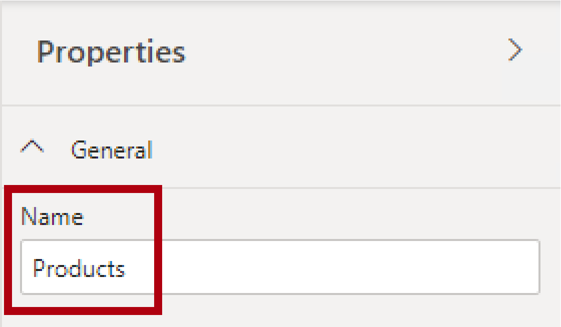
Task 1: Configure the Product table

In this task you will configure the **Product** table.

1. □ In Model view, in the **Fields** pane, if necessary, expand the **Product** table to reveal all fields.
2. □ To create a hierarchy, in the **Fields** pane, right-click the **Category** column, and then select **Create Hierarchy**.



3. □ In the **Properties** pane (to the left of the **Fields** pane), in the **Name** box, replace the text with **Products**.



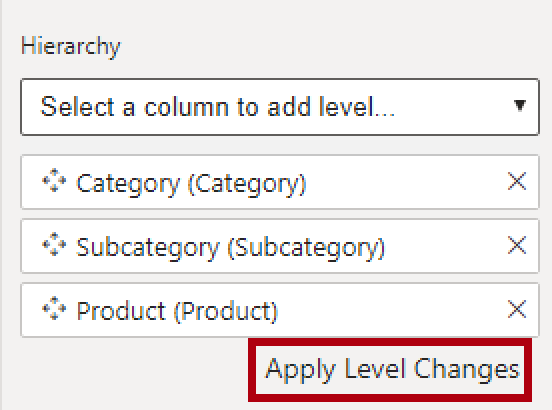
Properties

General

Name

Products

4. □ To add the second level to the hierarchy, in the **Properties** pane, in the **Hierarchy** dropdown list, select **Subcategory** (you might need to scroll down inside the pane).
5. □ To add the third level to the hierarchy, in the **Hierarchy** dropdown list, select **Product**.
6. □ To complete the hierarchy design, click **Apply Level Changes**.



Hierarchy

Select a column to add level...

Category (Category)

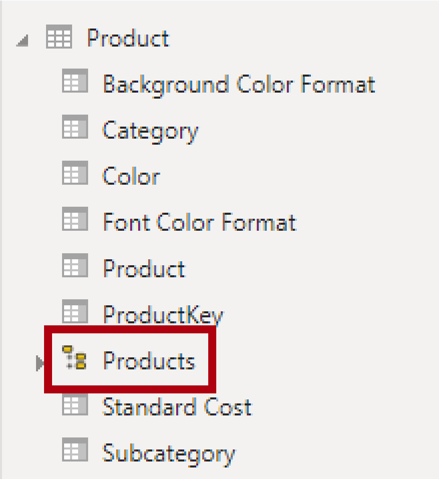
Subcategory (Subcategory)

Product (Product)

Apply Level Changes

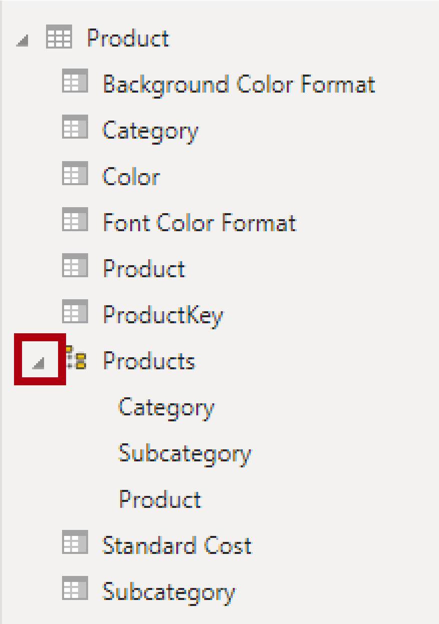
*Tip: Don't forget to click **Apply Level Changes**—it's a common mistake to overlook this step.*

7. □ In the **Fields** pane, notice the **Products** hierarchy.



- Product
  - Background Color Format
  - Category
  - Color
  - Font Color Format
  - Product
  - ProductKey
  - Products
  - Standard Cost
  - Subcategory

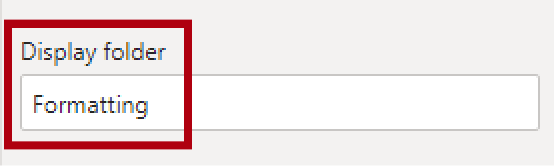
8. □ To reveal the hierarchy levels, expand the **Products** hierarchy.



9. □ To organize columns into a display folder, in the **Fields** pane, first select the **Background Color Format** column.

10. □ While pressing the **Ctrl** key, select the **Font Color Format** column.

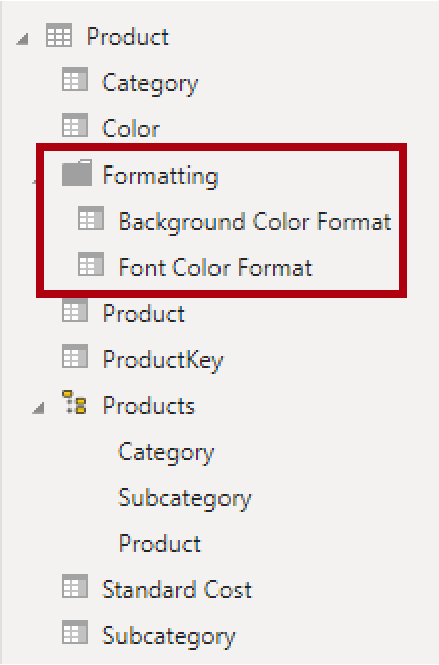
11. □ In the **Properties** pane, in the **Display Folder** box, enter **Formatting** and press **Enter**.



Display folder

Formatting

12. □ In the **Fields** pane, notice that the two columns are now inside a folder.



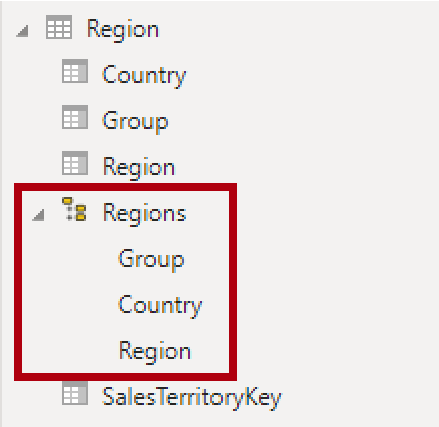
- Product
  - Category
  - Color
  - Formatting
    - Background Color Format
    - Font Color Format
  - Product
  - ProductKey
  - Products
    - Category
    - Subcategory
    - Product
  - Standard Cost
  - Subcategory

Display folders are a great way to declutter tables—especially for tables that comprise many fields.

Task 2: Configure the Region table

In this task you will configure the **Region** table.

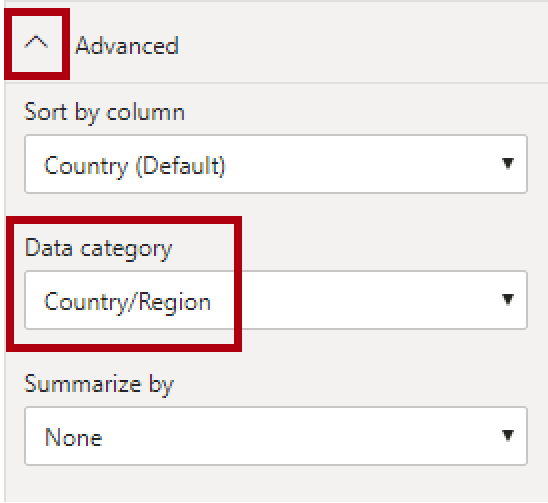
1. ☐ In the **Region** table, create a hierarchy named **Regions**, with the following three levels:
   - Group
   - Country
   - Region



- Region
  - Country
  - Group
  - Region
  - Regions
    - Group
    - Country
    - Region
  - SalesTerritoryKey

2. ☐ Select the **Country** column (not the **Country** hierarchy level).

3. □ In the **Properties** pane, expand the **Advanced** section (at the bottom of the pane), and then in the **Data Category** dropdown list, select **Country/Region**.



Advanced

Sort by column

Country (Default)

Data category

Country/Region

Summarize by

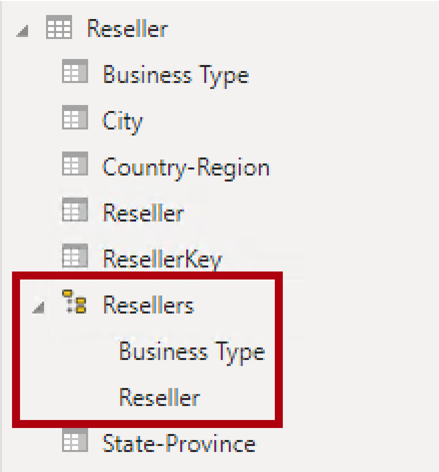
None

Data categorization can provide hints to the report designer. In this case, categorizing the column as country or region provides more accurate information to Power BI when it renders a map visualization.

Task 3: Configure the Reseller table

In this task you will configure the **Reseller** table.

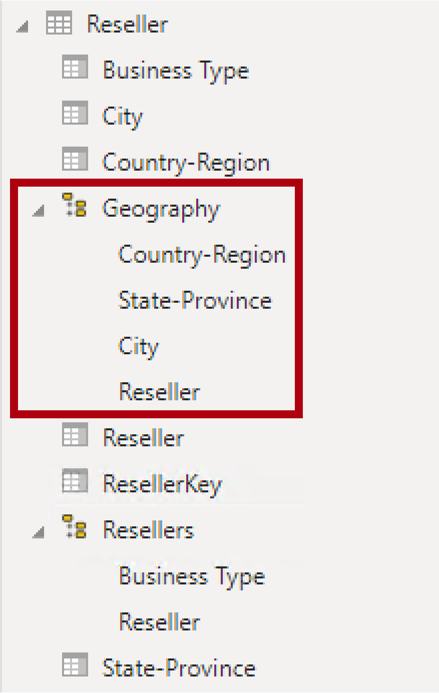
1. □ In the **Reseller** table, create a hierarchy named **Resellers**, with the following two levels:
   - Business Type
   - Reseller



- Reseller
  - Business Type
  - City
  - Country-Region
  - Reseller
  - ResellerKey
  - Resellers
    - Business Type
    - Reseller
  - State-Province

2. □ Create a second hierarchy named **Geography**, with the following four levels:

    - Country-Region
    - State-Province
    - City
    - Reseller

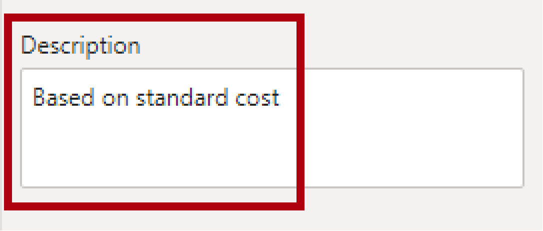


3. □ Categorize the following three columns:
    - **Country-Region** as **Country/Region**
    - **State-Province** as **State or Province**
    - **City** as **City**

Task 4: Configure the Sales table

In this task you will configure the **Sales** table.

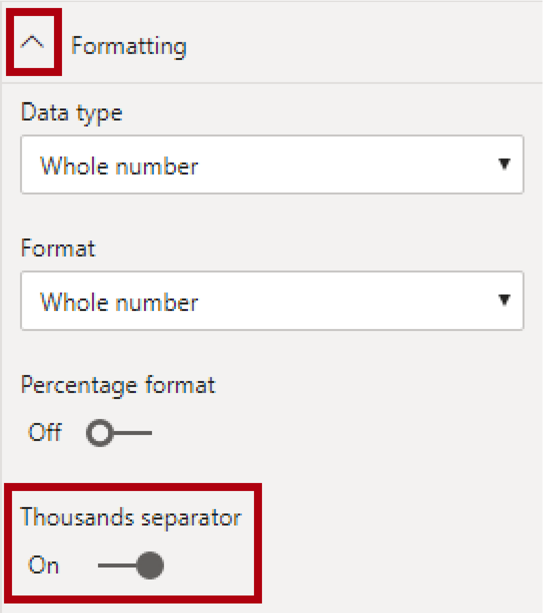
1. □ In the **Sales** table, select the **Cost** column.
2. □ In the **Properties** pane, in the **Description** box, enter: **Based on standard cost**


Description

Based on standard cost

*Descriptions can be applied to tables, columns, hierarchies, or measures. In the **Fields** pane, description text is revealed in a tooltip when a report author hovers their cursor over the field.*

3. □ Select the **Quantity** column.
4. □ In the **Properties** pane, from inside the **Formatting** section, slide the **Thousands Separator** property to **On**.



Formatting

Data type

Whole number

Format

Whole number

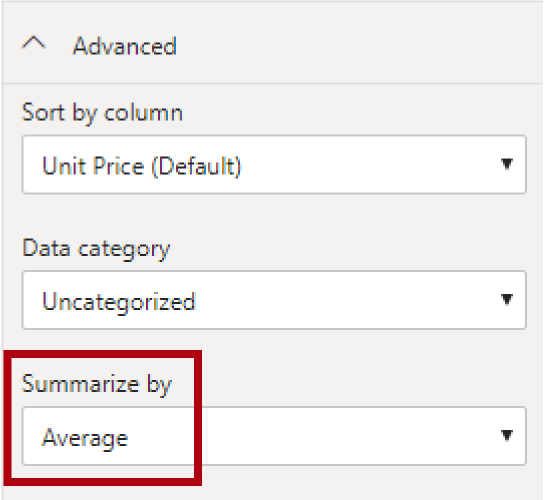
Percentage format

Off

Thousands separator

On

5. □ Select the **Unit Price** column.
6. □ In the **Properties** pane, from inside the **Formatting** section, slide the **Decimal Places** property to **2**.
7. □ In the **Advanced** group (you may need to scroll down to locate it), in the **Summarize By** dropdown list, select **Average**.



Advanced

Sort by column

Unit Price (Default)

Data category

Uncategorized

Summarize by

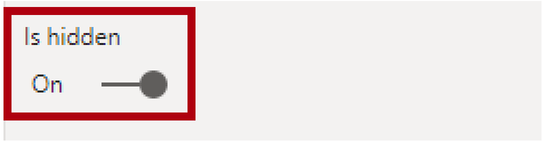
Average

*By default, numeric columns will summarize by summing values together. This default behavior is not suitable for a column like **Unit Price**, which represents a rate. Setting the default summarization to average will produce a meaningful result.*

Task 5: Bulk update properties

In this task you will update multiple columns using single bulk updates. You will use this approach to hide columns, and format column values.

1. □ In the **Fields** pane, select the **Product | ProductKey** column.
2. □ While pressing the **Ctrl** key, select the following 13 columns (spanning multiple tables):
   - Region | SalesTerritoryKey
   - Reseller | ResellerKey
   - Sales | EmployeeKey
   - Sales | ResellerKey
   - Sales | SalesOrderNumber
   - Sales | SalesTerritoryKey
   - Salesperson | EmployeeID
   - Salesperson | EmployeeKey
   - Salesperson | UPN
   - SalespersonRegion | EmployeeKey
   - SalespersonRegion | SalesTerritoryKey
   - Targets | EmployeeID
3. □ In the **Properties** pane, slide the **Is Hidden** property to **On**.



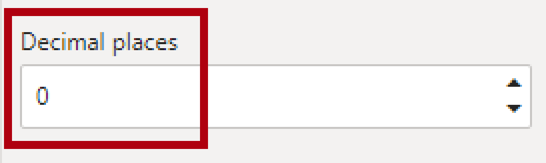
Is hidden

On

The columns were hidden because they're either used by relationships or will be used in row-level security configuration or calculation logic.

*You'll define row-level security in the **Model Data in Power BI Desktop, Part 2** lab using the **UPN** column. You'll use the **SalesOrderNumber** in a calculation in the **Create DAX Calculations in Power BI Desktop, Part 1** lab.*

4. □ Multi-select the following three columns:
   - Product | Standard Cost
   - Sales | Cost
   - Sales | Sales
5. □ In the **Properties** pane, from inside the **Formatting** section, set the **Decimal Places** property to **0** (zero).


Decimal places

0

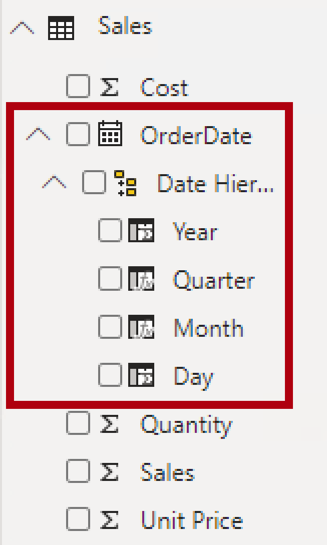
Exercise 3: Review the Model Interface

In this exercise you will switch to Report view, and review the model interface.

Task 1: Review the model interface

In this task you will switch to Report view, and review the model interface.

1. □ Switch to Report view.
2. □ In the **Fields** pane, notice the following:
   - Columns, hierarchies and their levels are fields, which can be used to configure report visuals
   - Only fields relevant to report authoring are visible
   - The **SalespersonRegion** table is not visible—because all of its fields are hidden
   - Spatial fields in the **Region** and **Reseller** table are adorned with a spatial icon
   - Fields adorned with the sigma symbol (Σ) will summarize, by default
   - A tooltip appears when hovering the cursor over the **Sales | Cost** field
3. □ Expand the **Sales | OrderDate** field, and then notice that it reveals a date hierarchy.


Sales

□ Σ Cost

□ OrderDate

□ Date Hier...

□ Year

□ Quarter

□ Month

□ Day

□ Σ Quantity

□ Σ Sales

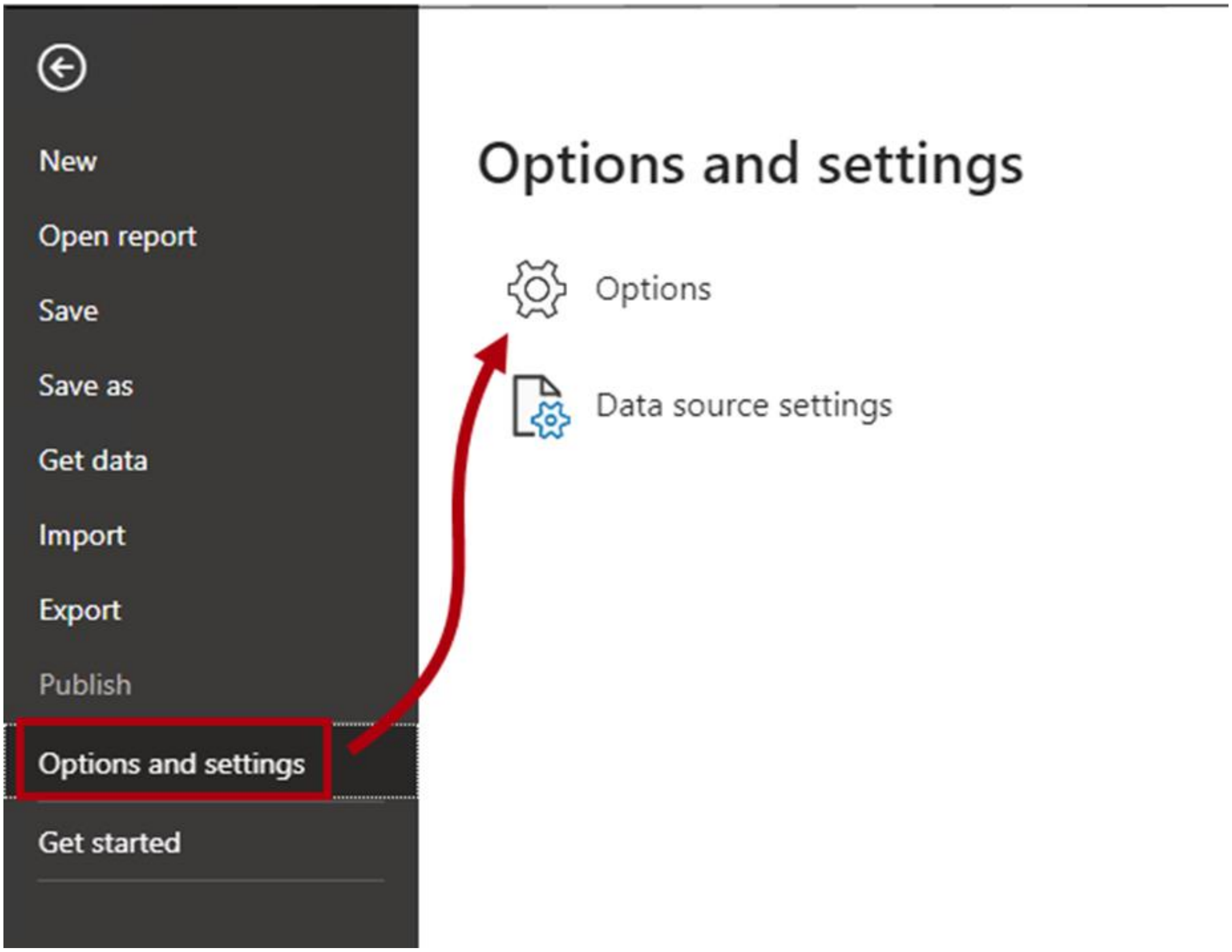
□ Σ Unit Price

*The **Targets | TargetMonth** field delivers a similar hierarchy. These hierarchies were not created by you. They were created automatically. There is a problem, however. The Adventure Works financial year commences on July 1 of each year. But, in these automatically created date hierarchies, the date hierarchy year commences on January 1 of each year.*

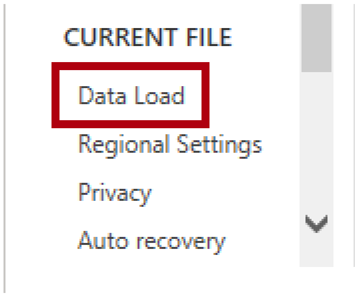
You'll now turn this automatic behavior off. In the ***Create DAX Calculations in Power BI Desktop, Part 1*** *lab, you'll use DAX to create a date table, and configure it define the Adventure Works' calendar.*

4. □ To turn off auto/date time, click the **File** ribbon tab to open the backstage view.

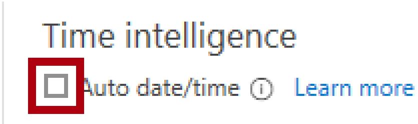
5. □ At the left, select **Options and Settings**, and then select **Options**.



6. □ In the **Options** window, at the left, in the **Current File** group, select **Data Load**.



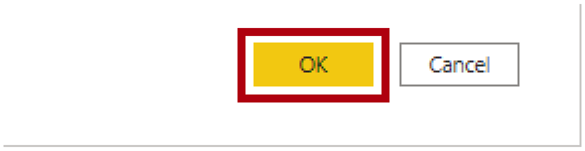
7. □ In the **Time Intelligence** section, uncheck **Auto Date/Time**.



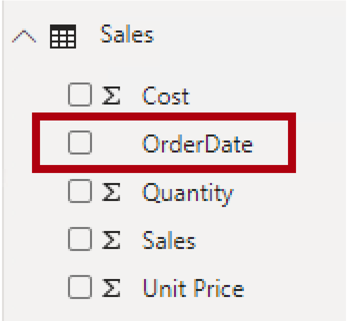
Time intelligence

□ Auto date/time ⓘ Learn more

8. □ Click **OK**.



9. □ In the **Fields** pane, notice that the date hierarchies are no longer available.



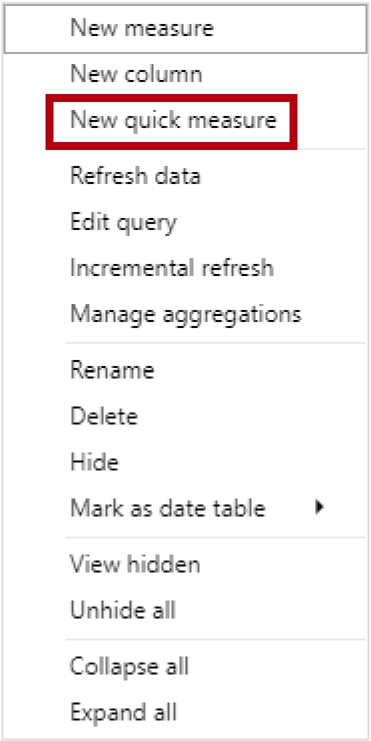
Exercise 4: Create Quick Measures

In this exercise you will create two quick measures.

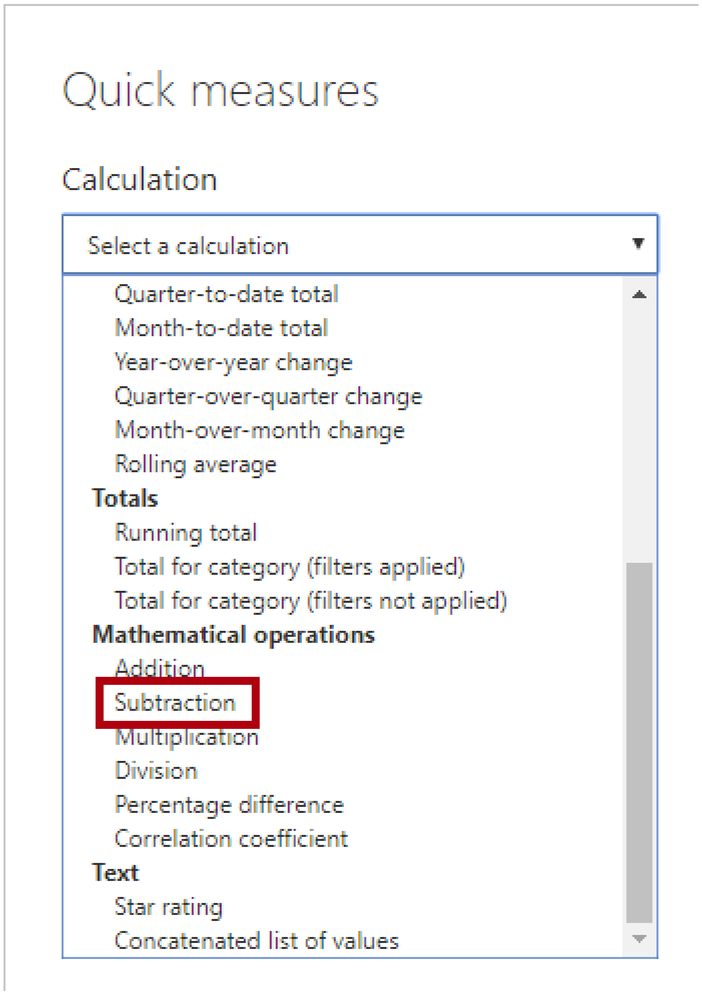
Task 1: Create quick measures

In this task you will create two quick measures to calculate profit and profit margin.

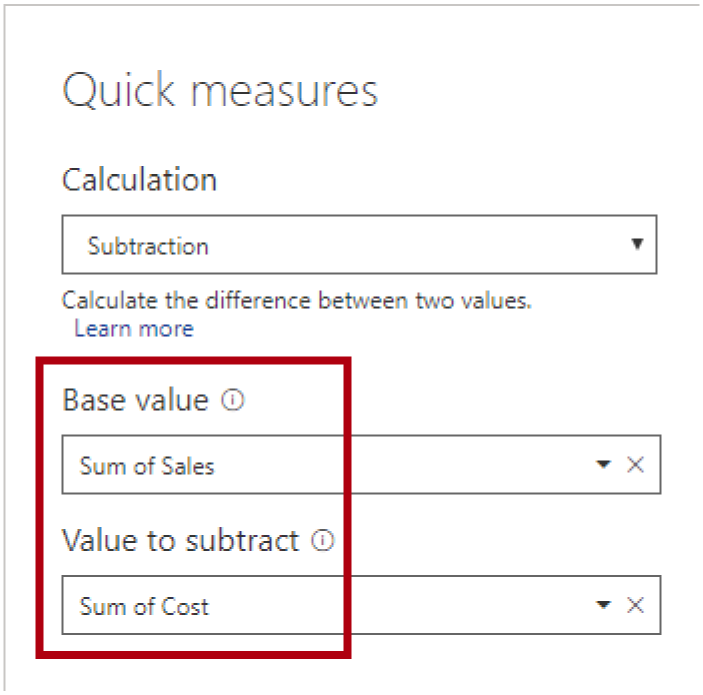
1. □ In the **Fields** pane, right-click the **Sales** table, and then select **New Quick Measure**.



2. □ In the **Quick Measures** window, in the **Calculation** dropdown list, from inside the **Mathematical Operations** group, select **Subtraction**.



3. ☐ In the **Fields** pane of the **Quick Measures** window, expand the **Sales** table.
4. ☐ Drag the **Sales** field into the **Base Value** box.
5. ☐ Drag the **Cost** field into the **Value to Subtract** box.



Quick measures

Calculation

Subtraction

Calculate the difference between two values.
Learn more

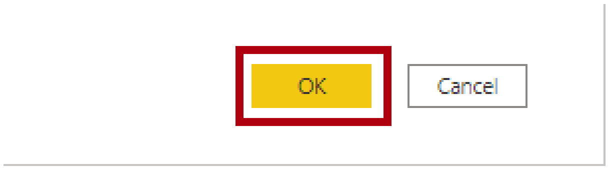
Base value ⓘ

Sum of Sales

Value to subtract ⓘ

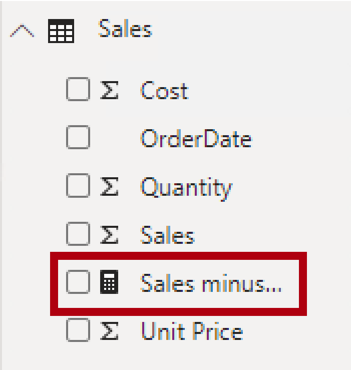
Sum of Cost

6. □ Click **OK**.



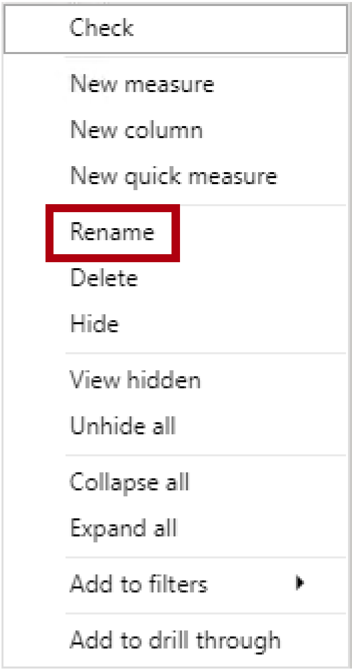
A quick measure creates the calculation formula for you. They're easy and fast to create for simple and common calculations. You'll create measures without using this tool in the ***Create DAX Calculations in Power BI Desktop, Part 1*** *lab.*

7. □ In the **Fields** pane, inside the **Sales** table, notice that new measure.



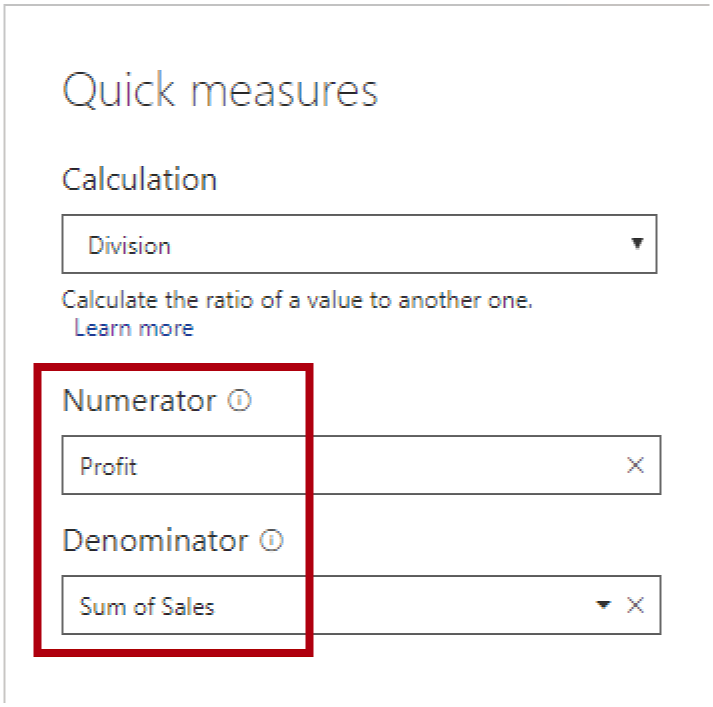
Measures are adorned with the calculator icon.

8. □ To rename the measure, right-click it, and then select **Rename**.



Tip: To rename a field, you can also double-click it, or select it and press ***F2****.*

9. □ Rename the measure to **Profit**, and then press **Enter**.
10. □ In the **Sales** table, add a second quick measure, based on the following requirements:
    - Use the **Division** mathematical operation
    - Set the **Numerator** to the **Sales | Profit** field
    - Set the **Denominator** to **Sales | Sales** field
    - Rename the measure as **Profit Margin**



Quick measures

Calculation

Division

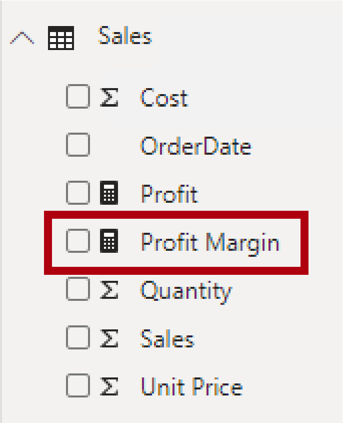
Calculate the ratio of a value to another one.
Learn more

Numerator

Profit

Denominator

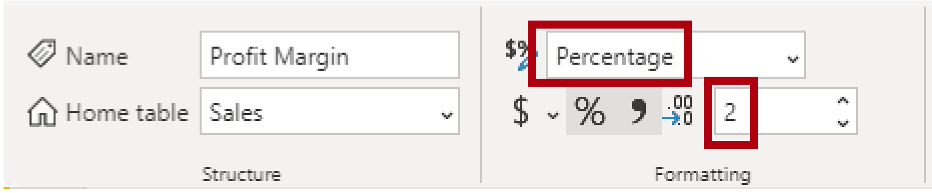
Sum of Sales



Sales

☐ Σ Cost
☐ OrderDate
☐ ▦ Profit
☐ ▦ Profit Margin
☐ Σ Quantity
☐ Σ Sales
☐ Σ Unit Price

11. ☐ Ensure the **Profit Margin** measure is selected, and then on the **Measure Tools** contextual ribbon, set the format to **Percentage**, with two decimal places.

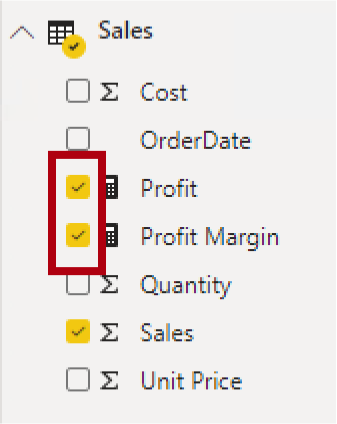


Name: Profit Margin
Home table: Sales

Percentage
2

12. ☐ To test the two measures, first select the table visual on the report page.

13. ☐ In the **Fields** pane, check the two measures.



Sales

☐ Σ Cost

☐ OrderDate

☑ Profit

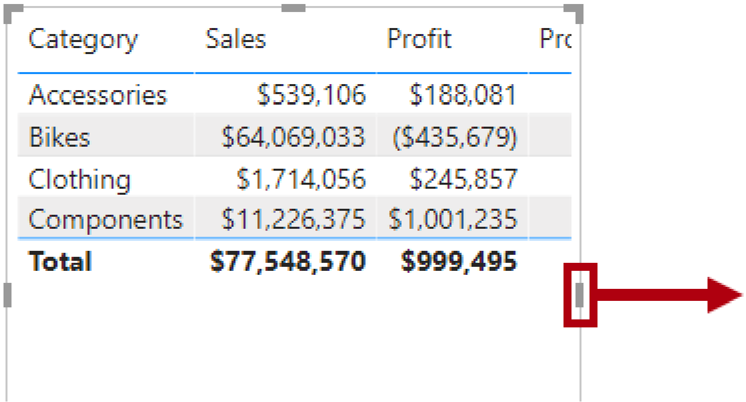
☑ Profit Margin

☐ Σ Quantity

☑ Σ Sales

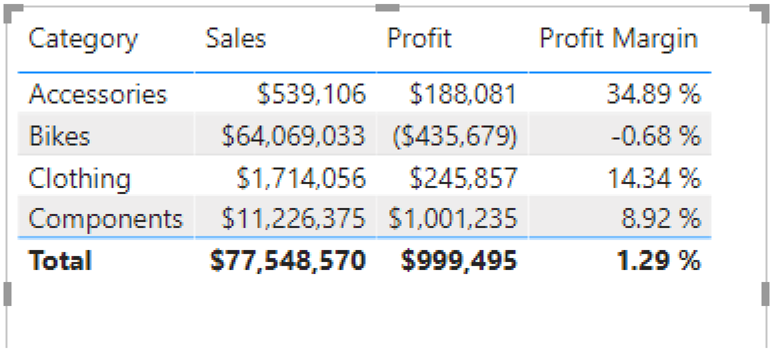
☐ Σ Unit Price

14. □ Click and drag the right guide to widen the table visual.



Category	Sales	Profit	Prc
Accessories	$539,106	$188,081	
Bikes	$64,069,033	($435,679)	
Clothing	$1,714,056	$245,857	
Components	$11,226,375	$1,001,235	
Total	**$77,548,570**	**$999,495**	

15. □ Verify that the measures produce reasonable results that are correctly formatted.



Category	Sales	Profit	Profit Margin
Accessories	$539,106	$188,081	34.89 %
Bikes	$64,069,033	($435,679)	-0.68 %
Clothing	$1,714,056	$245,857	14.34 %
Components	$11,226,375	$1,001,235	8.92 %
Total	**$77,548,570**	**$999,495**	**1.29 %**

Task 2: Finish up

In this task you will complete the lab.

1. □ To remove the table, select the table (by clicking it), and then press the **Delete** key.
2. □ Save the Power BI Desktop file.

3. □ If prompted to apply queries, click **Apply Later**.
4. □ If you intend to start the next lab, leave Power BI Desktop open.

You'll enhance the data model by configuring a many-to-many relationship and row-level security in the ***Model Data in Power BI Desktop, Part 2*** *lab.*

Congratulations

Model Data in Power BI Desktop, Part 2

In this lab you will create a many-to-many relationship between the **Salesperson** table and the **Sales** table.

In this lab you learn how to:

- Configure many-to-many relationships

Exercise 1: Create a Many-to many Relationship

In this exercise you will create a many-to-many relationship between the **Salesperson** table and the **Sales** table.

Task 1: Get started

In this task you will setup the environment for the lab.

Important: If you are continuing on from the previous lab (and you completed that lab successfully), do not complete this task; instead, continue from the next task.

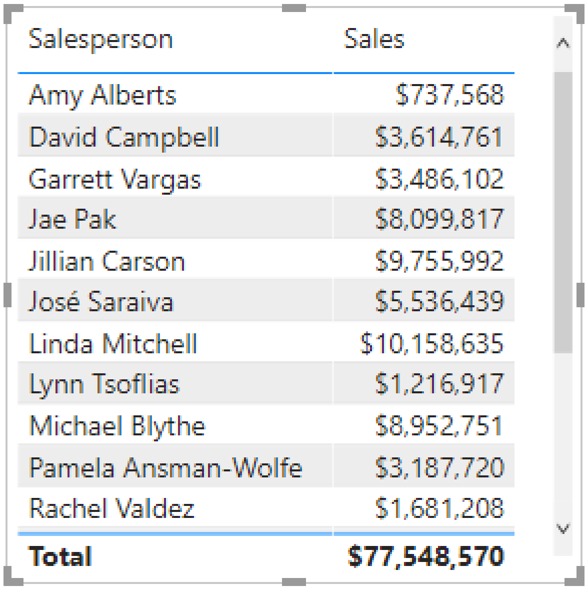
1. □ In the **Open** window, navigate to the **D:\DA100\Labs\04-configure-data-model-in-power-bi-desktop-advanced\Starter** folder.
2. □ Select the **Sales Analysis** file.

Task 2: Create a many-to-many relationship

In this task you will create a many-to-many relationship between the **Salesperson** table and the **Sales** table.

1. □ In Power BI Desktop, in Report view, in the **Fields** pane, check the follow two fields to create a table visual:
    - Salesperson | Salesperson
    - Sales | Sales

    *The labs use a shorthand notation to reference a field. It will look like this: **Salesperson | Salesperson** . In this example, **Salesperson** is the table name and **Salesperson** is the field name.*

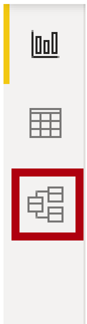


Salesperson	Sales
Amy Alberts	$737,568
David Campbell	$3,614,761
Garrett Vargas	$3,486,102
Jae Pak	$8,099,817
Jillian Carson	$9,755,992
José Saraiva	$5,536,439
Linda Mitchell	$10,158,635
Lynn Tsoflias	$1,216,917
Michael Blythe	$8,952,751
Pamela Ansman-Wolfe	$3,187,720
Rachel Valdez	$1,681,208
Total	**$77,548,570**

The table displays sales made by each salesperson. However, there's another relationship between salespeople and sales. Some salespeople belong to one, two, or possibly more sales regions. In addition, sales regions can have multiple salespeople assigned to them.

From a performance management perspective, a salesperson's sales (based on their assigned territories) need to be analyzed and compared with sales targets. You'll create relationships to support this analysis in the next exercise.

2. □ Notice that Michael Blythe has sold almost $9 million.
3. □ Switch to Model view.

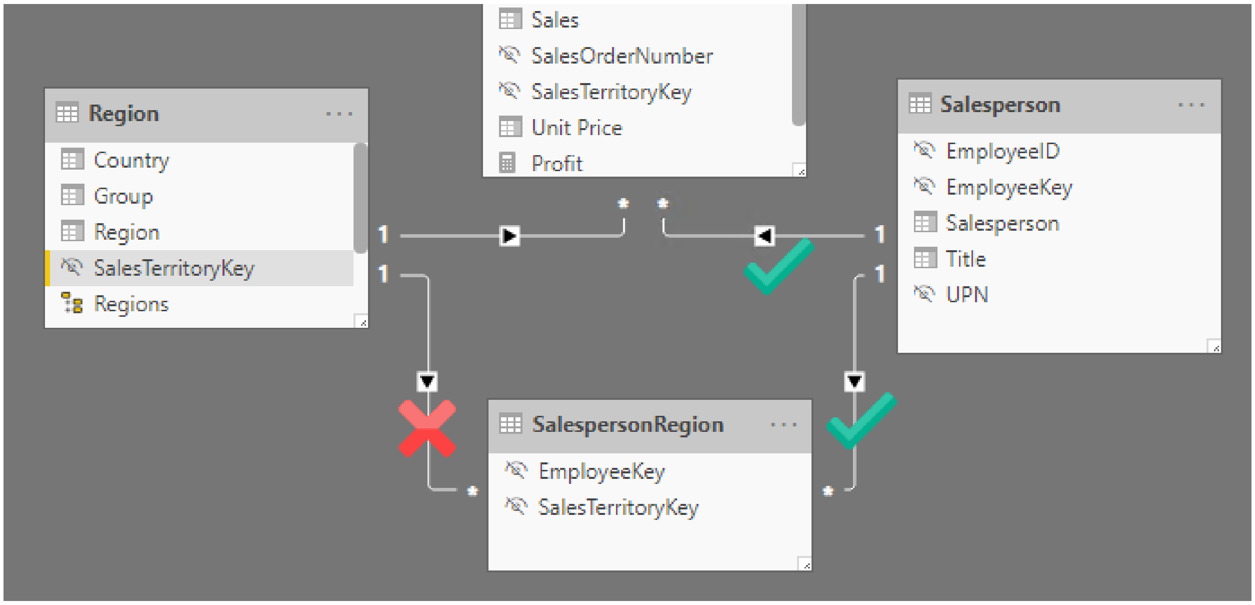


4. □ Drag the **SalespersonRegion** table to position it between the **Region** and **Salesperson** tables.
5. □ Use the drag-and-drop technique to create the following two model relationships:
   - **Salesperson | EmployeeKey** to **SalespersonRegion | EmployeeKey**
   - **Region | SalesTerritoryKey** to **SalespersonRegion | SalesTerritoryKey**

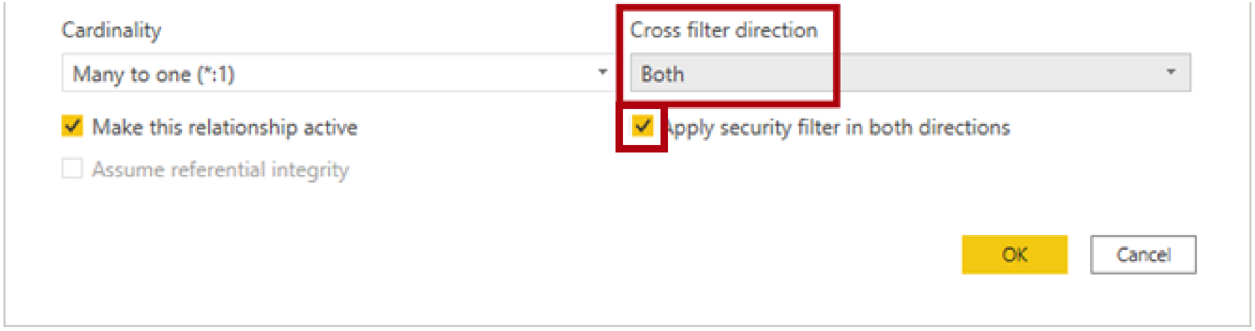
*The **SalespersonRegion** table can be considered to be a bridging table.*

6. □ Switch to Report view, and then notice that the visual has not updated—the sales result for Michael Blythe has not changed.
7. □ Switch back to Model view, and then follow the relationship filter directions (arrowhead) from the **Salesperson** table.

*Consider that the **Salesperson** table filters the **Sales** table. It also filters the **SalespersonRegion** table, but it does not continue by propagating filters to the **Region** table (the arrowhead is pointing the wrong direction).*



8. □ To edit the relationship between the **Region** and **SalespersonRegion** tables, double-click the relationship.
9. □ In the **Edit Relationship** window, in the **Cross Filter Direction** dropdown list, select **Both**.
10. □ Check the **Apply Security Filter in Both Directions** checkbox.



Cardinality: Many to one (*:1)

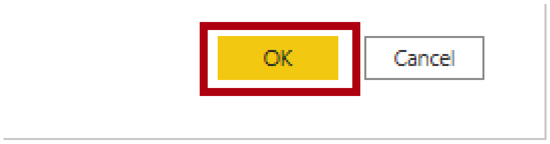
Cross filter direction: Both

☑ Make this relationship active

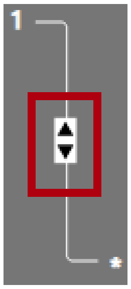
☑ Apply security filter in both directions

□ Assume referential integrity

11. □ Click **OK**.



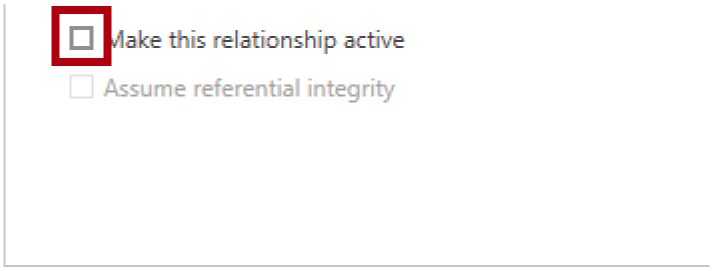
12. □ Notice that the relationship has a double arrowhead.



13. □ Switch to Report view, and then notice that the sales values have still not changed.

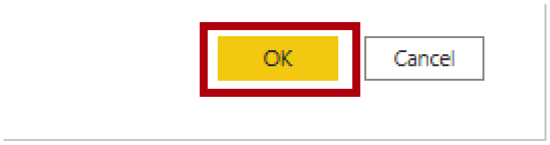
The issue now relates to the fact that there are two possible filter propagation paths between the ***Salesperson*** *and* ***Sales*** *tables. This ambiguity is internally resolved, based on a "least number of tables" assessment. To be clear, you shouldn't design models with this type of ambiguity—the issue will be addressed in part later in this lab, and by the completion of the* ***Create DAX Calculations in Power BI Desktop, Part 1*** *lab.*

14. □ Switch to Model view.
15. □ To force filter propagation via the bridging table, edit (double-click) the relationship between the **Salesperson** and **Sales** tables.
16. □ In the **Edit Relationship** window, uncheck the **Make This Relationship Active** checkbox.



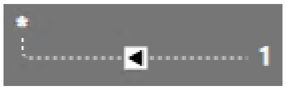
□ Make this relationship active
□ Assume referential integrity

17. □ Click **OK**.

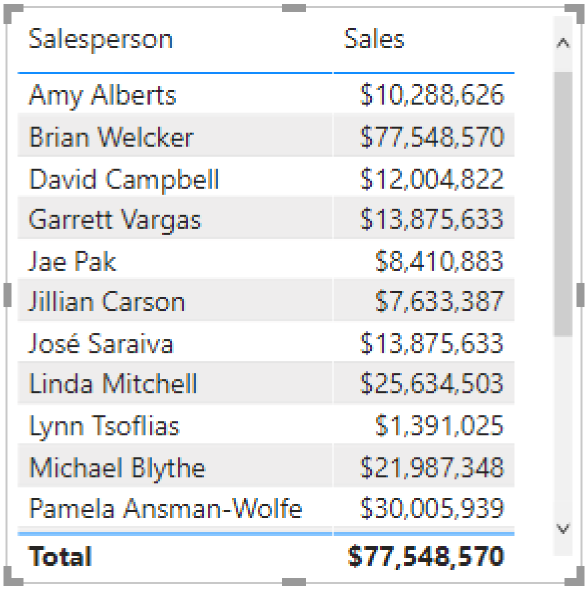


The filter propagation will now follow the only active path.

18. □ In the diagram, notice that the inactive relationship is represented by a dashed line.



19. □ Switch to Report view, and then notice that the sales for Michael Blythe is now nearly $22 million.



Salesperson	Sales
Amy Alberts	$10,288,626
Brian Welcker	$77,548,570
David Campbell	$12,004,822
Garrett Vargas	$13,875,633
Jae Pak	$8,410,883
Jillian Carson	$7,633,387
José Saraiva	$13,875,633
Linda Mitchell	$25,634,503
Lynn Tsoflias	$1,391,025
Michael Blythe	$21,987,348
Pamela Ansman-Wolfe	$30,005,939
Total	**$77,548,570**

20. □ Notice also, that the sales for each salesperson—if added—would exceed the table total.

It's a common observation of a many-to-many relationship due to the double, triple, etc. counting of regional sales results. Consider Brian Welcker, the second salesperson listed. His sales amount equals the total sales amount. It's the correct result simply due to the fact the he's the Director of Sales; his sales are measured by the sales of all regions.

While the many-to-many relationship is now working, it's now not possible to analyze sales made by a salesperson (because the relationship is inactive). You'll be able to reactive the relationship when you introduce a calculated table that to represent salesperson for performance analysis (of their regions) in the ***Create DAX Calculations in Power BI Desktop, Part 1*** *lab.*

21. □ Switch to Modeling view, and then in the diagram, select the **Salesperson** table.

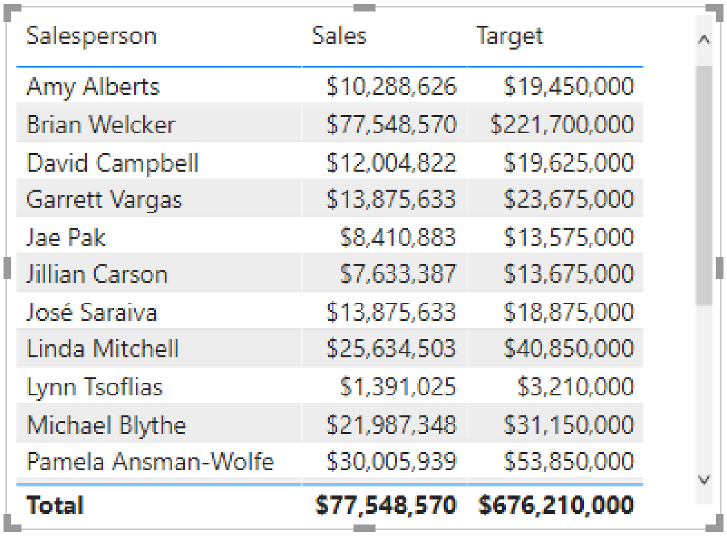
22. □ In the **Properties** pane, in the **Name** box, replace the text with **Salesperson (Performance)**.

The renamed table now reflects its purpose: it's used to report and analyze the performance of salespeople based on the sales of their assigned sales regions.

Task 3: Relate the Targets table

In this task you will create a relationship to the **Targets** table

1. □ Create a relationship from the **Salesperson (Performance) | EmployeeID** column and the **Targets | EmployeeID** column.
2. □ In Report view, add the **Targets | Target** field to the table visual.
3. □ Resize the table visual so all columns are visible.



Salesperson	Sales	Target
Amy Alberts	$10,288,626	$19,450,000
Brian Welcker	$77,548,570	$221,700,000
David Campbell	$12,004,822	$19,625,000
Garrett Vargas	$13,875,633	$23,675,000
Jae Pak	$8,410,883	$13,575,000
Jillian Carson	$7,633,387	$13,675,000
José Saraiva	$13,875,633	$18,875,000
Linda Mitchell	$25,634,503	$40,850,000
Lynn Tsoflias	$1,391,025	$3,210,000
Michael Blythe	$21,987,348	$31,150,000
Pamela Ansman-Wolfe	$30,005,939	$53,850,000
Total	**$77,548,570**	**$676,210,000**

It's now possible to visualize sales and targets—but take care for two reasons. First, there's no filter on a time period, and so targets also include future target amounts. Second, targets are not additive, and so the total should not be displayed. They can either disabled by formatting the visual or removed by using calculation logic. You'll follow the second approach by creationg a target measure in the ***Create DAX Calculations in Power BI Desktop, Part 2*** *lab that'll return BLANK when more than one salesperson is filtered.*

Task 4: Finish up

In this task you will complete the lab.

1. □ Save the Power BI Desktop file.
2. □ If prompted to apply queries, click **Apply Later**.
3. □ If you intend to start the next lab, leave Power BI Desktop open.

You'll enhance the data model with calculations using DAX in the ***Create DAX Calculations in Power BI Desktop, Part 2*** *lab.*

Congratulations!

Model Data in Power BI Desktop, Part 2

In this lab you will create a many-to-many relationship between the **Salesperson** table and the **Sales** table.

In this lab you learn how to:

- Configure many-to-many relationships

Exercise 1: Create a Many-to many Relationship

In this exercise you will create a many-to-many relationship between the **Salesperson** table and the **Sales** table.

Task 1: Get started

In this task you will setup the environment for the lab.

Important: If you are continuing on from the previous lab (and you completed that lab successfully), do not complete this task; instead, continue from the next task.

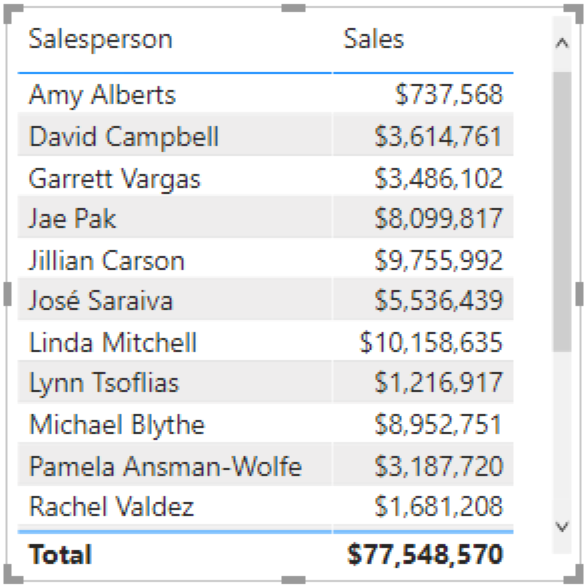
1. □ In the **Open** window, navigate to the **D:\DA100\Labs\04-configure-data-model-in-power-bi-desktop-advanced\Starter** folder.
2. □ Select the **Sales Analysis** file.

Task 2: Create a many-to-many relationship

In this task you will create a many-to-many relationship between the **Salesperson** table and the **Sales** table.

1. □ In Power BI Desktop, in Report view, in the **Fields** pane, check the follow two fields to create a table visual:
   - Salesperson | Salesperson
   - Sales | Sales

   *The labs use a shorthand notation to reference a field. It will look like this: **Salesperson | Salesperson** . In this example, **Salesperson** is the table name and **Salesperson** is the field name.*

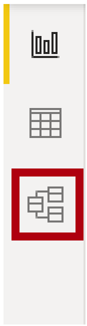


Salesperson	Sales
Amy Alberts	$737,568
David Campbell	$3,614,761
Garrett Vargas	$3,486,102
Jae Pak	$8,099,817
Jillian Carson	$9,755,992
José Saraiva	$5,536,439
Linda Mitchell	$10,158,635
Lynn Tsoflias	$1,216,917
Michael Blythe	$8,952,751
Pamela Ansman-Wolfe	$3,187,720
Rachel Valdez	$1,681,208
Total	**$77,548,570**

The table displays sales made by each salesperson. However, there's another relationship between salespeople and sales. Some salespeople belong to one, two, or possibly more sales regions. In addition, sales regions can have multiple salespeople assigned to them.

From a performance management perspective, a salesperson's sales (based on their assigned territories) need to be analyzed and compared with sales targets. You'll create relationships to support this analysis in the next exercise.

2. □ Notice that Michael Blythe has sold almost $9 million.
3. □ Switch to Model view.



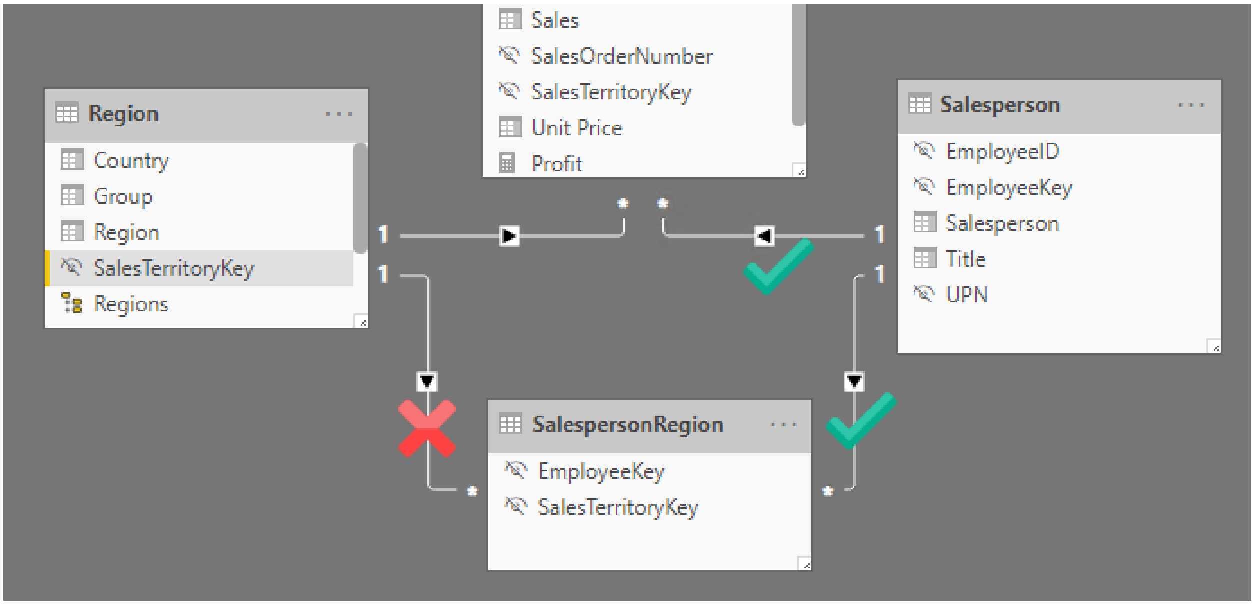
4. □ Drag the **SalespersonRegion** table to position it between the **Region** and **Salesperson** tables.
5. □ Use the drag-and-drop technique to create the following two model relationships:
   - **Salesperson | EmployeeKey** to **SalespersonRegion | EmployeeKey**
   - **Region | SalesTerritoryKey** to **SalespersonRegion | SalesTerritoryKey**

*The **SalespersonRegion** table can be considered to be a bridging table.*

6. □ Switch to Report view, and then notice that the visual has not updated—the sales result for Michael Blythe has not changed.

7. □ Switch back to Model view, and then follow the relationship filter directions (arrowhead) from the **Salesperson** table.

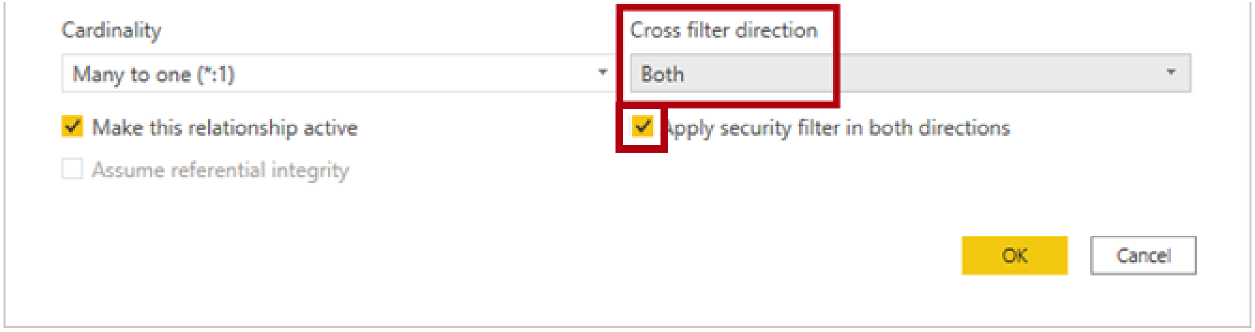
   *Consider that the **Salesperson** table filters the **Sales** table. It also filters the **SalespersonRegion** table, but it does not continue by propagating filters to the **Region** table (the arrowhead is pointing the wrong direction).*



8. □ To edit the relationship between the **Region** and **SalespersonRegion** tables, double-click the relationship.

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

10. □ Check the **Apply Security Filter in Both Directions** checkbox.



Cardinality: Many to one (*:1)

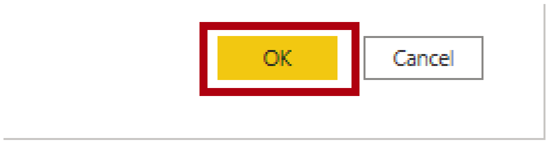
Cross filter direction: Both

☑ Make this relationship active

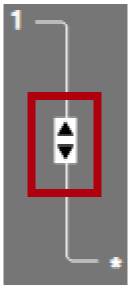
☑ Apply security filter in both directions

☐ Assume referential integrity

11. □ Click **OK**.



12. □ Notice that the relationship has a double arrowhead.



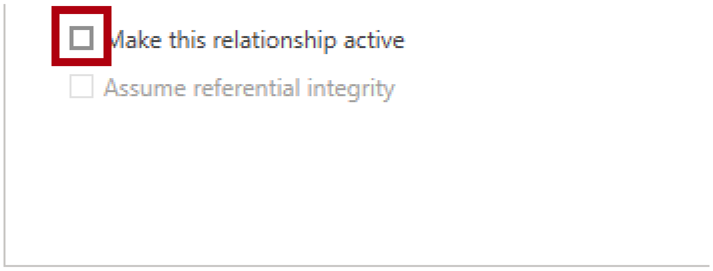
13. □ Switch to Report view, and then notice that the sales values have still not changed.

*The issue now relates to the fact that there are two possible filter propagation paths between the **Salesperson** and **Sales** tables. This ambiguity is internally resolved, based on a "least number of tables" assessment. To be clear, you shouldn't design models with this type of ambiguity—the issue will be addressed in part later in this lab, and by the completion of the **Create DAX Calculations in Power BI Desktop, Part 1** lab.*

14. □ Switch to Model view.

15. □ To force filter propagation via the bridging table, edit (double-click) the relationship between the **Salesperson** and **Sales** tables.

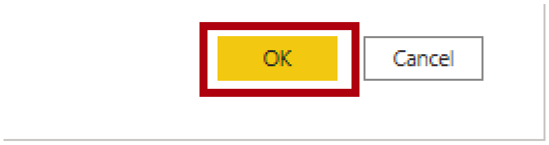
16. □ In the **Edit Relationship** window, uncheck the **Make This Relationship Active** checkbox.



☐ Make this relationship active

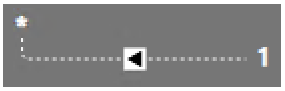
☐ Assume referential integrity

17. □ Click **OK**.

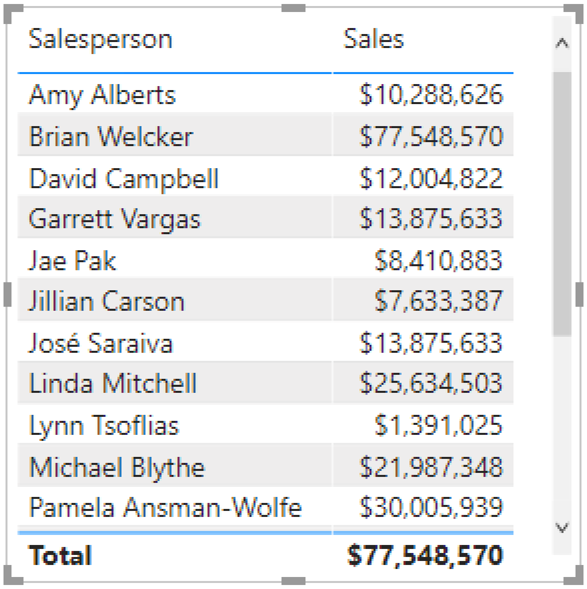


The filter propagation will now follow the only active path.

18. □ In the diagram, notice that the inactive relationship is represented by a dashed line.



19. □ Switch to Report view, and then notice that the sales for Michael Blythe is now nearly $22 million.



Salesperson	Sales
Amy Alberts	$10,288,626
Brian Welcker	$77,548,570
David Campbell	$12,004,822
Garrett Vargas	$13,875,633
Jae Pak	$8,410,883
Jillian Carson	$7,633,387
José Saraiva	$13,875,633
Linda Mitchell	$25,634,503
Lynn Tsoflias	$1,391,025
Michael Blythe	$21,987,348
Pamela Ansman-Wolfe	$30,005,939
Total	**$77,548,570**

20. □ Notice also, that the sales for each salesperson—if added—would exceed the table total.

It's a common observation of a many-to-many relationship due to the double, triple, etc. counting of regional sales results. Consider Brian Welcker, the second salesperson listed. His sales amount equals the total sales amount. It's the correct result simply due to the fact the he's the Director of Sales; his sales are measured by the sales of all regions.

While the many-to-many relationship is now working, it's now not possible to analyze sales made by a salesperson (because the relationship is inactive). You'll be able to reactive the relationship when you introduce a calculated table that to represent salesperson for performance analysis (of their regions) in the ***Create DAX Calculations in Power BI Desktop, Part 1*** *lab.*

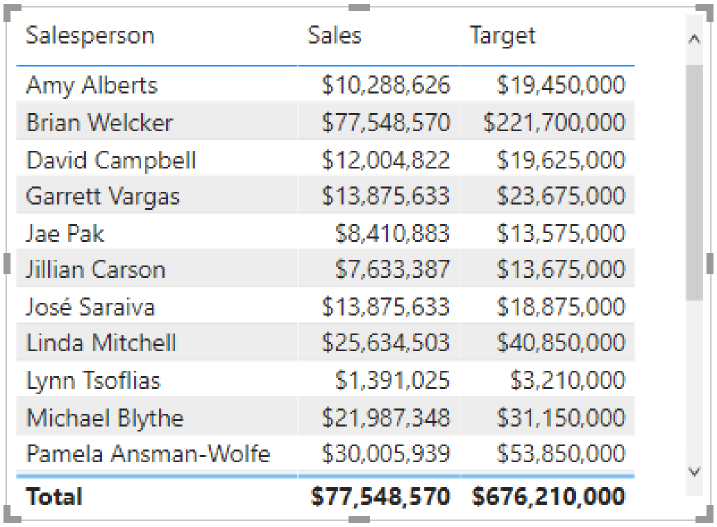
21. □ Switch to Modeling view, and then in the diagram, select the **Salesperson** table.
22. □ In the **Properties** pane, in the **Name** box, replace the text with **Salesperson (Performance)**.

The renamed table now reflects its purpose: it's used to report and analyze the performance of salespeople based on the sales of their assigned sales regions.

Task 3: Relate the Targets table

In this task you will create a relationship to the **Targets** table

1. □ Create a relationship from the **Salesperson (Performance) | EmployeeID** column and the **Targets | EmployeeID** column.
2. □ In Report view, add the **Targets | Target** field to the table visual.
3. □ Resize the table visual so all columns are visible.



Salesperson	Sales	Target
Amy Alberts	$10,288,626	$19,450,000
Brian Welcker	$77,548,570	$221,700,000
David Campbell	$12,004,822	$19,625,000
Garrett Vargas	$13,875,633	$23,675,000
Jae Pak	$8,410,883	$13,575,000
Jillian Carson	$7,633,387	$13,675,000
José Saraiva	$13,875,633	$18,875,000
Linda Mitchell	$25,634,503	$40,850,000
Lynn Tsoflias	$1,391,025	$3,210,000
Michael Blythe	$21,987,348	$31,150,000
Pamela Ansman-Wolfe	$30,005,939	$53,850,000
Total	**$77,548,570**	**$676,210,000**

It's now possible to visualize sales and targets—but take care for two reasons. First, there's no filter on a time period, and so targets also include future target amounts. Second, targets are not additive, and so the total should not be displayed. They can either disabled by formatting the visual or removed by using calculation logic. You'll follow the second approach by creationg a target measure in the ***Create DAX Calculations in Power BI Desktop, Part 2*** *lab that'll return BLANK when more than one salesperson is filtered.*

Task 4: Finish up

In this task you will complete the lab.

1. □ Save the Power BI Desktop file.
2. □ If prompted to apply queries, click **Apply Later**.
3. □ If you intend to start the next lab, leave Power BI Desktop open.

You'll enhance the data model with calculations using DAX in the ***Create DAX Calculations in Power BI Desktop, Part 2*** *lab.*

Congratulations!

Create DAX Calculations in Power BI Desktop, Part 1

In this lab you will create calculated tables, calculated columns, and simple measures using Data Analysis Expressions (DAX).

In this lab you learn how to:

- Create calculated tables
- Create calculated columns
- Create measures

Exercise 1: Create Calculated Tables

In this exercise you will create two calculated tables. The first will be the **Salesperson** table, to allow a direct relationship between it and the **Sales** table. The second will be the **Date** table.

Task 1: Get started

In this task you will setup the environment for the lab.

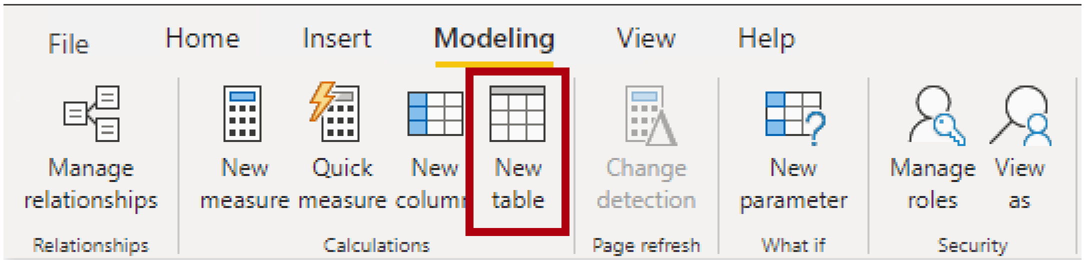
Important: If you are continuing on from the previous lab (and you completed that lab successfully), do not complete this task; instead, continue from the next task.

1. □ In the **Open** window, navigate to the **D:\DA100\Labs\05-create-dax-calculations-in-power-bi-desktop\Starter** folder.
2. □ Select the **Sales Analysis** file.

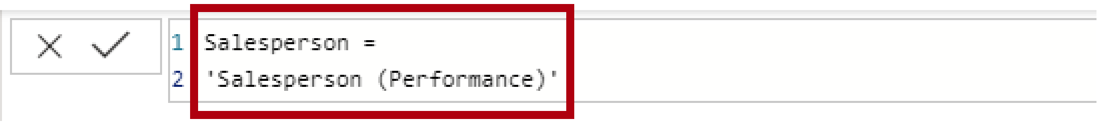
Task 2: Create the Salesperson table

In this task you will create the **Salesperson** table (direct relationship to **Sales**).

1. □ In Power BI Desktop, in Report view, on the **Modeling** ribbon, from inside the **Calculations** group, click **New Table**.



2. □ In the formula bar (which opens directly beneath the ribbon when creating or editing calculations), type **Salesperson =**, press **Shift+Enter**, type **'Salesperson (Performance)'**, and then press **Enter**.



```
Salesperson =
'Salesperson (Performance)'
```

*For your convenience, all DAX definitions in this lab can be copied from the snippets file, located in **D:\DA100\Labs\05-create-dax-calculations-in-power-bi-desktop\Assets\Snippets.txt**.*

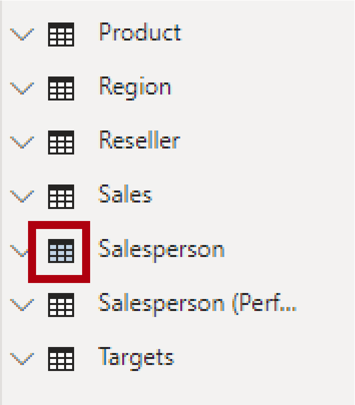
A calculated table is created by first entering the table name, followed by the equals symbol (=), followed by a DAX formula that returns a table. Note that the table name cannot already exist in the data model.

The formula bar supports entering a valid DAX formula. It includes features like auto-complete, Intellisense and color-coding, enabling you to quickly and accurately enter the formula.

*This table definition creates a copy of the **Salesperson (Performance)** table. It copies the data only, however model properties like visibility, formatting, etc. are not copied.*

*Tip: You're encouraged to enter "white space" (i.e. carriage returns and tabs) to layout formulas in an intuitive and easy-to-read format—especially when formulas are long and complex. To enter a carriage return, press **Shift+Enter**. "White space" is optional.*

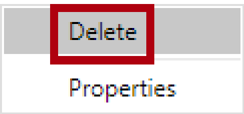
3. □ In the **Fields** pane, notice that the table icon is a shade of blue (denoting a calculated table).



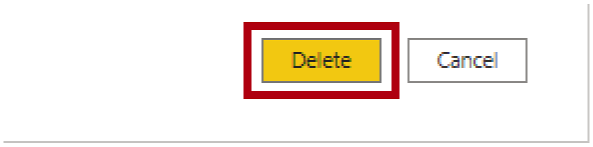
Calculated tables are defined by using a DAX formula which returns a table. It's important to understand that calculated tables increase the size of the data model because they materialize and store values. They're recomputed whenever formula dependencies are refreshed, as will be the case for this data model when new (future) date values are loaded into tables.

Unlike Power Query-sourced tables, calculated tables can't be used to load data from external data sources. They can only transform data based on what has already been loaded into the data model.

4. □ Switch to Model view.
5. □ Notice that the **Salesperson** table is available (take care, it might be hidden from view, in which case scroll horizontally to locate it).
6. □ Create a relationship from the **Salesperson | EmployeeKey** column to the **Sales | EmployeeKey** column.
7. □ Right-click the inactive relationship between the **Salesperson (Performance)** and **Sales** tables, and then select **Delete**.



8. □ When prompted to confirm the deletion, click **Delete**.



9. □ In the **Salesperson** table, multi-select the following columns, and then hide them (set the **Is Hidden** property to **Yes**):
    - EmployeeID
    - EmployeeKey
    - UPN
10. □ In the model diagram, select the **Salesperson** table.
11. □ In the **Properties** pane, in the **Description** box, enter: **Salesperson related to Sales**

*You may recall that descriptions appear as tooltips in the **Fields** pane when the user hovers their cursor over a table or field.*

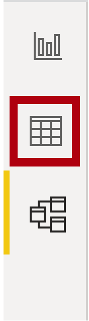
12. □ For the **Salesperson (Performance)** table, set the description to: **Salesperson related to region(s)**

*The data model now provides two alternatives when analyzing salespeople. The **Salesperson** table allows analyzing sales made by a salesperson, while the **Salesperson (Performance)** table allows analyzing sales made in the sales region(s) assigned to the salesperson.*

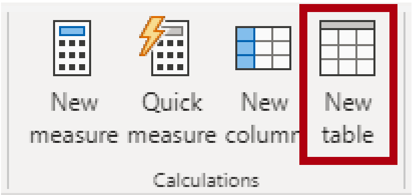
Task 3: Create the Date table

In this task you will create the **Date** table.

1. □ Switch to Data view.



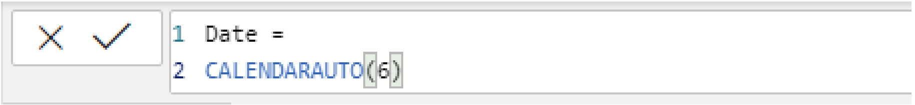
2. □ On the **Home** ribbon tab, from inside the **Calculations** group, click **New Table**.



3. □ In the formula bar, enter the following:

DAX

```
Date =
CALENDARAUTO(6)
```

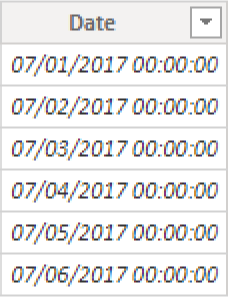


```
Date =
CALENDARAUTO(6)
```

The CALENDARAUTO() function returns a single-column table consisting of date values. The "auto" behavior scans all data model date columns to determine the earliest and latest date values stored in the data model. It then creates one row for each date within this range, extending the range in either direction to ensure full years of data is stored.

This function can take a single optional argument that is the last month number of a year. When omitted, the value is 12, meaning that December is the last month of the year. In this case, 6 is entered, meaning that June is the last month of the year.

4. □ Notice the column of date values.



Date
07/01/2017 00:00:00
07/02/2017 00:00:00
07/03/2017 00:00:00
07/04/2017 00:00:00
07/05/2017 00:00:00
07/06/2017 00:00:00

The dates shown are formatted using US regional settings (i.e. mm/dd/yyyy).

5. ☐ At the bottom-left corner, in the status bar, notice the table statistics, confirming that 1826 rows of data have been generated, which represents five full years' data.

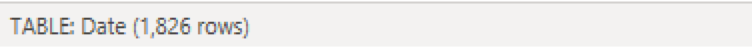


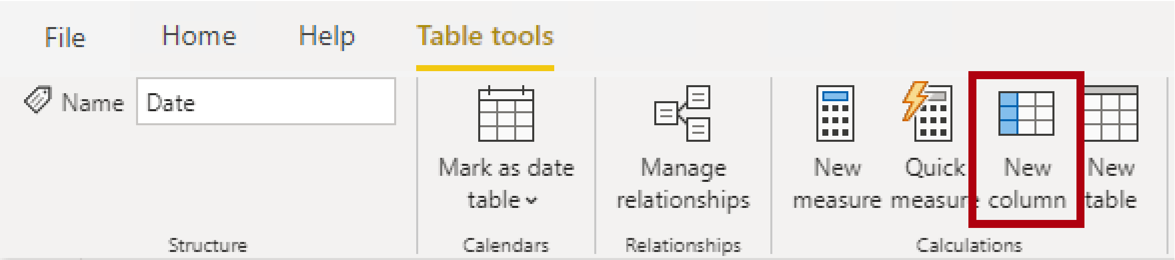
TABLE: Date (1,826 rows)

Task 4: Create calculated columns

In this task you will add additional columns to enable filtering and grouping by different time periods. You will also create a calculated column to control the sort order of other columns.

For your convenience, all DAX definitions in this lab can be copied from the snippets file, located in ***D:\DA100\Labs\05-create-dax-calculations-in-power-bi-desktop\Assets\Snippets.txt****.*

1. ☐ On the **Table Tools** contextual ribbon, from inside the **Calculations** group, click **New Column**.



Name: Date

2. ☐ In the formula bar, type the following (or copy from the snippets file), and then press **Enter**:

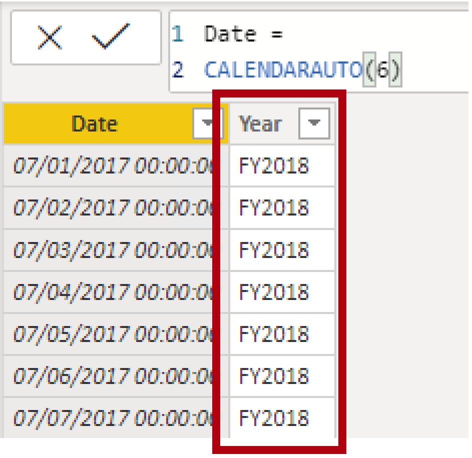
DAX

```
Year =
"FY" & YEAR('Date'[Date]) + IF(MONTH('Date'[Date]) > 6, 1)
```

A calculated column is created by first entering the column name, followed by the equals symbol (=), followed by a DAX formula that returns a single-value result. The column name cannot already exist in the table.

The formula uses the date's year value but adds one to the year value when the month is after June. It's how fiscal years at Adventure Works are calculated.

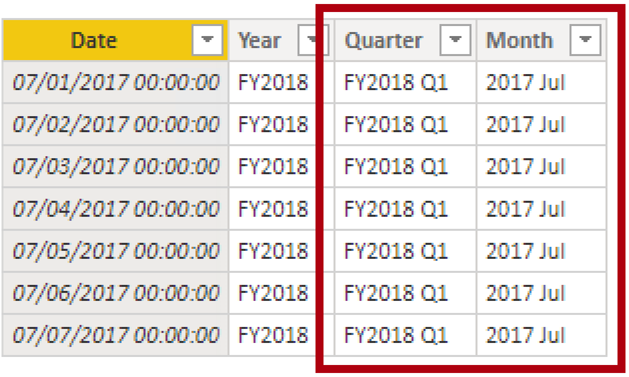
3. □ Verify that the new column was added.



```
Date =
CALENDARAUTO(6)
```

Date	Year
07/01/2017 00:00:0	FY2018
07/02/2017 00:00:0	FY2018
07/03/2017 00:00:0	FY2018
07/04/2017 00:00:0	FY2018
07/05/2017 00:00:0	FY2018
07/06/2017 00:00:0	FY2018
07/07/2017 00:00:0	FY2018

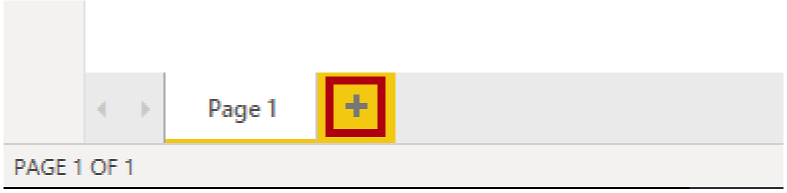
4. □ Use the snippets file definitions to create the following two calculated columns for the **Date** table:
   - Quarter
   - Month



Date	Year	Quarter	Month
07/01/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul
07/02/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul
07/03/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul
07/04/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul
07/05/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul
07/06/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul
07/07/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul

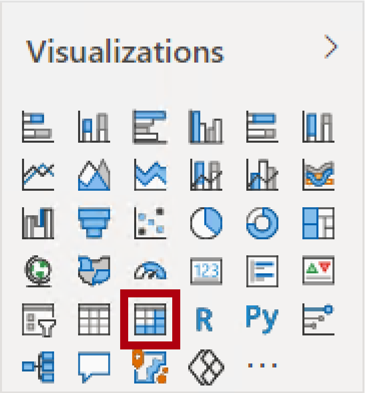
5. □ To validate the calculations, switch to Report view.

6. □ To create a new report page, at the bottom-left, click the plus icon.

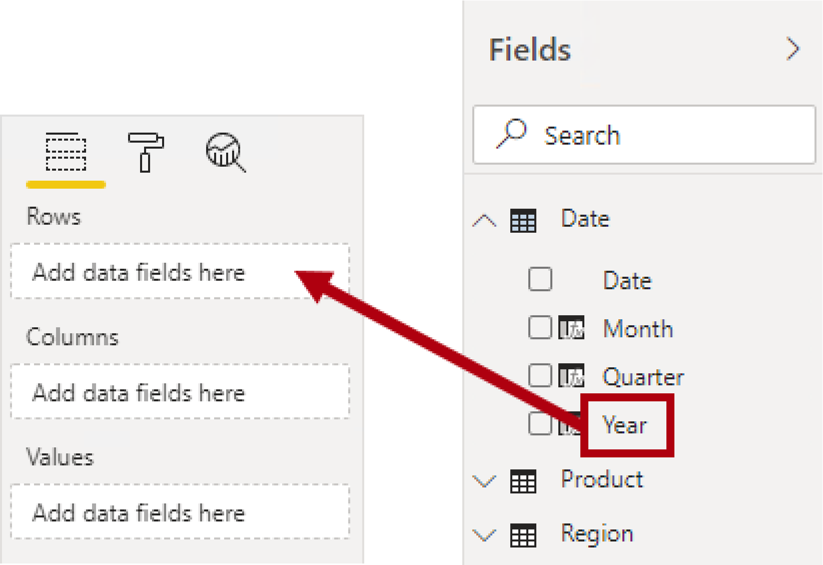


7. □ To add a matrix visual to the new report page, in the **Visualizations** pane, select the matrix visual type.

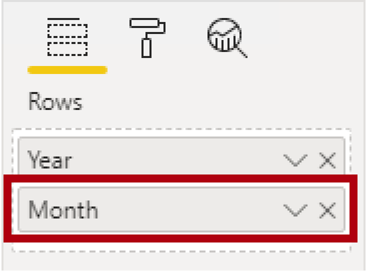
Tip: You can hover the cursor over each icon to reveal a tooltip describing the visual type.



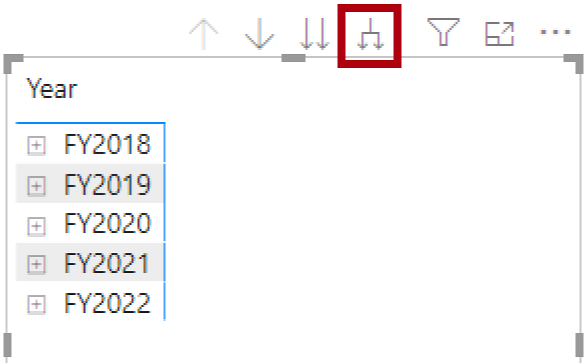
8. □ In the **Fields** pane, from inside the **Date** table, drag the **Year** field into the **Rows** well/area.



9. □ Drag the **Month** field into the **Rows** well/area, directly beneath the **Year** field.



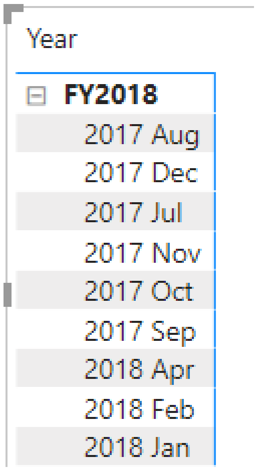
10. ☐ At the top-right of the matrix visual (or bottom, depending on the location of the visual), click the forked-double arrow icon (which will expand all years down one level).



Year

⊞ FY2018
⊞ FY2019
⊞ FY2020
⊞ FY2021
⊞ FY2022

11. ☐ Notice that the years expand to months, and that the months are sorted alphabetically rather than chronologically.



Year

⊟ **FY2018**
2017 Aug
2017 Dec
2017 Jul
2017 Nov
2017 Oct
2017 Sep
2018 Apr
2018 Feb
2018 Jan

By default, text values sort alphabetically, numbers sort from smallest to largest, and dates sort from earliest to latest.

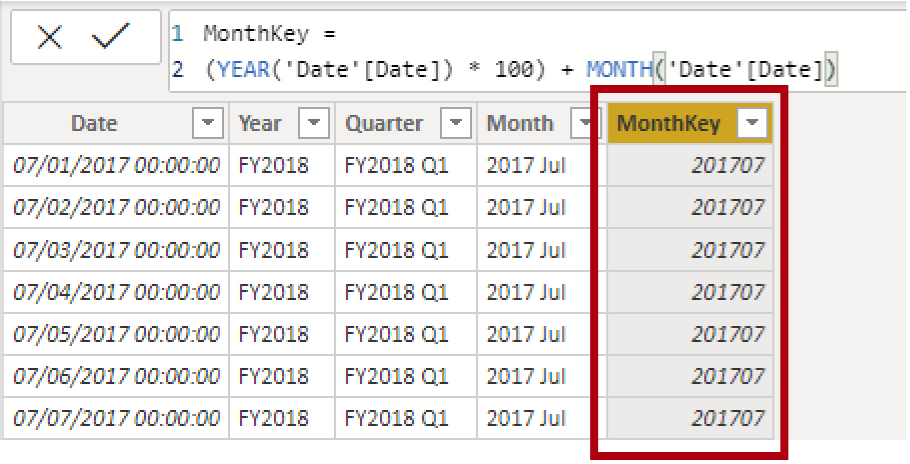
12. ☐ To customize the **Month** field sort order, switch to Data view.
13. ☐ Add the **MonthKey** column to the **Date** table.

DAX

```
MonthKey =
(YEAR('Date'[Date]) * 100) + MONTH('Date'[Date])
```

This formula computes a numeric value for each year/month combination.

14. □ In Data view, verify that the new column contains numeric values (e.g. 201707 for July 2017, etc.).



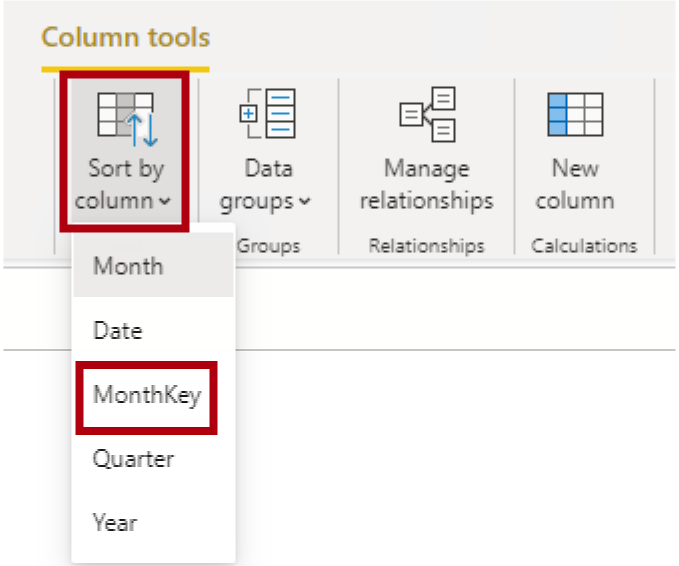
```
MonthKey =
(YEAR('Date'[Date]) * 100) + MONTH('Date'[Date])
```

Date	Year	Quarter	Month	MonthKey
07/01/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	201707
07/02/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	201707
07/03/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	201707
07/04/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	201707
07/05/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	201707
07/06/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	201707
07/07/2017 00:00:00	FY2018	FY2018 Q1	2017 Jul	201707

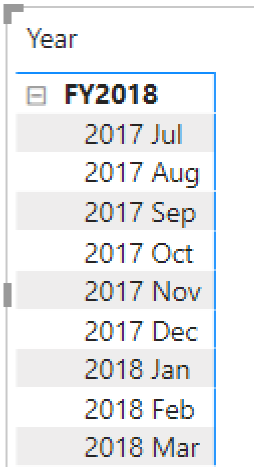
15. □ Switch back to Report view.

16. □ In the **Fields** pane, ensure that the **Month** field is selected (when selected, it will have a dark gray background).

17. □ On the **Column Tools** contextual ribbon, from inside the **Sort** group, click **Sort by Column**, and then select **MonthKey**.



18. □ In the matrix visual, notice that the months are now chronologically sorted.



Year

⊟ **FY2018**

2017 Jul

2017 Aug

2017 Sep

2017 Oct

2017 Nov

2017 Dec

2018 Jan

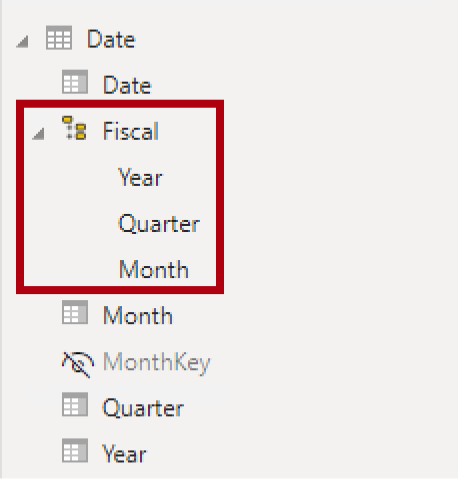
2018 Feb

2018 Mar

Task 5: Complete the Date table

In this task you will complete the design of the **Date** table by hiding a column and creating a hierarchy. You will then create relationships to the **Sales** and **Targets** tables.

1. □ Switch to Model view.
2. □ In the **Date** table, hide the **MonthKey** column (set **Is Hidden** to **Yes**).
3. □ In the **Date** table, create a hierarchy named **Fiscal**, with the following three levels:
   - Year
   - Quarter
   - Month



Date

Date

Fiscal

Year

Quarter

Month

Month

MonthKey

Quarter

Year

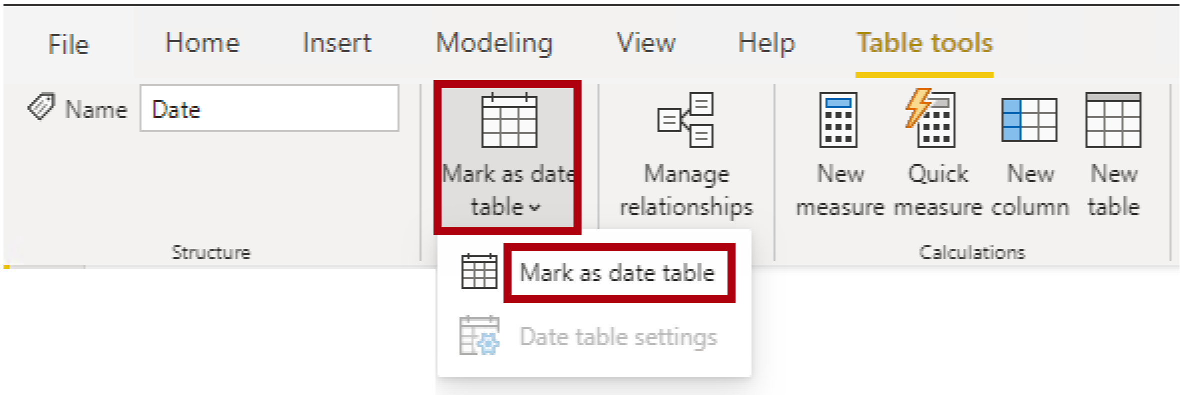
4. □ Create the following two model relationships:
   - **Date | Date** to **Sales | OrderDate**
   - **Date | Date** to **Targets | TargetMonth**
5. □ Hide the following two columns:

- Sales | OrderDate
- Targets | TargetMonth

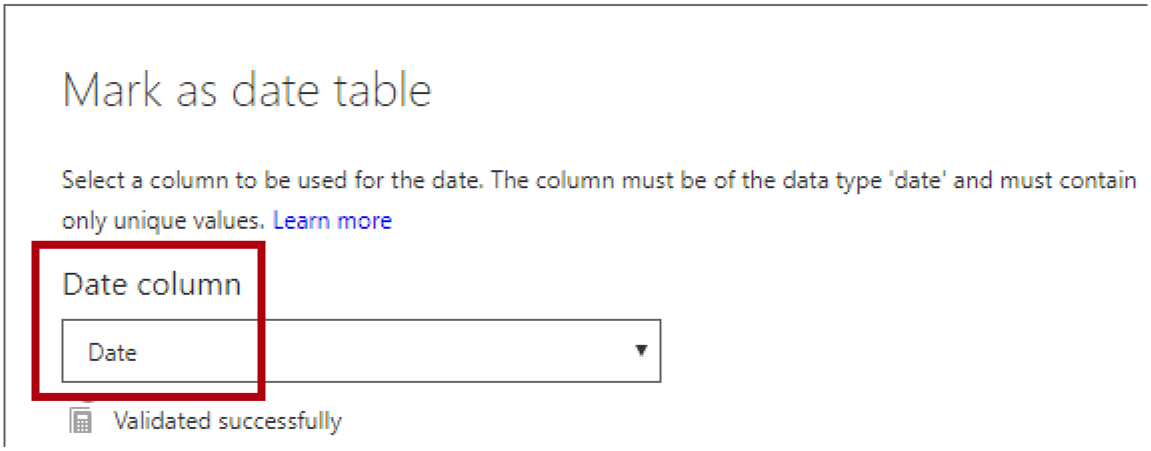
Task 6: Mark the Date table

In this task you will mark the **Date** table as a date table.

1. ☐ Switch to Report view.
2. ☐ In the **Fields** pane, select the **Date** table (not the **Date** field).
3. ☐ On the **Table Tools** contextual ribbon, from inside the **Calendars** group, click **Mark as Date Table**, and then select **Mark as Date Table**.



4. ☐ In the **Mark as Date Table** window, in the **Date Column** dropdown list, select **Date**.



Mark as date table

Select a column to be used for the date. The column must be of the data type 'date' and must contain only unique values. Learn more

Date column

Date

Validated successfully

5. ☐ Click **OK**.

6. □ Save the Power BI Desktop file.

Power BI Desktop now understands that this table defines date (time). It's important when relying on time intelligence calculations. You'll work with time intelligence calculations in the ***Create DAX Calculations in Power BI Desktop, Part 2*** *lab.*

Note that this design approach for a date table is suitable when you don't have a date table in your data source. If you have a data warehouse, it would be appropriate to load date data from its date dimension table rather than "redefining" date logic in your data model.

Exercise 2: Create Measures

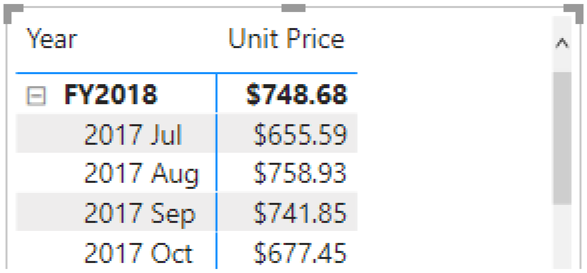
In this exercise you will create and format several measures.

Task 1: Create simple measures

In this task you will create simple measures. Simple measures aggregate values in a single column or count rows of a table.

1. □ In Report view, on **Page 2**, in the **Fields** pane, drag the **Sales | Unit Price** field into the matrix visual.

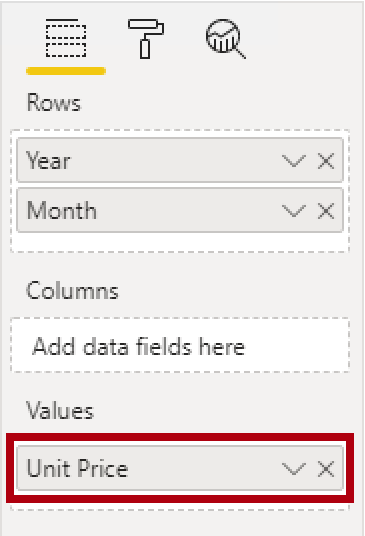
The labs use a shorthand notation to reference a field. It will look like this: ***Sales | Unit Price****. In this example,* ***Sales*** *is the table name and* ***Unit Price*** *is the field name.*



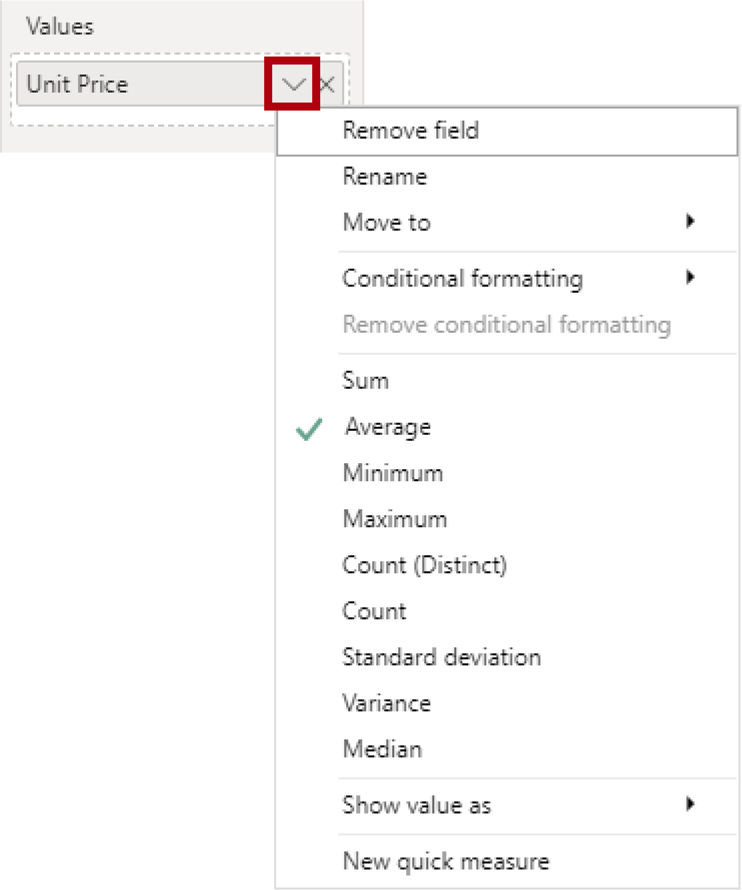
Year	Unit Price
⊟ **FY2018**	**$748.68**
2017 Jul	$655.59
2017 Aug	$758.93
2017 Sep	$741.85
2017 Oct	$677.45

You may recall that in the ***Model Data in Power BI Desktop, Part 2*** *lab, you set the* ***Unit Price*** *column to summarize by* ***Average****. The result you see in the matrix visual is the monthly average unit price (sum of unit price values divided by the count of unit prices).*

2. □ In the visual fields pane (located beneath the **Visualizations** pane), in the **Values** field well/area, notice that **Unit Price** is listed.

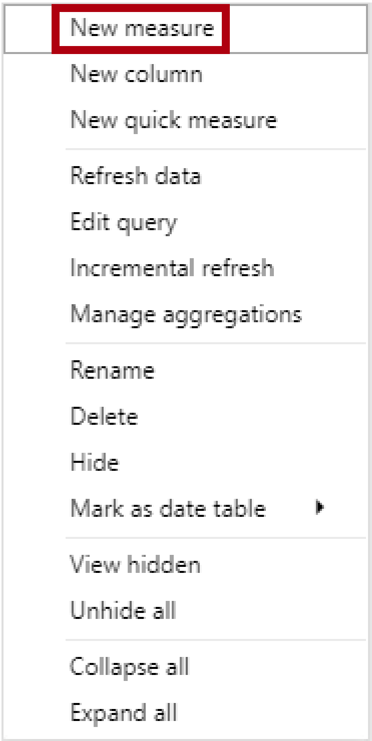


3. □ Click the down-arrow for **Unit Price**, and then notice the available menu options.



Visible numeric columns allow report authors at report design time to decide how column values will summarize (or not). It can result in inappropriate reporting. Some data modelers don't like *leaving things to chance, however, and choose to hide these columns and instead expose aggregation logic defined in measures. It's the approach you will now take in this lab.*

4. □ To create a measure, in the **Fields** pane, right-click the **Sales** table, and then select **New Measure**.

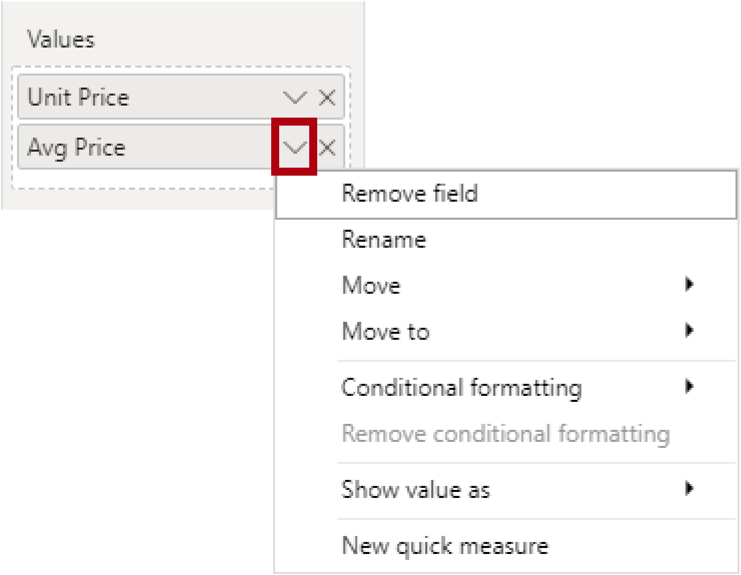


5. □ In the formula bar, add the following measure definition:

DAX

```
Avg Price =
AVERAGE(Sales[Unit Price])
```

6. □ Add the **Avg Price** measure to the matrix visual.
7. □ Notice that it produces the same result as the **Unit Price** column (but with different formatting).
8. □ In the **Values** well, open the context menu for the **Avg Price** field, and notice that it is not possible to change the aggregation technique.



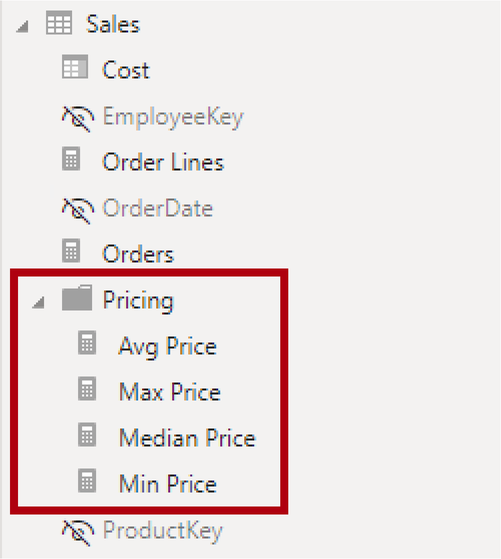
It's not possible to modify the aggregation behavior of a measure.

9. □ Use the snippets file definitions to create the following five measures for the **Sales** table:
   - Median Price
   - Min Price
   - Max Price
   - Orders
   - Order Lines

   *The DISTINCTCOUNT() function used in the **Orders** measure will count orders only once (ignoring duplicates). The COUNTROWS() function used in the **Order Lines** measure operates over a table.*

   *In this case, the number of orders is calculated by counting the distinct **SalesOrderNumber** column values, while the number of order lines is simply the number of table rows (each row is a line of an order).*

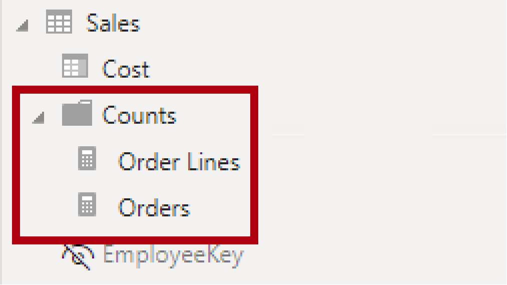
10. □ Switch to Model view, and then multi-select the four price measures: **Avg Price**, **Max Price**, **Median Price**, and **Min Price**.
11. □ For the multi-selection of measures, configure the following requirements:
    - Set the format to two decimal places
    - Assign to a display folder named **Pricing**



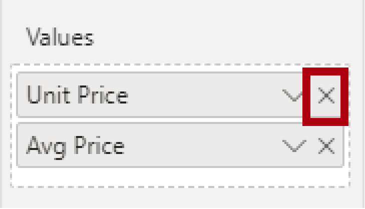
12. □ Hide the **Unit Price** column.

*The **Unit Price** column is now not available to report authors. They must use the pricing measures you've added to the model. This design approach ensures that report authors won't inappropriately aggregate prices, for example, by summing them.*

13. □ Multi-select the **Order Lines** and **Orders** measures, and then configure the following requirements:
    - Set the format use the thousands separator
    - Assign to a display folder named **Counts**



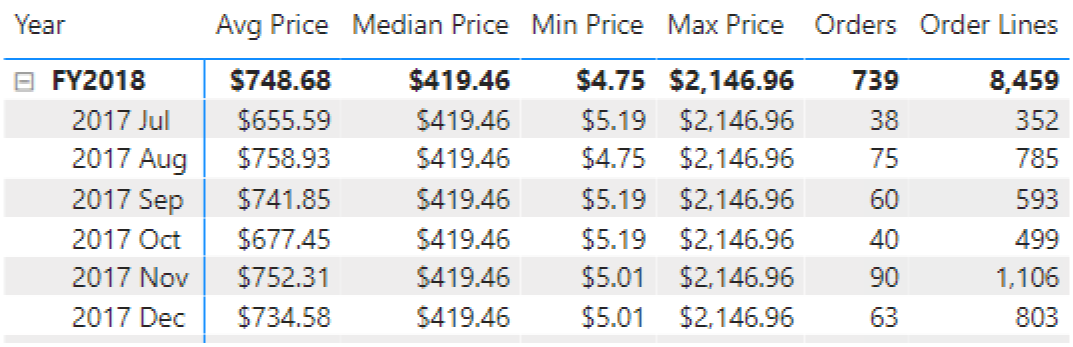
14. □ In Report view, in the **Values** well/area of the matrix visual, for the **Unit Price** field, click **X** to remove it.



15. □ Increase the size of the matrix visual to fill the page width and height.

16. □ Add the following five measures to the matrix visual:
    - Median Price
    - Min Price
    - Max Price
    - Orders
    - Order Lines

17. □ Verify that the results looks sensible and are correctly formatted.

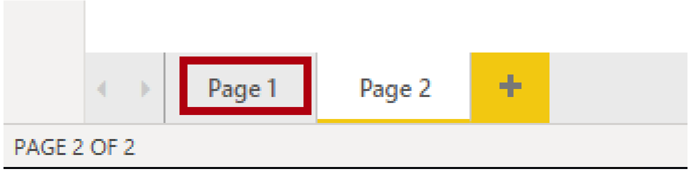


Year	Avg Price	Median Price	Min Price	Max Price	Orders	Order Lines
⊟ **FY2018**	**$748.68**	**$419.46**	**$4.75**	**$2,146.96**	**739**	**8,459**
2017 Jul	$655.59	$419.46	$5.19	$2,146.96	38	352
2017 Aug	$758.93	$419.46	$4.75	$2,146.96	75	785
2017 Sep	$741.85	$419.46	$5.19	$2,146.96	60	593
2017 Oct	$677.45	$419.46	$5.19	$2,146.96	40	499
2017 Nov	$752.31	$419.46	$5.01	$2,146.96	90	1,106
2017 Dec	$734.58	$419.46	$5.01	$2,146.96	63	803

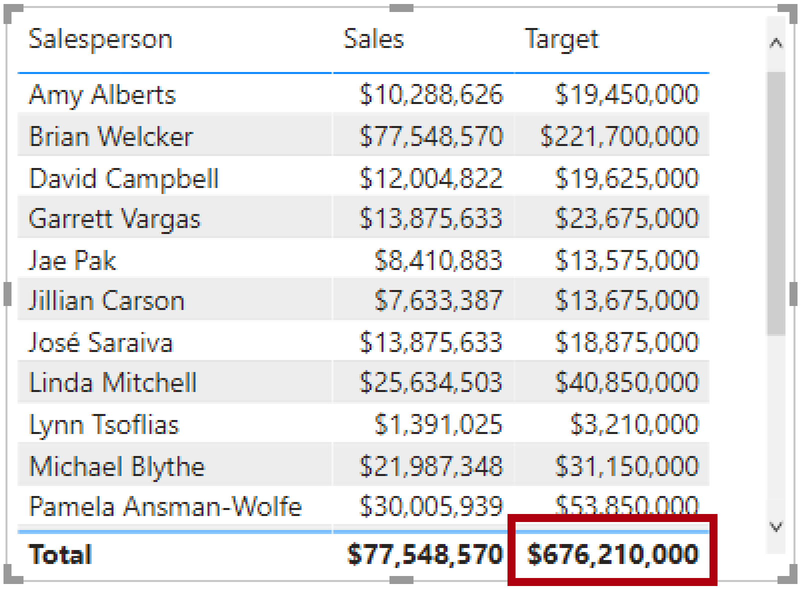
Task 2: Create additional measures

In this task you will create additional measures that use more complex formulas.

1. □ In Report view, select **Page 1**.



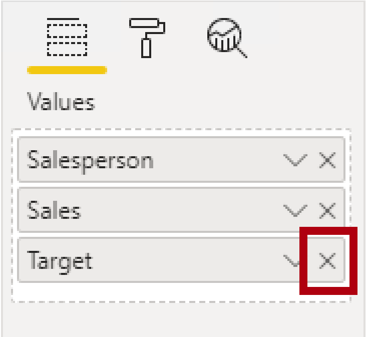
2. □ Review the table visual, noticing the total for the **Target** column.



Salesperson	Sales	Target
Amy Alberts	$10,288,626	$19,450,000
Brian Welcker	$77,548,570	$221,700,000
David Campbell	$12,004,822	$19,625,000
Garrett Vargas	$13,875,633	$23,675,000
Jae Pak	$8,410,883	$13,575,000
Jillian Carson	$7,633,387	$13,675,000
José Saraiva	$13,875,633	$18,875,000
Linda Mitchell	$25,634,503	$40,850,000
Lynn Tsoflias	$1,391,025	$3,210,000
Michael Blythe	$21,987,348	$31,150,000
Pamela Ansman-Wolfe	$30,005,939	$53,850,000
Total	**$77,548,570**	**$676,210,000**

You may recall from a previous lab that there's a many-to-many relationship between salespeople and regions. This means that summing the target values together doesn't make sense because salespeople targets are set for each salesperson based on their sales region assignment(s). A target value should only be shown when a single salesperson is filtered. You'll now implement a measure now to do just that.

3. □ Select the table visual, and then in the **Visualizations** pane, remove the **Target** field.



4. □ Rename the **Targets | Target** column as **Targets | TargetAmount**.

*Tip: There are several ways to rename the column in Report view: In the **Fields** pane, you can right-click the column, and then select **Rename**—or, double-click the column, or press **F2**.*

*You're about to create a measure named **Target**. It's not possible to have a column and measure in the same table with the same name.*

5. □ Create the following measure on the **Targets** table:

DAX

```
Target =

IF(

HASONEVALUE('Salesperson (Performance)'[Salesperson]),

SUM(Targets[TargetAmount])

)
```

*The HASONEVALUE() function tests whether a single value in the **Salesperson** column is filtered. When true, the expression returns the sum of target amounts (for just that salesperson). When false, BLANK is returned.*

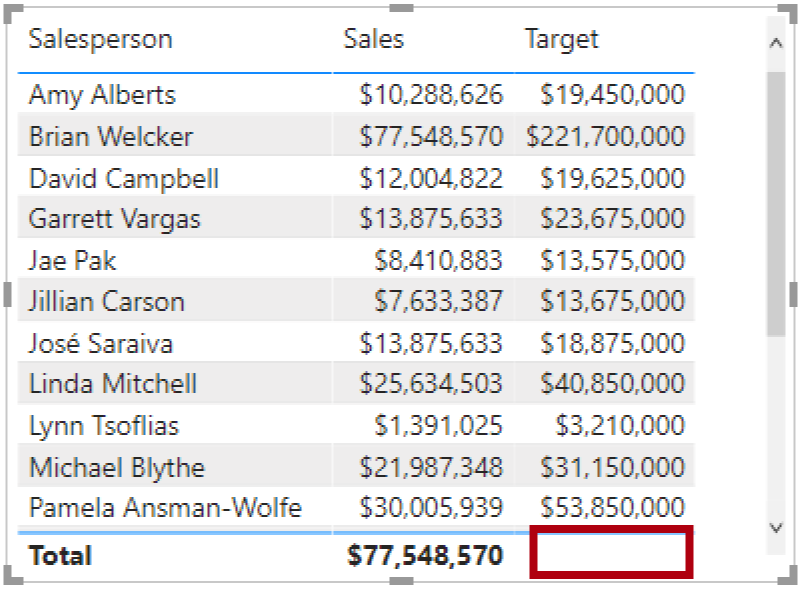
6. □ Format the **Target** measure for zero decimal places.

   *Tip: You can use the **Measure Tools** contextual ribbon.*

7. □ Hide the **TargetAmount** column.

   *Tip: You can right-click the column in the **Fields** pane, and then select **Hide**.*

8. □ Add the **Target** measure to the table visual.
9. □ Notice that the **Target** column total is now BLANK.



Salesperson	Sales	Target
Amy Alberts	$10,288,626	$19,450,000
Brian Welcker	$77,548,570	$221,700,000
David Campbell	$12,004,822	$19,625,000
Garrett Vargas	$13,875,633	$23,675,000
Jae Pak	$8,410,883	$13,575,000
Jillian Carson	$7,633,387	$13,675,000
José Saraiva	$13,875,633	$18,875,000
Linda Mitchell	$25,634,503	$40,850,000
Lynn Tsoflias	$1,391,025	$3,210,000
Michael Blythe	$21,987,348	$31,150,000
Pamela Ansman-Wolfe	$30,005,939	$53,850,000
Total	**$77,548,570**	

10. □ Use the snippets file definitions to create the following two measures for the **Targets** table:
    - Variance

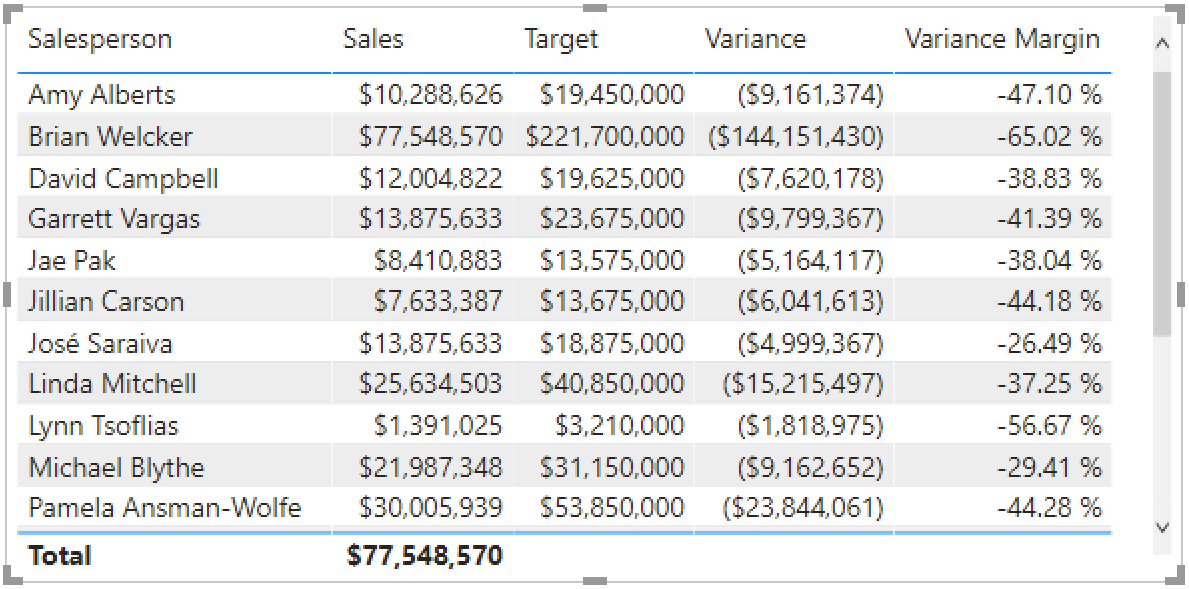
- Variance Margin

11. □ Format the **Variance** measure for zero decimal places.

12. □ Format the **Variance Margin** measure as percentage with two decimal places.

13. □ Add the **Variance** and **Variance Margin** measures to the table visual.

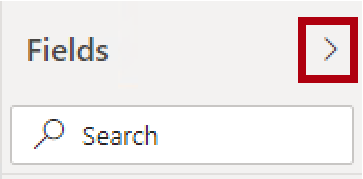
14. □ Resize the table visual so all columns and rows can be seen.



Salesperson	Sales	Target	Variance	Variance Margin
Amy Alberts	$10,288,626	$19,450,000	($9,161,374)	-47.10 %
Brian Welcker	$77,548,570	$221,700,000	($144,151,430)	-65.02 %
David Campbell	$12,004,822	$19,625,000	($7,620,178)	-38.83 %
Garrett Vargas	$13,875,633	$23,675,000	($9,799,367)	-41.39 %
Jae Pak	$8,410,883	$13,575,000	($5,164,117)	-38.04 %
Jillian Carson	$7,633,387	$13,675,000	($6,041,613)	-44.18 %
José Saraiva	$13,875,633	$18,875,000	($4,999,367)	-26.49 %
Linda Mitchell	$25,634,503	$40,850,000	($15,215,497)	-37.25 %
Lynn Tsoflias	$1,391,025	$3,210,000	($1,818,975)	-56.67 %
Michael Blythe	$21,987,348	$31,150,000	($9,162,652)	-29.41 %
Pamela Ansman-Wolfe	$30,005,939	$53,850,000	($23,844,061)	-44.28 %
Total	**$77,548,570**			

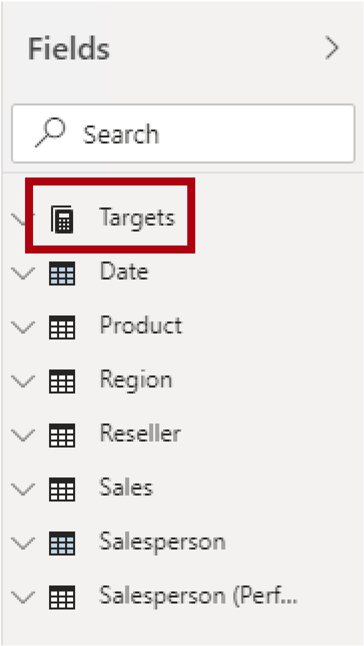
*While it appears all salespeople are not meeting target, remember that the table visual isn't yet filtered by a specific time period. You'll produce sales performance reports that filter by a user-selected time period in the **Design a Report in Power BI Desktop, Part 1** lab.*

15. □ At the top-right corner of the **Fields** pane, collapse and then expand open the pane.



Collapsing and re-opening the pane resets the content.

16. □ Notice that the **Targets** table now appears at the top of the list.



Tables that comprise only visible measures are automatically listed at the top of the list.

Task 3: Finish up

In this task you will complete the lab.

1. □ Save the Power BI Desktop file.
2. □ If you intend to start the next lab, leave Power BI Desktop open.

You'll enhance the data model with more advanced calculations using DAX in the ***Create DAX Calculations in Power BI Desktop, Part 2*** *lab.*

Congratulations

Create DAX Calculations in Power BI Desktop, Part 2

In this lab you will create measures with DAX expressions involving filter context manipulation.

In this lab you learn how to:

- Use the CALCULATE() function to manipulate filter context
- Use Time Intelligence functions

Exercise 1: Work with Filter Context

In this exercise you will create measures with DAX expressions involving filter context manipulation.

Task 1: Get started

In this task you will setup the environment for the lab.

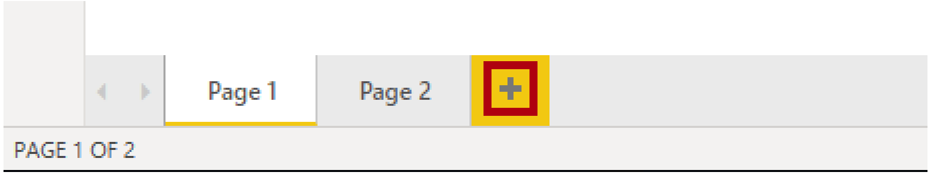
Important: If you are continuing on from the previous lab (and you completed that lab successfully), do not complete this task; instead, continue from the next task.

1. □ In the **Open** window, navigate to the **D:\DA100\Labs\06-create-dax-calculations-in-power-bi-desktop-advanced\Starter** folder.
2. □ Select the **Sales Analysis** file.

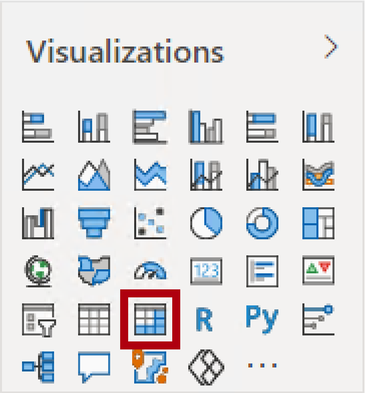
Task 2: Create a matrix visual

In this task you will create a matrix visual to support testing your new measures.

1. □ In Power BI Desktop, in Report view, create a new report page.



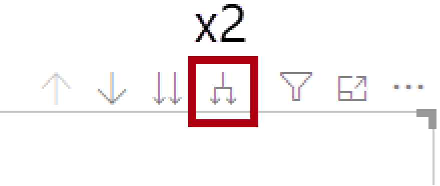
2. □ On **Page 3**, add a matrix visual.



3. □ Resize the matrix visual to fill the entire page.
4. □ To configure the matrix visual fields, from the **Fields** pane, drag the **Region | Regions** hierarchy, and drop it inside the visual.

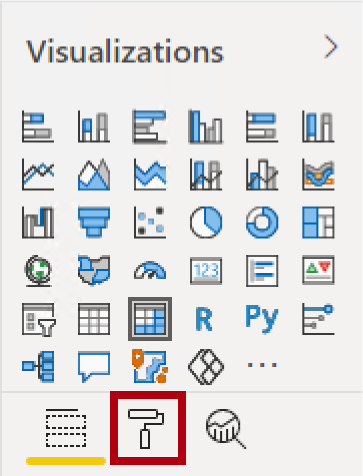
   The labs use a shorthand notation to reference a field or hierarchy. It will look like this: ***Region | Regions****. In this example,* ***Region*** *is the table name and* ***Regions*** *is the hierarchy name.*

5. □ Add also the **Sales | Sales** field.
6. □ To expand the entire hierarchy, at the top-right of the matrix visual, click the forked-double arrow icon twice.

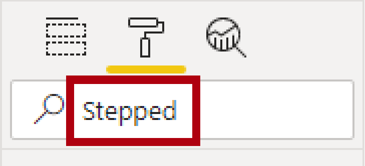


   You may recall that the ***Regions*** *hierarchy has the levels* ***Group****,* ***Country****, and* ***Region****.*

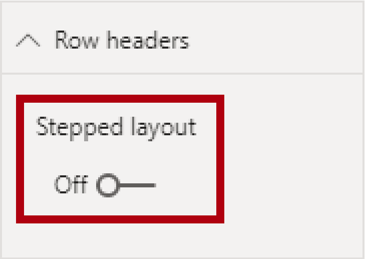
7. □ To format the visual, beneath the **Visualizations** pane, select the **Format** pane.



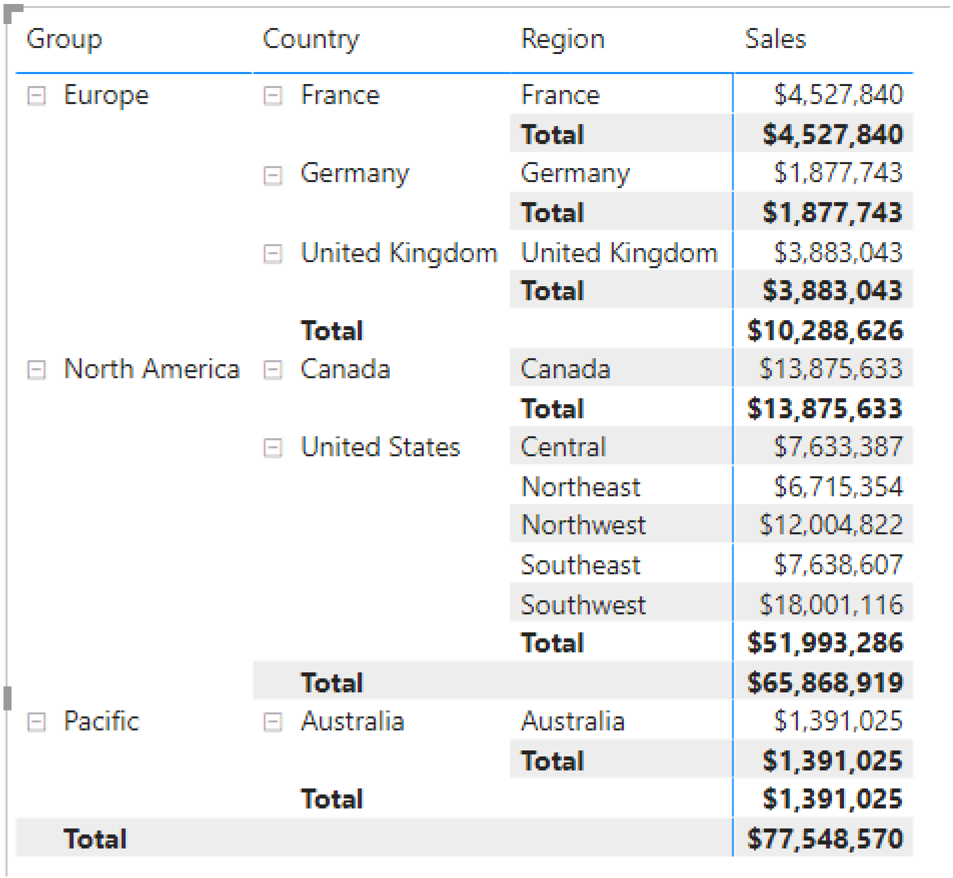
8. □ In the **Search** box, enter **Stepped**.


Stepped

9. □ Set the **Stepped Layout** property to **Off**.


Row headers
Stepped layout
Off

10. □ Verify that the matrix visual now has four column headers.



Group	Country	Region	Sales
⊟ Europe	⊟ France	France	$4,527,840
		Total	**$4,527,840**
	⊟ Germany	Germany	$1,877,743
		Total	**$1,877,743**
	⊟ United Kingdom	United Kingdom	$3,883,043
		Total	**$3,883,043**
	Total		**$10,288,626**
⊟ North America	⊟ Canada	Canada	$13,875,633
		Total	**$13,875,633**
	⊟ United States	Central	$7,633,387
		Northeast	$6,715,354
		Northwest	$12,004,822
		Southeast	$7,638,607
		Southwest	$18,001,116
		Total	**$51,993,286**
	Total		**$65,868,919**
⊟ Pacific	⊟ Australia	Australia	$1,391,025
		Total	**$1,391,025**
	Total		**$1,391,025**
Total			**$77,548,570**

At Adventure Works, the sales regions are organized into groups, countries, and regions. All countries—except the United States—have just one region, which is named after the country. As the United States is such a large sales territory, it's divided into five sales regions.

You'll create several measures in this exercise, and then test them by adding them to the matrix visual.

Task 3: Manipulate filter context

In this task you will create several measures with DAX expressions that use the CALCULATE() function to manipulate filter context.

1. □ Add a measure to the **Sales** table, based on the following expression:

   *For your convenience, all DAX definitions in this lab can be copied from the **D:\DA100\Labs\06-create-dax-calculations-in-power-bi-desktop-advanced\Assets\Snippets.txt** file.*

   DAX

   ```
   Sales All Region =
   ```

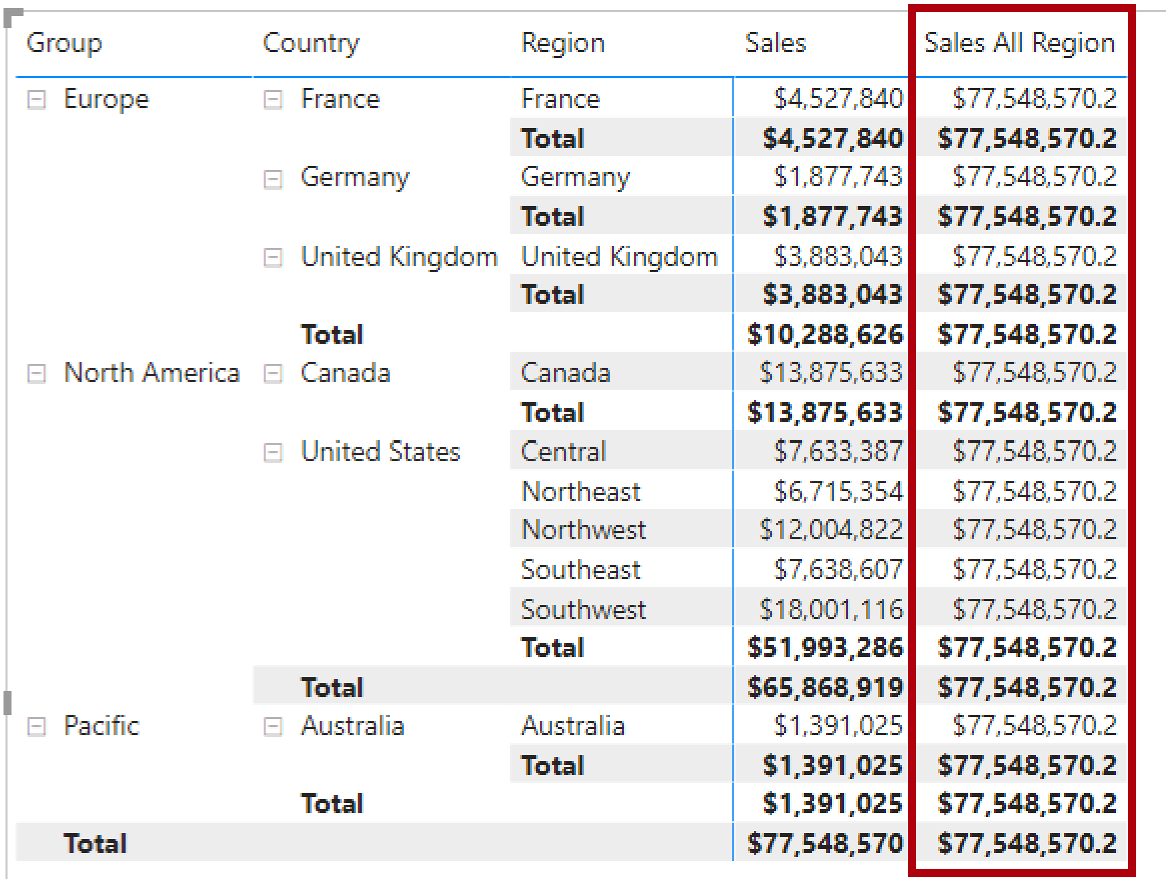
```
CALCULATE(SUM(Sales[Sales]), REMOVEFILTERS(Region))
```

The CALCULATE() function is a powerful function used to manipulate the filter context. The first argument takes an expression or a measure (a measure is just a named expression). Subsequent arguments allow modifying the filter context.

The REMOVEFILTERS() function removes active filters. It can take either no arguments, or a table, a column, or multiple columns as its argument.

In this formula, the measure evaluates the sum of the ***Sales*** *column in a modified filter context, which removes any filters applied to the columns of the* ***Region*** *table.*

2. □ Add the **Sales All Region** measure to the matrix visual.



Group	Country	Region	Sales	Sales All Region
⊟ Europe	⊟ France	France	$4,527,840	$77,548,570.2
		Total	**$4,527,840**	**$77,548,570.2**
	⊟ Germany	Germany	$1,877,743	$77,548,570.2
		Total	**$1,877,743**	**$77,548,570.2**
	⊟ United Kingdom	United Kingdom	$3,883,043	$77,548,570.2
		Total	**$3,883,043**	**$77,548,570.2**
	Total		**$10,288,626**	**$77,548,570.2**
⊟ North America	⊟ Canada	Canada	$13,875,633	$77,548,570.2
		Total	**$13,875,633**	**$77,548,570.2**
	⊟ United States	Central	$7,633,387	$77,548,570.2
		Northeast	$6,715,354	$77,548,570.2
		Northwest	$12,004,822	$77,548,570.2
		Southeast	$7,638,607	$77,548,570.2
		Southwest	$18,001,116	$77,548,570.2
		Total	**$51,993,286**	**$77,548,570.2**
	Total		**$65,868,919**	**$77,548,570.2**
⊟ Pacific	⊟ Australia	Australia	$1,391,025	$77,548,570.2
		Total	**$1,391,025**	**$77,548,570.2**
	Total		**$1,391,025**	**$77,548,570.2**
Total			**$77,548,570**	**$77,548,570.2**

3. □ Notice that the **Sales All Region** measure computes the total of all region sales for each region, country (subtotal) and group (subtotal).

The new measure is yet to deliver a useful result. When the sales for a group, country, or region is divided by this value it will produce a useful ratio known as "percent of grand total".

4. □ In the **Fields** pane, ensure that the **Sales All Region** measure is selected (when selected, it will have a dark gray background), and then in the formula bar, replace the measure name and formula with the following formula:

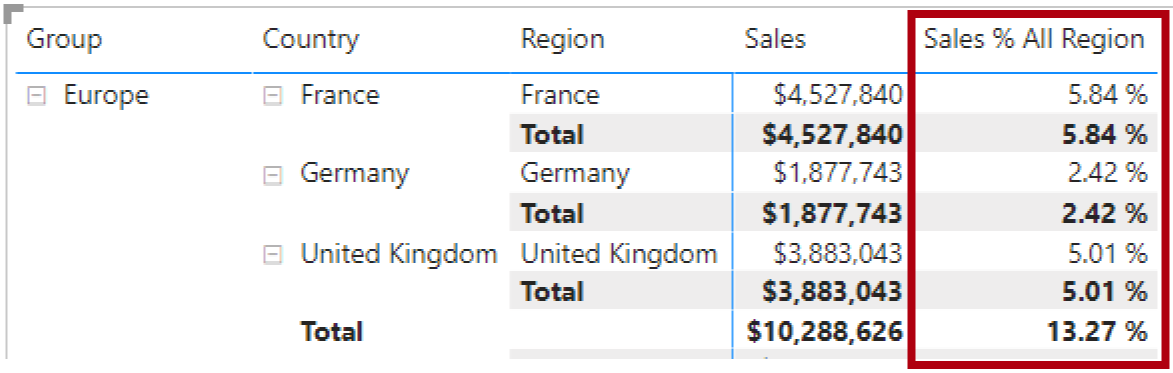
   *Tip: To replace the existing formula, first copy the snippet. Then, click inside the formula bar and press **Ctrl+A** to select all text. Then, press **Ctrl+V** to paste the snippet to overwrite the selected text. Then press **Enter**.*

   DAX

```
Sales % All Region =
DIVIDE(
 SUM(Sales[Sales]),
 CALCULATE(
 SUM(Sales[Sales]),
 REMOVEFILTERS(Region)
 )
)
```

   *The measure has been renamed to accurately reflect the updated formula. The DIVIDE() function divides the **Sales** measure (not modified by filter context) by the **Sales** measure in a modified context, which removes any filters applied to the **Region** table.*

5. □ In the matrix visual, notice that the measure has been renamed and that a different values now appear for each group, country, and region.
6. □ Format the **Sales % All Region** measure as a percentage with two decimal places.
7. □ In the matrix visual, review the **Sales % All Region** measure values.



Group	Country	Region	Sales	Sales % All Region
⊟ Europe	⊟ France	France	$4,527,840	5.84 %
		Total	**$4,527,840**	**5.84 %**
	⊟ Germany	Germany	$1,877,743	2.42 %
		Total	**$1,877,743**	**2.42 %**
	⊟ United Kingdom	United Kingdom	$3,883,043	5.01 %
		Total	**$3,883,043**	**5.01 %**
	Total		**$10,288,626**	**13.27 %**

8. □ Add another measure to the **Sales** table, based on the following expression, and format as a percentage:

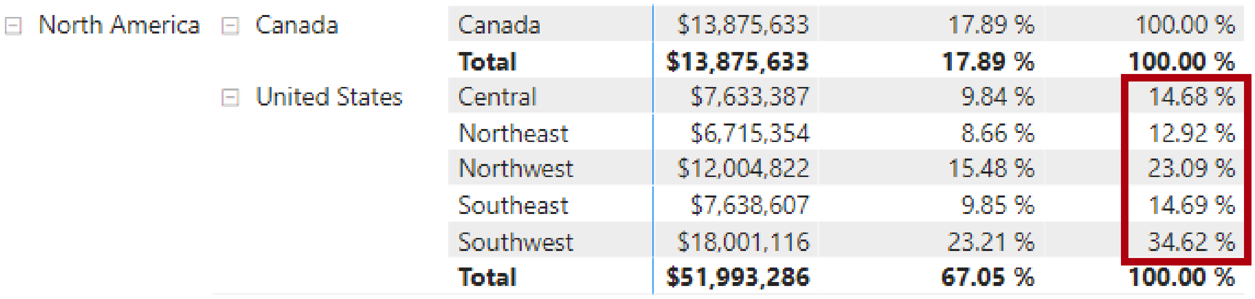
   DAX

```
Sales % Country =
DIVIDE(
 SUM(Sales[Sales]),
 CALCULATE(
 SUM(Sales[Sales]),
 REMOVEFILTERS(Region[Region])
 )
)
```

9. □ Notice that the **Sales % Country** measure formula differs slightly from the **Sales % All Region** measure formula.

*The difference is that the denominator modifies the filter context by removing filters on the **Region** column of the **Region** table, not all columns of the **Region** table. It means that any filters applied to the group or country columns are preserved. It'll achieve a result that represents the sales as a percentage of country.*

10. □ Add the **Sales % Country** measure to the matrix visual.
11. □ Notice that only the United States' regions produce a value which is not 100%.



⊟ North America	⊟ Canada	Canada	$13,875,633	17.89 %	100.00 %
		Total	**$13,875,633**	**17.89 %**	**100.00 %**
	⊟ United States	Central	$7,633,387	9.84 %	14.68 %
		Northeast	$6,715,354	8.66 %	12.92 %
		Northwest	$12,004,822	15.48 %	23.09 %
		Southeast	$7,638,607	9.85 %	14.69 %
		Southwest	$18,001,116	23.21 %	34.62 %
		Total	**$51,993,286**	**67.05 %**	**100.00 %**

You may recall that only the United States has multiple regions. All other countries comprise a single region, which explains why they are all 100%.

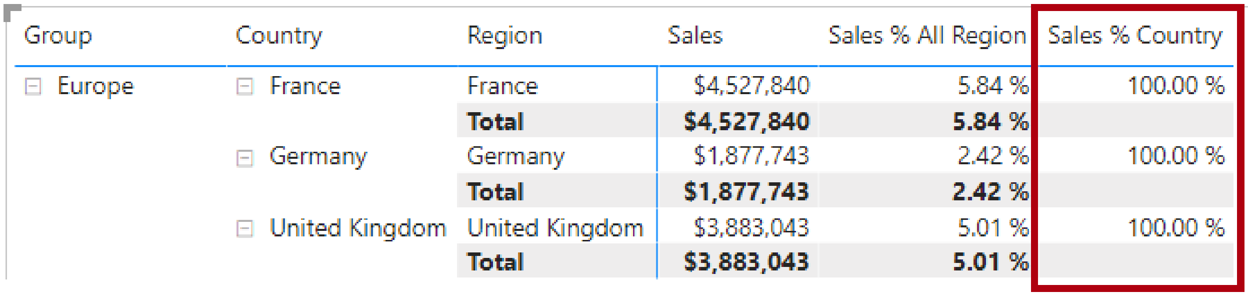
12. □ To improve the readability of this measure in visual, overwrite the **Sales % Country** measure with this improved formula.

DAX

```
Sales % Country =
IF(
 ISINSCOPE(Region[Region]),
 DIVIDE(
 SUM(Sales[Sales]),
 CALCULATE(
 SUM(Sales[Sales]),
 REMOVEFILTERS(Region[Region])
 )
 )
)
```

Embedded within the IF() function, the ISINSCOPE() function is used to test whether the region column is the level in a hierarchy of levels. When true, the DIVIDE() function is evaluated. The absence of a false part means that blank is returned when the region column is not in scope.

13. □ Notice that the **Sales % Country** measure now only returns a value when a region is in scope.



Group	Country	Region	Sales	Sales % All Region	Sales % Country
⊟ Europe	⊟ France	France	$4,527,840	5.84 %	100.00 %
		Total	**$4,527,840**	**5.84 %**	
	⊟ Germany	Germany	$1,877,743	2.42 %	100.00 %
		Total	**$1,877,743**	**2.42 %**	
	⊟ United Kingdom	United Kingdom	$3,883,043	5.01 %	100.00 %
		Total	**$3,883,043**	**5.01 %**	

14. □ Add another measure to the **Sales** table, based on the following expression, and format as a percentage:

DAX

```
Sales % Group =
DIVIDE(
 SUM(Sales[Sales]),
 CALCULATE(
 SUM(Sales[Sales]),
 REMOVEFILTERS(
 Region[Region],
 Region[Country]
 )
 )
)
```

To achieve sales as a percentage of group, two filters can be applied to effectively remove the filters on two columns.

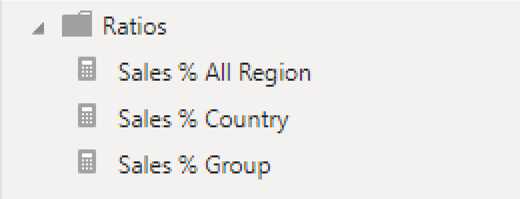
15. □ Add the **Sales % Group** measure to the matrix visual.

16. □ To improve the readability of this measure in visual, overwrite the **Sales % Group** measure with this improved formula.

DAX

```
Sales % Group =
IF(
 ISINSCOPE(Region[Region])
 || ISINSCOPE(Region[Country]),
 DIVIDE(
 SUM(Sales[Sales]),
 CALCULATE(
 SUM(Sales[Sales]),
 REMOVEFILTERS(
 Region[Region],
 Region[Country]
 )
 )
 )
)
```

17. □ Notice that the **Sales % Group** measure now only returns a value when a region or country is in scope.
18. □ In Model view, place the three new measures into a display folder named **Ratios**.



Ratios

Sales % All Region

Sales % Country

Sales % Group

19. □ Save the Power BI Desktop file.

*The measures added to the **Sales** table have modified filter context to achieve hierarchical navigation. Notice that the pattern to achieve the calculation of a subtotal requires removing some columns from the filter context, and to arrive at a grand total, all columns must be removed.*

Exercise 2: Work with Time Intelligence

In this exercise you will create a sales year-to-date (YTD) measure and sales year-over-year (YoY) growth measure.

Task 1: Create a YTD measure

In this task you will create a sales YTD measure.

1. □ In Report view, on **Page 2**, notice the matrix visual that displays various measures with years and months grouped on the rows.
2. □ Add a measure to the **Sales** table, based on the following expression, and formatted to zero decimal places:

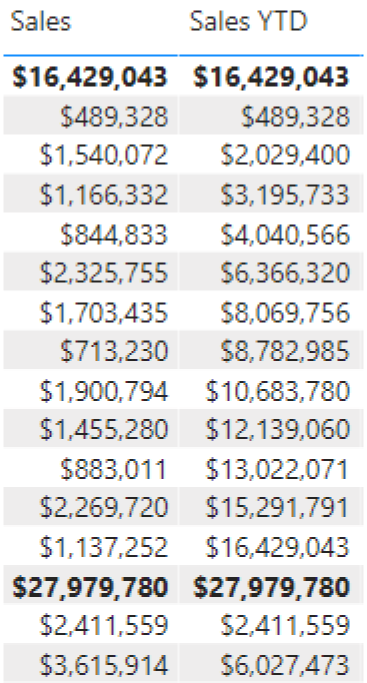
DAX

```
Sales YTD =
TOTALYTD(SUM(Sales[Sales]), 'Date'[Date], "6-30")
```

The TOTALYTD() function evaluates an expression—in this case the sum of the ***Sales*** *column—over a given date column. The date column must belong to a date table marked as a date table, as was done in the* ***Create DAX Calculations in Power BI Desktop, Part 1*** *lab.*

The function can also take a third optional argument representing the last date of a year. The absence of this date means that December 31 is the last date of the year. For Adventure Works, June in the last month of their year, and so "6-30" is used.

3. ☐ Add the **Sales** field and the **Sales YTD** measure to the matrix visual.
4. ☐ Notice the accumulation of sales values within the year.



Sales	Sales YTD
$16,429,043	**$16,429,043**
$489,328	$489,328
$1,540,072	$2,029,400
$1,166,332	$3,195,733
$844,833	$4,040,566
$2,325,755	$6,366,320
$1,703,435	$8,069,756
$713,230	$8,782,985
$1,900,794	$10,683,780
$1,455,280	$12,139,060
$883,011	$13,022,071
$2,269,720	$15,291,791
$1,137,252	$16,429,043
$27,979,780	**$27,979,780**
$2,411,559	$2,411,559
$3,615,914	$6,027,473

The TOTALYTD() function performs filter manipulation, specifically time filter manipulation. For example, to compute YTD sales for September 2017 (the third month of the fiscal year), all filters on the ***Date*** *table are removed and replaced with a new filter of dates commencing at the beginning of the year (July 1, 2017) and extending through to the last date of the in-context date period (September 30, 2017).*

Note that many Time Intelligence functions are available in DAX to support common time filter manipulations.

Task 2: Create a YoY growth measureSales

In this task you will create a sales YoY growth measure.

1. ☐ Add an additional measure to the **Sales** table, based on the following expression:

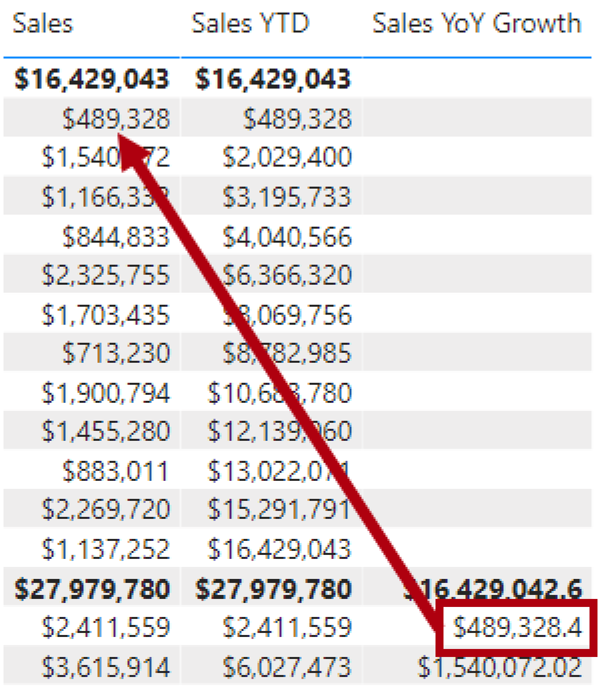
DAX

```
Sales YoY Growth =
VAR SalesPriorYear =
 CALCULATE(
 SUM(Sales[Sales]),
 PARALLELPERIOD(
 'Date'[Date],
 -12,
 MONTH
 )
 )
RETURN
 SalesPriorYear
```

*The **Sales YoY Growth** measure formula declares a variable. Variables can be useful for simplifying the formula logic, and more efficient when an expression needs to be evaluated multiple times within the formula (which will be the case for the YoY growth logic). Variables are declared by a unique name, and the measure expression must then be output after the **RETURN** keyword.*

*The **SalesPriorYear** variable is assigned an expression which calculates the sum of the **Sales** column in a modified context that uses the PARALLELPERIOD() function to shift 12 months back from each date in filter context.*

2. □ Add the **Sales YoY Growth** measure to the matrix visual.
3. □ Notice that the new measure returns BLANK for the first 12 months (because there were no sales recorded before fiscal year 2017).
4. □ Notice that the **Sales YoY Growth** measure value for **2018 Jul** is the **Sales** value for **2018 Jul**.



Sales	Sales YTD	Sales YoY Growth
$16,429,043	**$16,429,043**	
$489,328	$489,328	
$1,540,072	$2,029,400	
$1,166,333	$3,195,733	
$844,833	$4,040,566	
$2,325,755	$6,366,320	
$1,703,435	$8,069,756	
$713,230	$8,782,985	
$1,900,794	$10,683,780	
$1,455,280	$12,139,060	
$883,011	$13,022,071	
$2,269,720	$15,291,791	
$1,137,252	$16,429,043	
$27,979,780	**$27,979,780**	**$16,429,042.6**
$2,411,559	$2,411,559	$489,328.4
$3,615,914	$6,027,473	$1,540,072.02

Now that the "difficult part" of the formula has been tested, you can overwrite the measure with the final formula which computes the growth result.

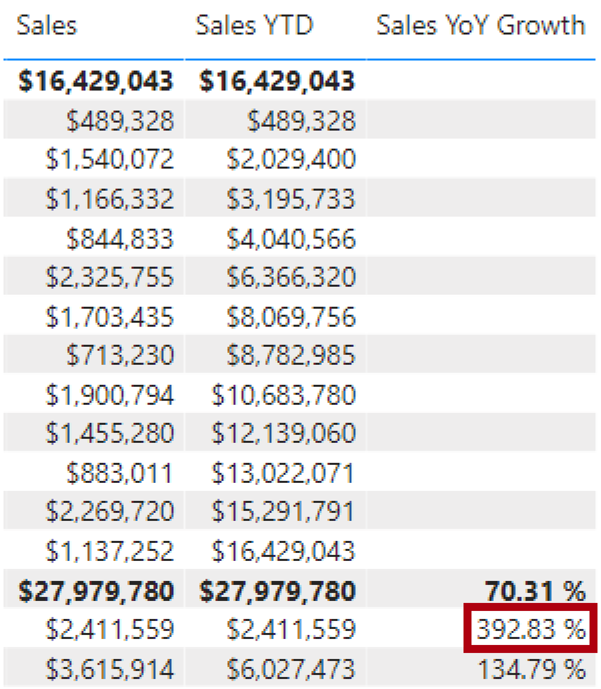
5. ☐ To complete the measure, overwrite the **Sales YoY Growth** measure with this formula, formatting it as a percentage with two decimal places:

DAX

```
Sales YoY Growth =
VAR SalesPriorYear =
 CALCULATE(
 SUM(Sales[Sales]),
 PARALLELPERIOD(
 'Date'[Date],
 -12,
 MONTH
 )
 )
RETURN
 DIVIDE(
 (SUM(Sales[Sales]) - SalesPriorYear),
 SalesPriorYear
 )
```

6. ☐ In the formula, in the **RETURN** clause, notice that the variable is referenced twice.

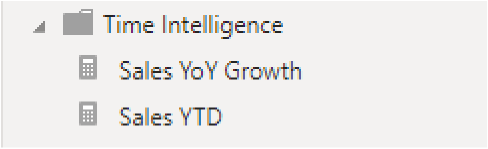
7. ☐ Verify that the YoY growth for **2018 Jul** is **392.83%**.



Sales	Sales YTD	Sales YoY Growth
$16,429,043	**$16,429,043**	
$489,328	$489,328	
$1,540,072	$2,029,400	
$1,166,332	$3,195,733	
$844,833	$4,040,566	
$2,325,755	$6,366,320	
$1,703,435	$8,069,756	
$713,230	$8,782,985	
$1,900,794	$10,683,780	
$1,455,280	$12,139,060	
$883,011	$13,022,071	
$2,269,720	$15,291,791	
$1,137,252	$16,429,043	
$27,979,780	**$27,979,780**	**70.31 %**
$2,411,559	$2,411,559	392.83 %
$3,615,914	$6,027,473	134.79 %

It means that July 2018 sales ($2,411,559) represents a nearly 400% (almost 4x) improvement over the sales achieved at the same time of the prior year ($489,328).

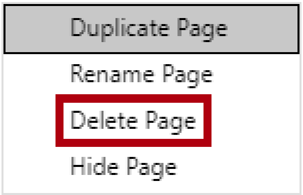
8. □ In Model view, place the two new measures into a display folder named **Time Intelligence**.



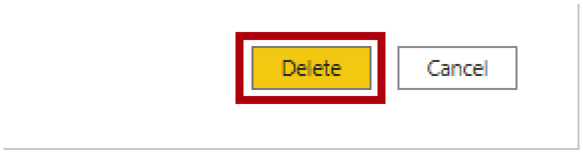
Task 3: Finish up

In this task you will complete the lab.

1. □ To clean up the solution ready for report development, at the bottom-left, right-click the **Page 2** tab, and then select **Delete** page.



2. □ When prompted to delete the page, click **Delete**.



3. □ Delete **Page 3** also.
4. □ On the remaining page, to clear the page, select the table visual, and the press the **Delete** key.
5. □ Save the Power BI Desktop file.
6. □ If you intend to start the next lab, leave Power BI Desktop open.

*You'll create a report based on the data model in the **Design a Report in Power BI Desktop, Part 1** lab.*

Congratulations!

Design a Report in Power BI Desktop, Part 1

In this l

ab you will create a three-page report. You will then publish it to Power BI, whereupon you will open and interact with the report.

In this lab you learn how to:

- Design a report
- Configure visual fields and format properties

Exercise 1: Create a Report

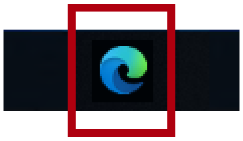
In this exercise you will create a three-page report named **Sales Report**.

Task 1: Get started – Sign in

In this task you will setup the environment for the lab by signing in to Power BI.

Important: If you have already signed in to Power BI, continue from the next task.

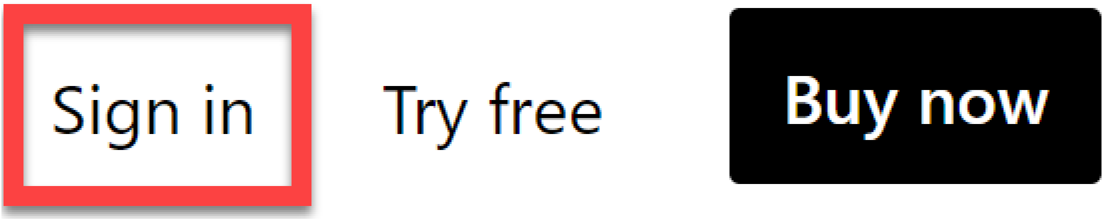
1. □ To open Microsoft Edge, on the taskbar, click the Microsoft Edge program shortcut.



2. □ In the Microsoft Edge browser window, navigate to **https://powerbi.com**.

   Tip: You can also use the Power BI Service favorite on the Microsoft Edge favorites bar.

3. □ Click **Sign In** (located at the top-right corner).



4. □ Enter the account details provided to you (check out the **Resources**).
5. □ If prompted to update the password, reenter the provided password, and then enter and confirm a new password.

   Important: Be sure to record your new password.

6. □ Complete the sign in process.
7. □ If prompted by Microsoft Edge to stay signed in, click **Yes**.
8. □ Leave the Microsoft Edge browser window open.

Task 2: Get started – Open report

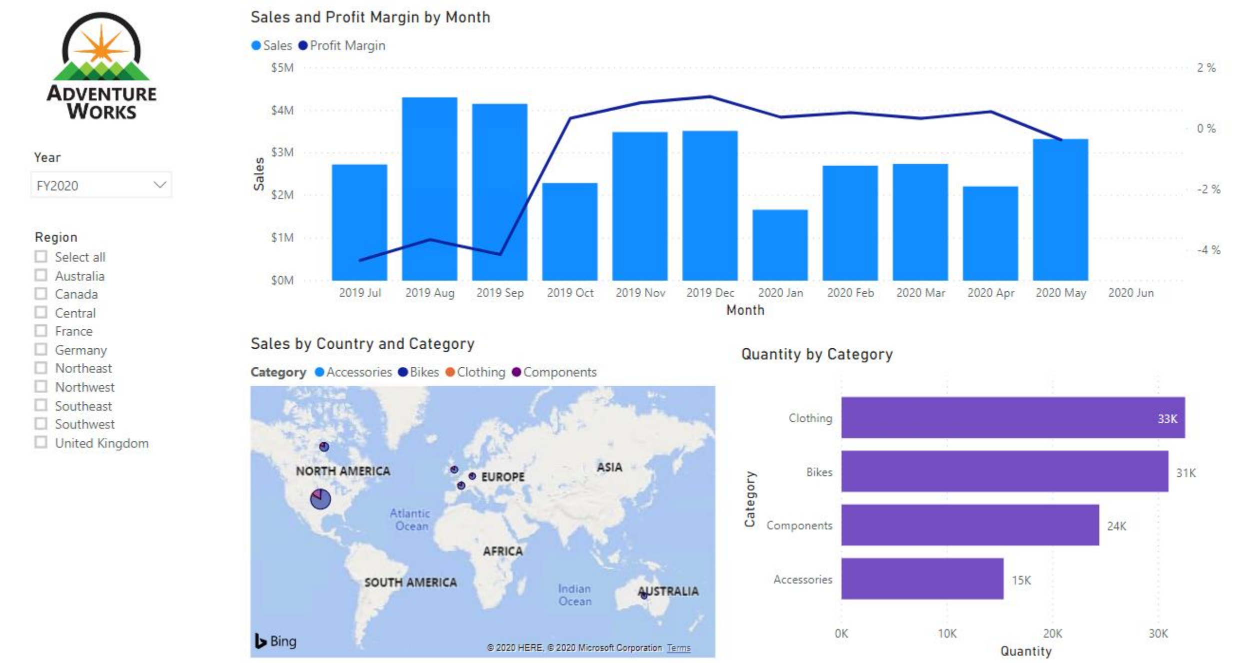
In this task you will setup the environment for the lab by opening the starter report.

Important: If you are continuing on from the previous lab (and you completed that lab successfully), do not complete this task; instead, continue from the next task.

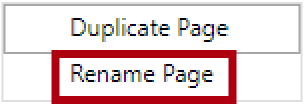
1. □ In the **Open** window, navigate to the **D:\DA100\Labs\07-design-report-in-power-bi-desktop\Starter** folder.
2. □ Select the **Sales Analysis** file.

Task 3: Design page 1

In this task you will design the first report page. When you've completed the design, the page will look like the following:

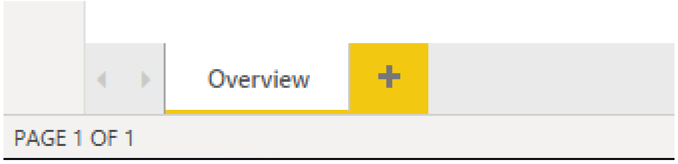


1. □ In Power BI Desktop, to rename the page, at the bottom-left, right-click **Page 1**, and then select **Rename**.

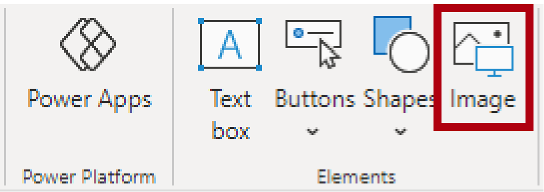


Tip: You can also double-click the page name to rename it.

2. □ Rename the page as **Overview**, and then press **Enter**.

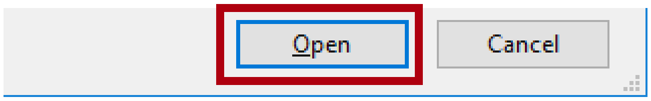


3. □ To add an image, on the **Insert** ribbon tab, from inside the **Elements** group, click **Image**.

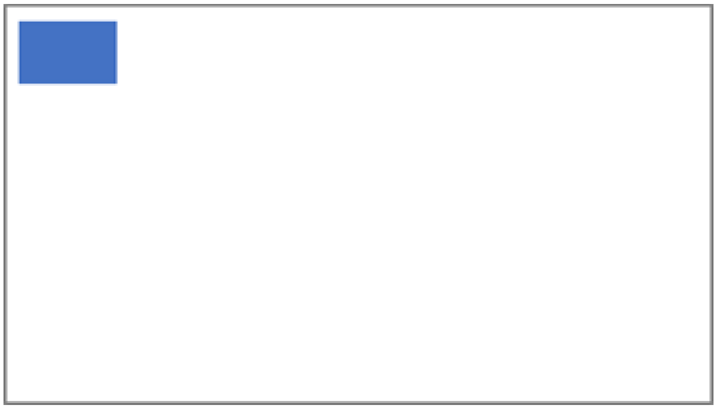


4. □ In the **Open** window, navigate to the **D:\DA100\Resources** folder.

5. □ Select the **AdventureWorksLogo.jpg** file, and then click **Open**.



6. □ Drag the image to position it at the top-left corner, and also drag the guide markers to resize it.



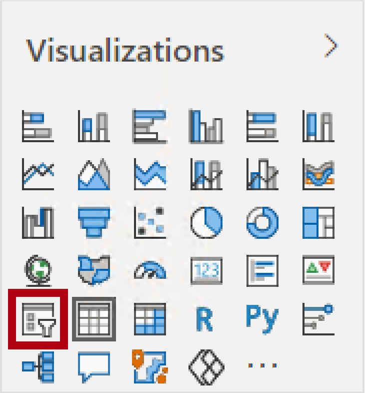
7. □ To add a slicer, first de-select the image by clicking an empty area of the report page.

8. □ In the **Fields** pane, select the **Date | Year** field (not the **Year** level of the hierarchy).

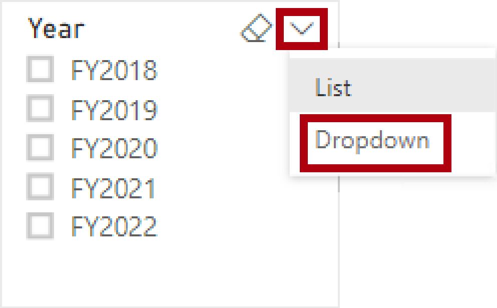
The labs use a shorthand notation to reference a field. It will look like this: ***Date* | *Year****. In this example,* ***Date*** *is the table name and* ***Year*** *is the field name.*

9. □ Notice that a table of year values has been added to the report page.

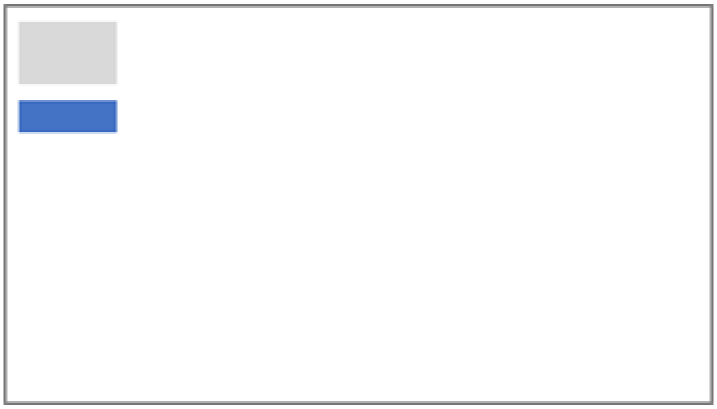
10. □ To convert the visual from a table to a slicer, in the **Visualizations** pane, select the **Slicer**.



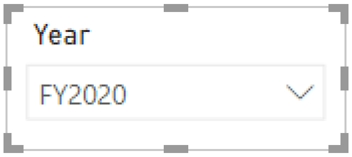
11. □ To convert the slicer from a list to a dropdown, at the top-right of the slicer, click the down-arrow, and then select **Dropdown**.



12. □ Resize and position the slicer so it sits beneath the image and is the same width as the image.



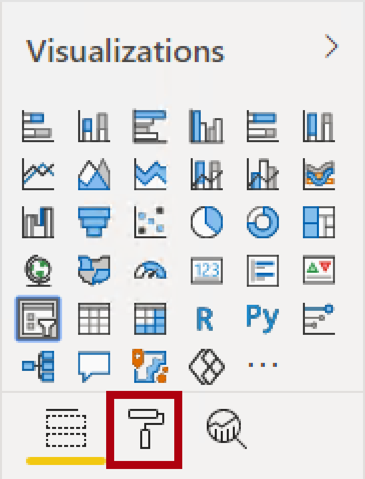
13. □ In the **Year** slicer, open the dropdown list, select **FY2020**, and then collapse the dropdown list.



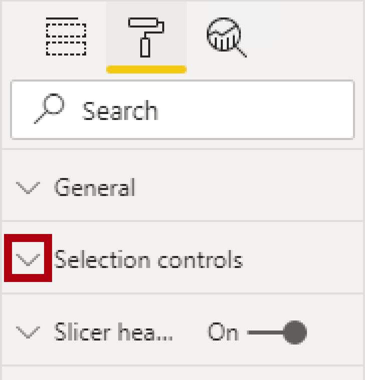
*The report page is now filtered by year **FY2020**.*

14. □ De-select the slicer by clicking an empty area of the report page.
15. □ Create a second slicer, based on the **Region | Region** field (not the **Region** level of the hierarch).
16. □ Leave the slicer as a list, and then resize and position the slicer beneath the **Year** slicer.

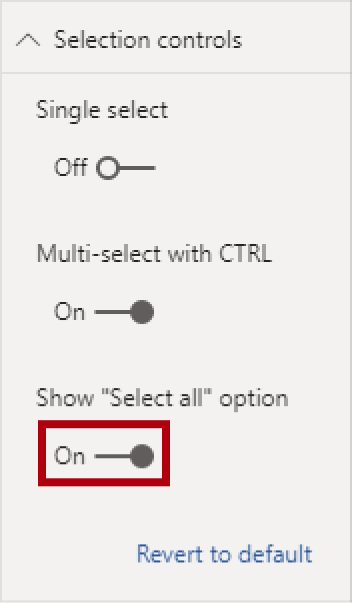
17. □ To format the slicer, beneath the **Visualizations** pane, open the **Format** pane.



18. □ Expand the **Selection Controls** group.



19. □ Set the **Show "Select All" Option** to **On**.

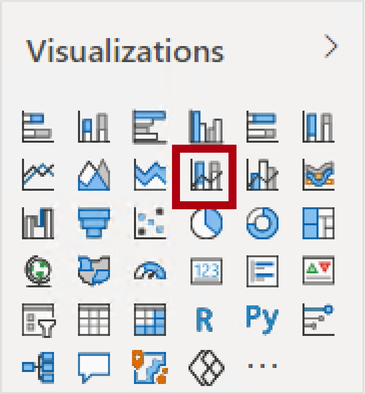


20. □ In the **Region** slicer, notice that the first item is now **Select All**.

When selected, this item either selects all or de-selects all items. It makes it easier for report users to set the required slicer items.

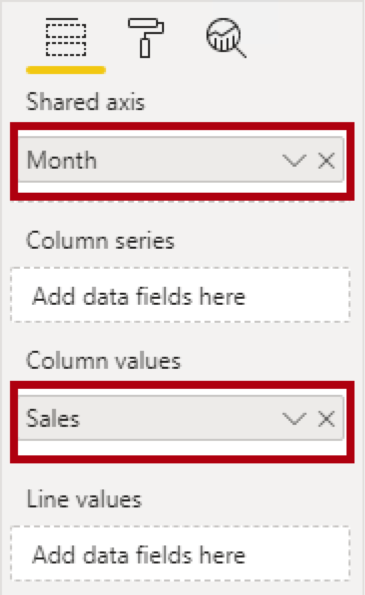
21. □ De-select the slicer by clicking an empty area of the report page.

22. □ To add a chart to the page, in the **Visualizations** pane, click the **Line and Stacked Column Chart** visual type.



23. □ Resize and position the visual so it sits to the right of the logo, and so it fills the width of the report page.

24. □ Drag and drop the following fields into the visual:
    - Date | Month
    - Sales | Sales
25. □ In the visual fields pane (not the **Fields** pane—the visual fields pane is located beneath the **Visualizations** pane), notice that the fields are assigned to the **Shared Axis** and **Column Values** wells/areas.



Shared axis

Month

Column series

Add data fields here

Column values

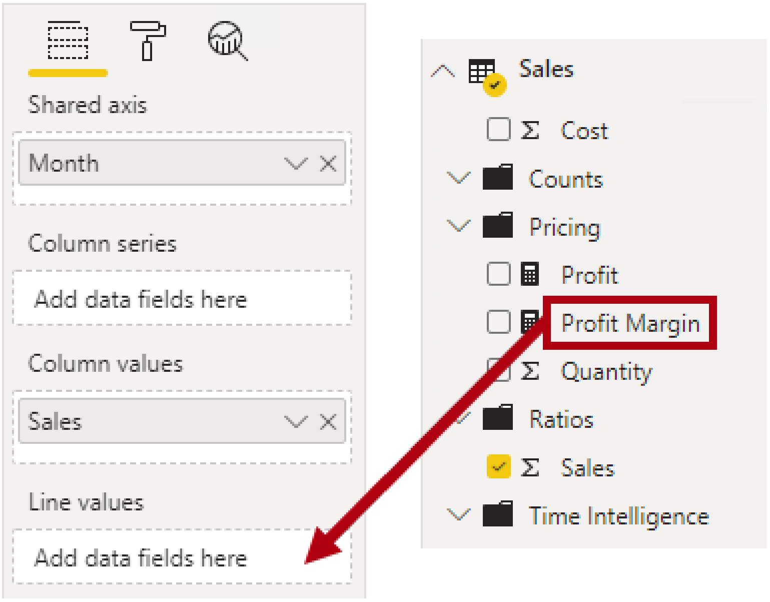
Sales

Line values

Add data fields here

By dragging fields into a visual, they will be added to default wells/areas. For precision, you can drag fields directly into the wells/areas, as you will do next.

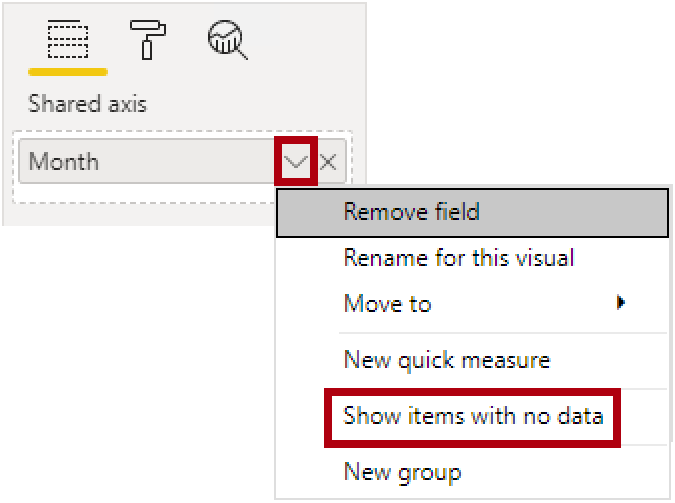
26. □ From the **Fields** pane, drag the **Sales | Profit Margin** field into the **Line Values** well/area.



27. □ Notice that the visual has 11 months only.

The last month of the year, 2020 June, does not have any sales (yet). By default, the visual has eliminated months with BLANK sales. You will now configure the visual to show all months.

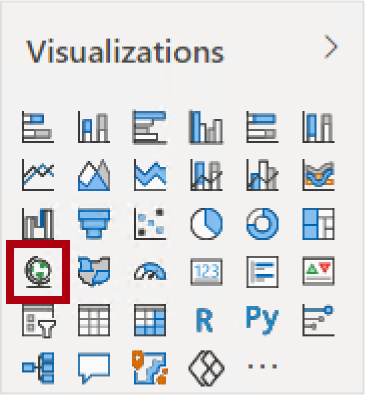
28. □ In the visual fields pane, in the **Shared Axis** well/area, for the **Month** field, click the down-arrow, and then select **Show Items With No Data**.



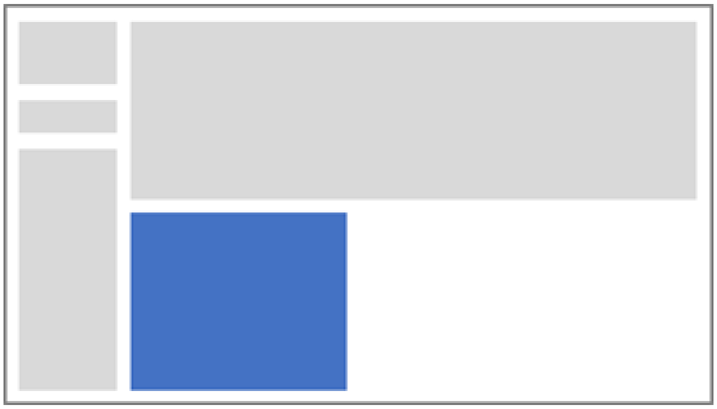
29. □ Notice that the month **2020 June** now appears.

30. □ De-select the chart by clicking an empty area of the report page.

31. □ To add a chart to the page, in the **Visualizations** pane, click the **Map** visual type.



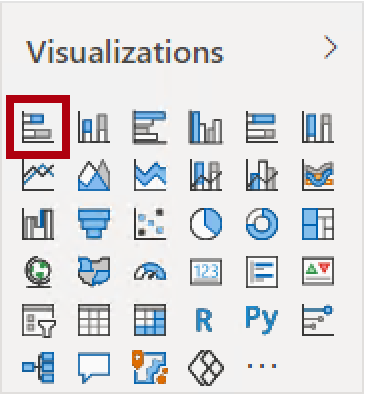
32. □ Resize and position the visual so it sits beneath the column/line chart, and so it fills half the width of the chart above.



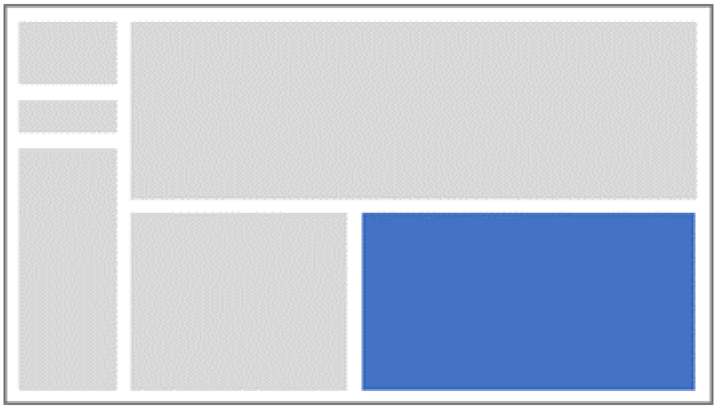
33. □ Add the following fields to the visual wells/areas:
    - Location: **Region | Country**
    - Legend: **Product | Category**
    - Size: **Sales | Sales**

34. □ De-select the chart by clicking an empty area of the report page.

35. □ To add a chart to the page, in the **Visualizations** pane, click the **Stacked Bar Chart** visual type.

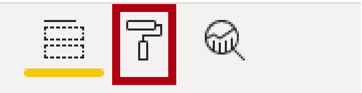


36. □ Resize and position the visual so it fills the remaining report page space.



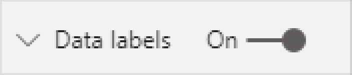
37. □ Add the following fields to the visual wells/areas:
    - Axis: **Product | Category**
    - Value: **Sales | Quantity**

38. □ To format the visual, open the **Format** pane.



39. □ Expand the **Data Colors** group, and then set the **Default Color** property to a suitable color (to complement the column/line chart).

40. □ Set the **Data Labels** property to **On**.



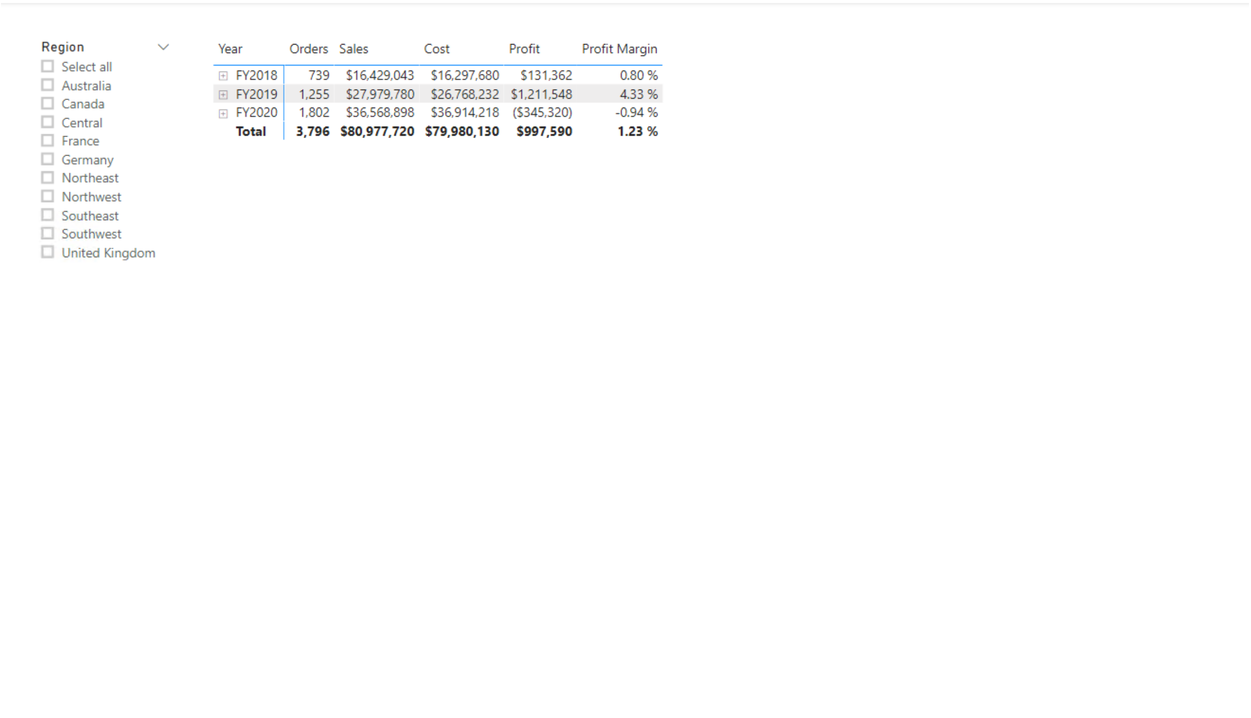
Data labels On

41. □ Save the Power BI Desktop file.

The design of the first page is now complete.

Task 4: Design page 2

In this task you will design the second report page. When you've completed the design, the page will look like the following:



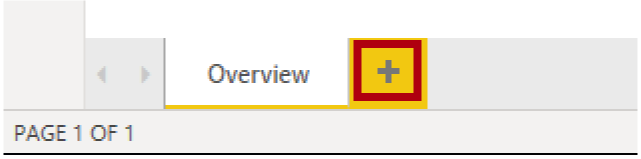
Region

- ☐ Select all
- ☐ Australia
- ☐ Canada
- ☐ Central
- ☐ France
- ☐ Germany
- ☐ Northeast
- ☐ Northwest
- ☐ Southeast
- ☐ Southwest
- ☐ United Kingdom

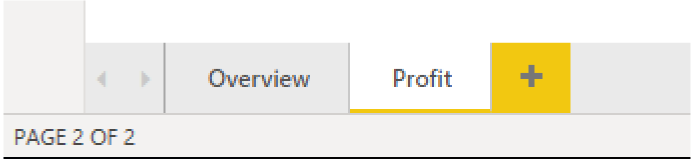
Year	Orders	Sales	Cost	Profit	Profit Margin
⊞ FY2018	739	$16,429,043	$16,297,680	$131,362	0.80 %
⊞ FY2019	1,255	$27,979,780	$26,768,232	$1,211,548	4.33 %
⊞ FY2020	1,802	$36,568,898	$36,914,218	($345,320)	-0.94 %
Total	**3,796**	**$80,977,720**	**$79,980,130**	**$997,590**	**1.23 %**

Important: When detailed instructions have already been provided in the labs, the lab steps will provide more concise instructions. If you need the detailed instructions, you can refer back to other tasks in this lab.

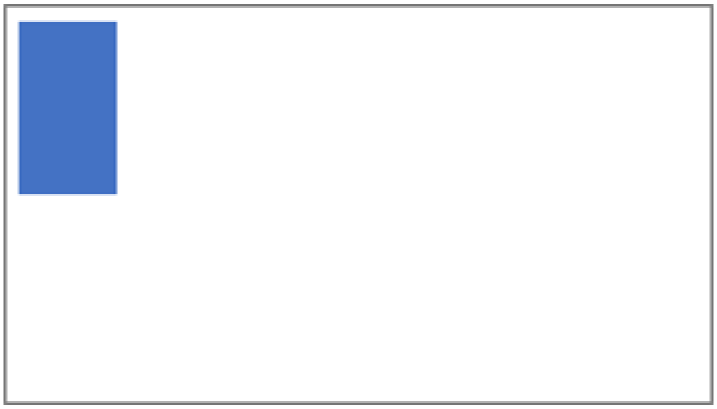
1. ☐ To create a new page, at the bottom-left, click the plus icon.



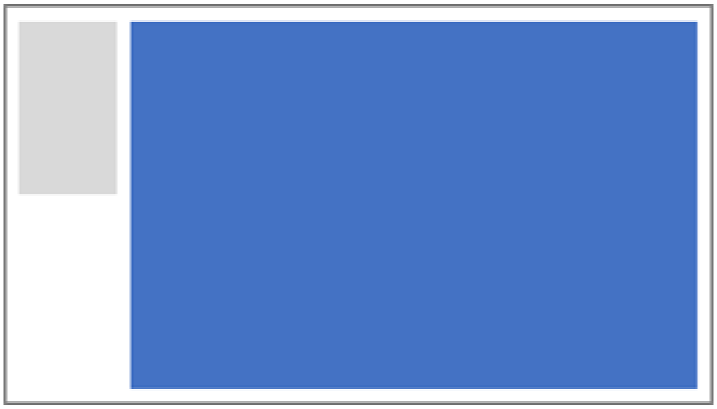
2. ☐ Rename the page to **Profit**.



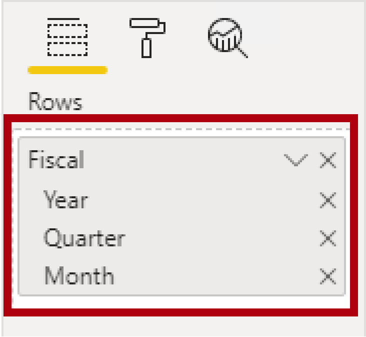
1. □ Add a slicer based on the **Region | Region** field.
2. □ Use the **Format** pane to enable the “Select All” option (in the **Selection Controls** group).
3. □ Resize and position the slicer so it sits at the left side of the report page, and so it is about half the page height.



4. □ Add a matrix visual, and resize and position it so it fills the remaining space of the report page



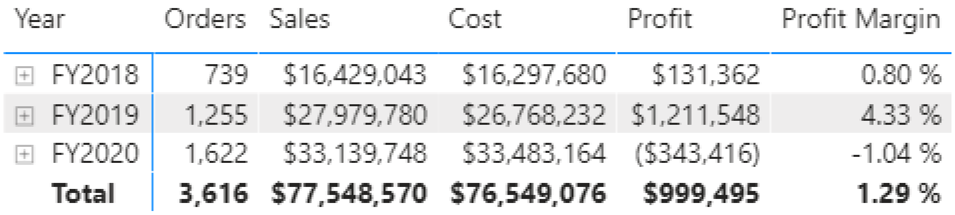
5. □ Add the **Date | Fiscal** hierarchy to the matrix **Rows** well/area.



Rows

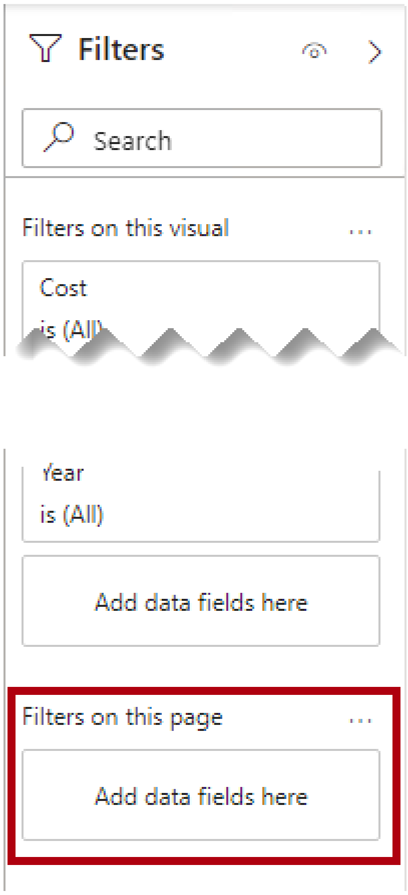
Fiscal
- Year
- Quarter
- Month

6. □ Add the following five **Sales** table fields to the **Values** well/area:
   - ○ Orders (from the **Counts** folder)
   - ○ Sales
   - ○ Cost
   - ○ Profit
   - ○ Profit Margin

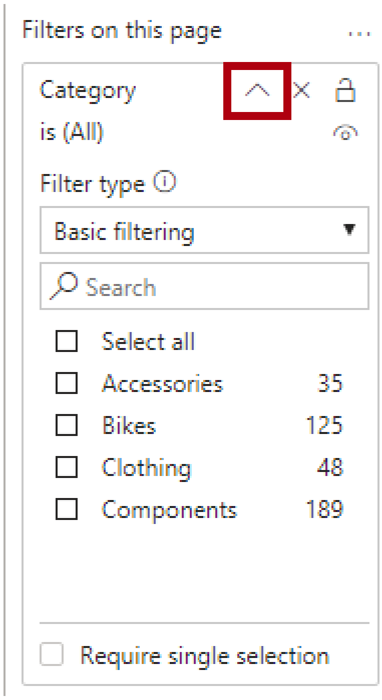


Year	Orders	Sales	Cost	Profit	Profit Margin
⊞ FY2018	739	$16,429,043	$16,297,680	$131,362	0.80 %
⊞ FY2019	1,255	$27,979,780	$26,768,232	$1,211,548	4.33 %
⊞ FY2020	1,622	$33,139,748	$33,483,164	($343,416)	-1.04 %
Total	**3,616**	**$77,548,570**	**$76,549,076**	**$999,495**	**1.29 %**

7. □ In the **Filters** pane (located at the left of the **Visualizations** pane), notice the **Filter On This Page** well/area (you may need to scroll down).



8. □ From the **Fields** pane, drag the **Product | Category** field into the **Filter On This Page** well/area.
9. □ Inside the filter card, at the top-right, click the arrow to collapse the card.



Filters on this page ...

Category

is (All)

Filter type

Basic filtering ▾

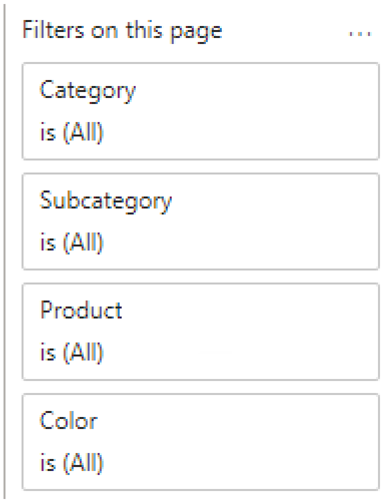
Search

□ Select all	
□ Accessories	35
□ Bikes	125
□ Clothing	48
□ Components	189

□ Require single selection

Fields added to the ***Filters*** *pane can achieve the same result as a slicer. One difference is they don't take up space on the report page. Another difference is that they can be configured to achieve more sophisticated filtering requirements.*

10. □ Add each of the following **Product** table fields to the **Filter On This Page** well/area, collapsing each, directly beneath the **Category** card:
    - Subcategory
    - Product
    - Color



Filters on this page ...

Category
is (All)

Subcategory
is (All)

Product
is (All)

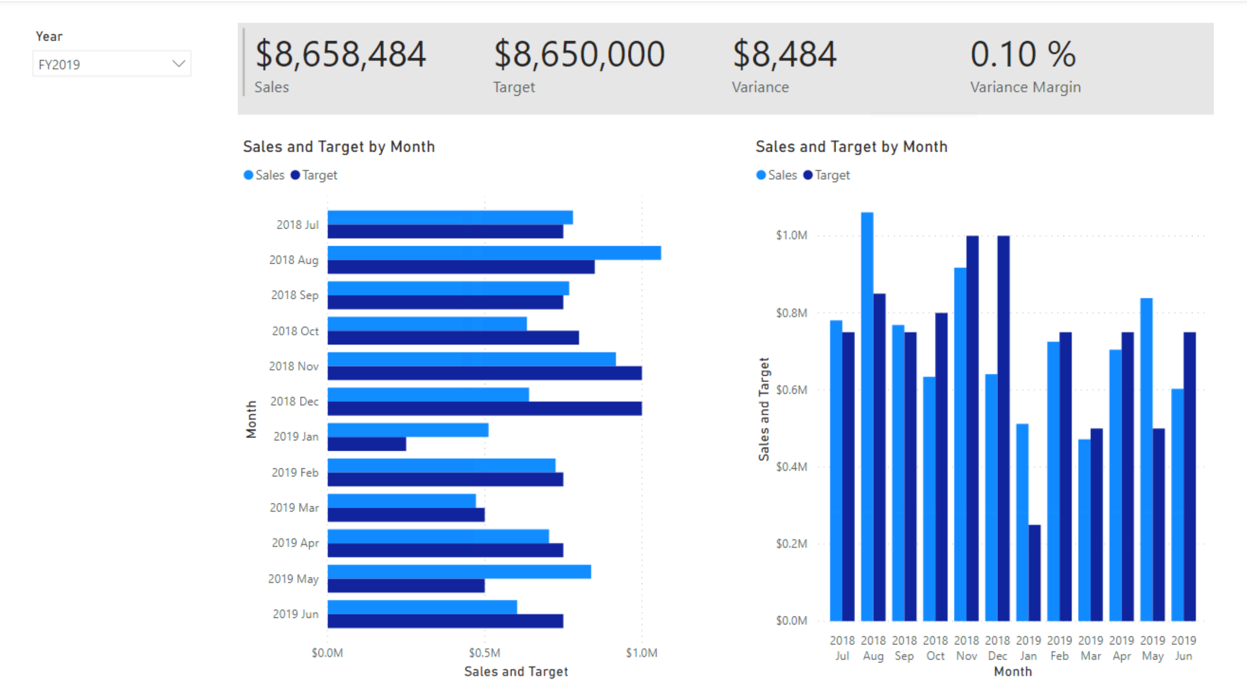
Color
is (All)

11. □ Save the Power BI Desktop file.

The design of the second page is now complete.

Task 5: Design page 3

In this task you will design the third—and final—report page. When you've completed the design, the page will look like the following:



Year: FY2019

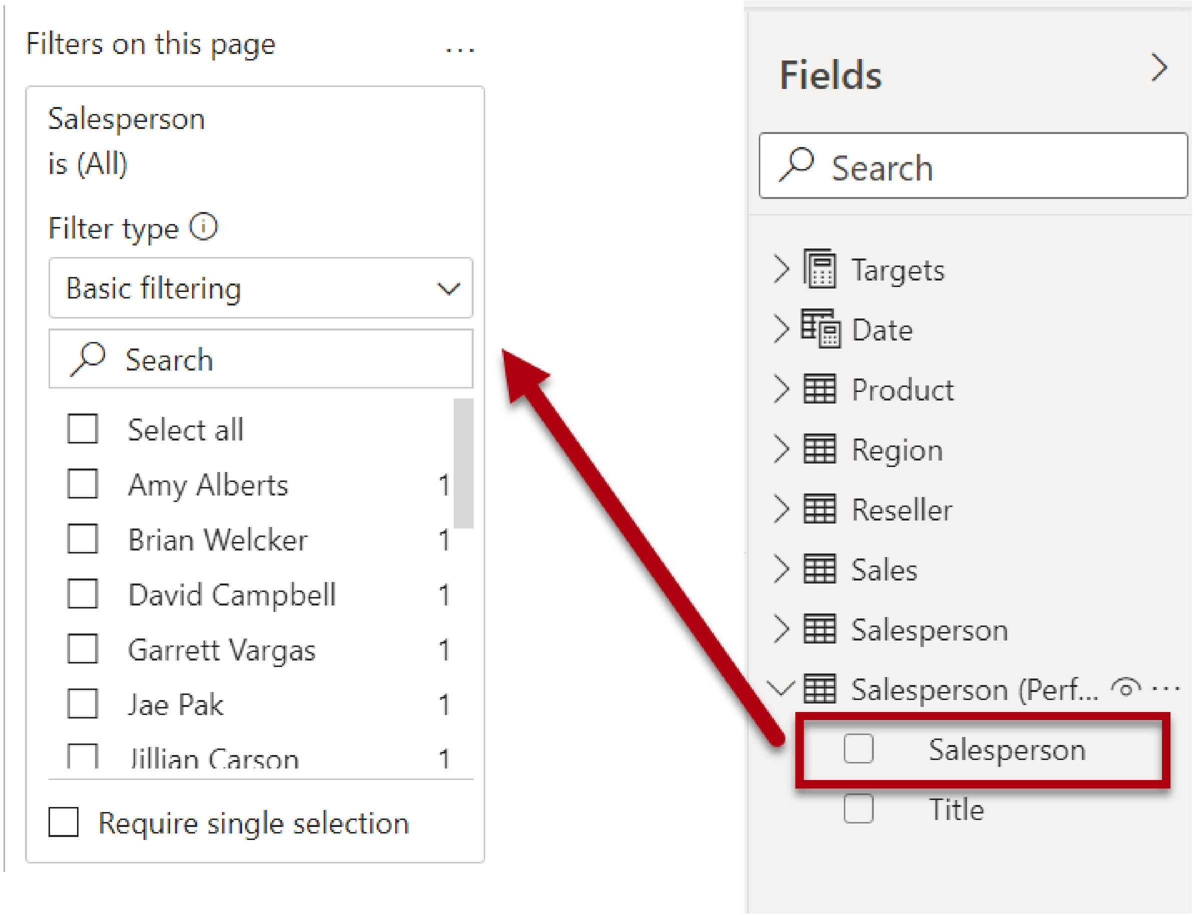
$8,658,484 Sales

$8,650,000 Target

$8,484 Variance

0.10 % Variance Margin

1. □ Create a new page, and then rename it as **My Performance**.
2. □ To simulate the performance of row-level security filters, drag the **Salesperson (Performance) | Salesperson** field to the page level filters in the filter pane.



Filters on this page ...

Salesperson
is (All)

Filter type ⓘ

Basic filtering

Search

- □ Select all
- □ Amy Alberts 1
- □ Brian Welcker 1
- □ David Campbell 1
- □ Garrett Vargas 1
- □ Jae Pak 1
- □ Jillian Carson 1

□ Require single selection

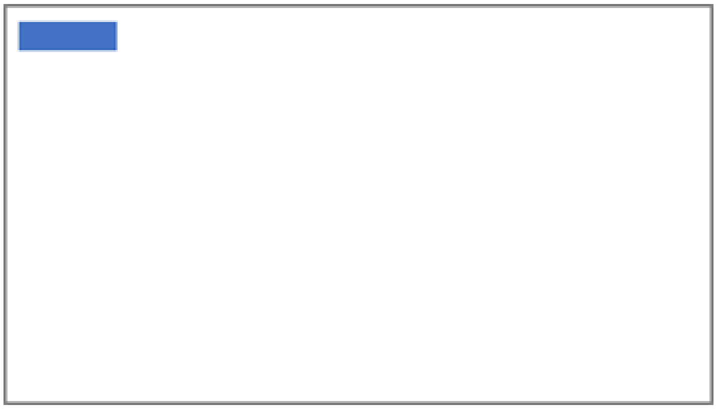
Fields

Search

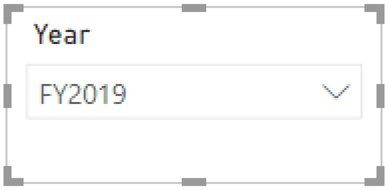
- Targets
- Date
- Product
- Region
- Reseller
- Sales
- Salesperson
- Salesperson (Perf... ⋯
  - □ Salesperson
  - □ Title

3. □ Select **Michael Blythe**. Data on the **My Performance** report page will now be filtered to display data for Michael Blythe only.

4. □ Add a dropdown slicer based on the **Date | Year** field, and then resize and position it so it sits at the top-left corner of the page.



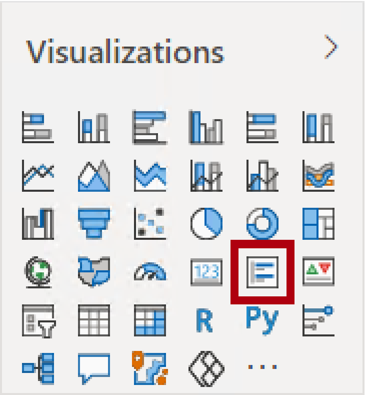
5. □ In the slicer, set the page to filter by **FY2019**.

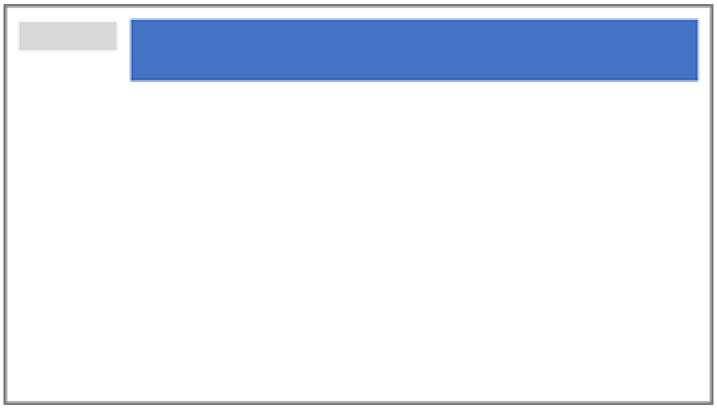


Year

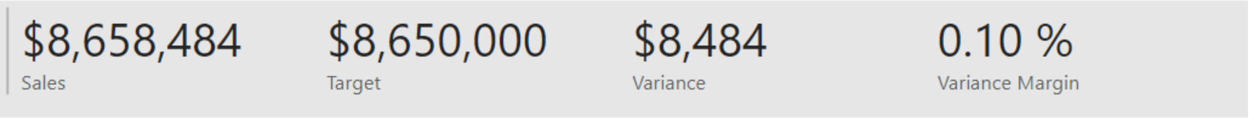
FY2019

6. ☐ Add a **Multi-row Card** visual, and then resize and reposition it so it sits to the right of the slicer and fills the remaining width of the page.





7. ☐ Add the following four fields to the visual:
   - Sales | Sales
   - Targets | Target
   - Targets | Variance
   - Targets | Variance Margin
8. ☐ Format the visual:
   - In the **Data Labels** group, increase the **Text Size** property to **28pt**
   - In the **Background** group, set the **Color** to a light gray color



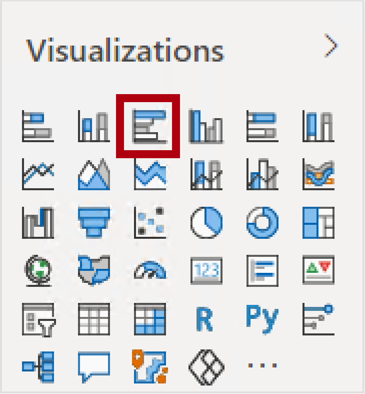
$8,658,484 Sales

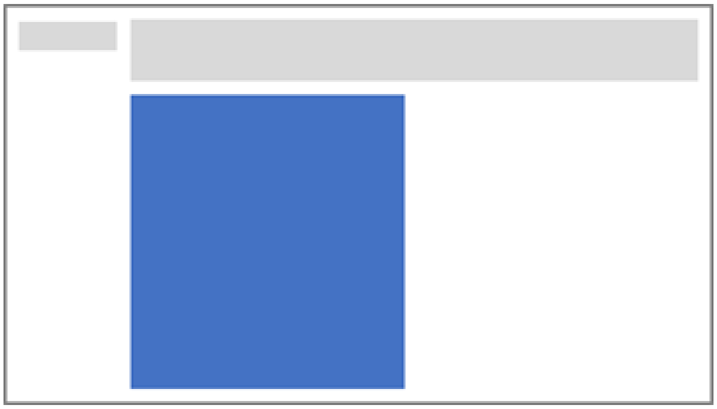
$8,650,000 Target

$8,484 Variance

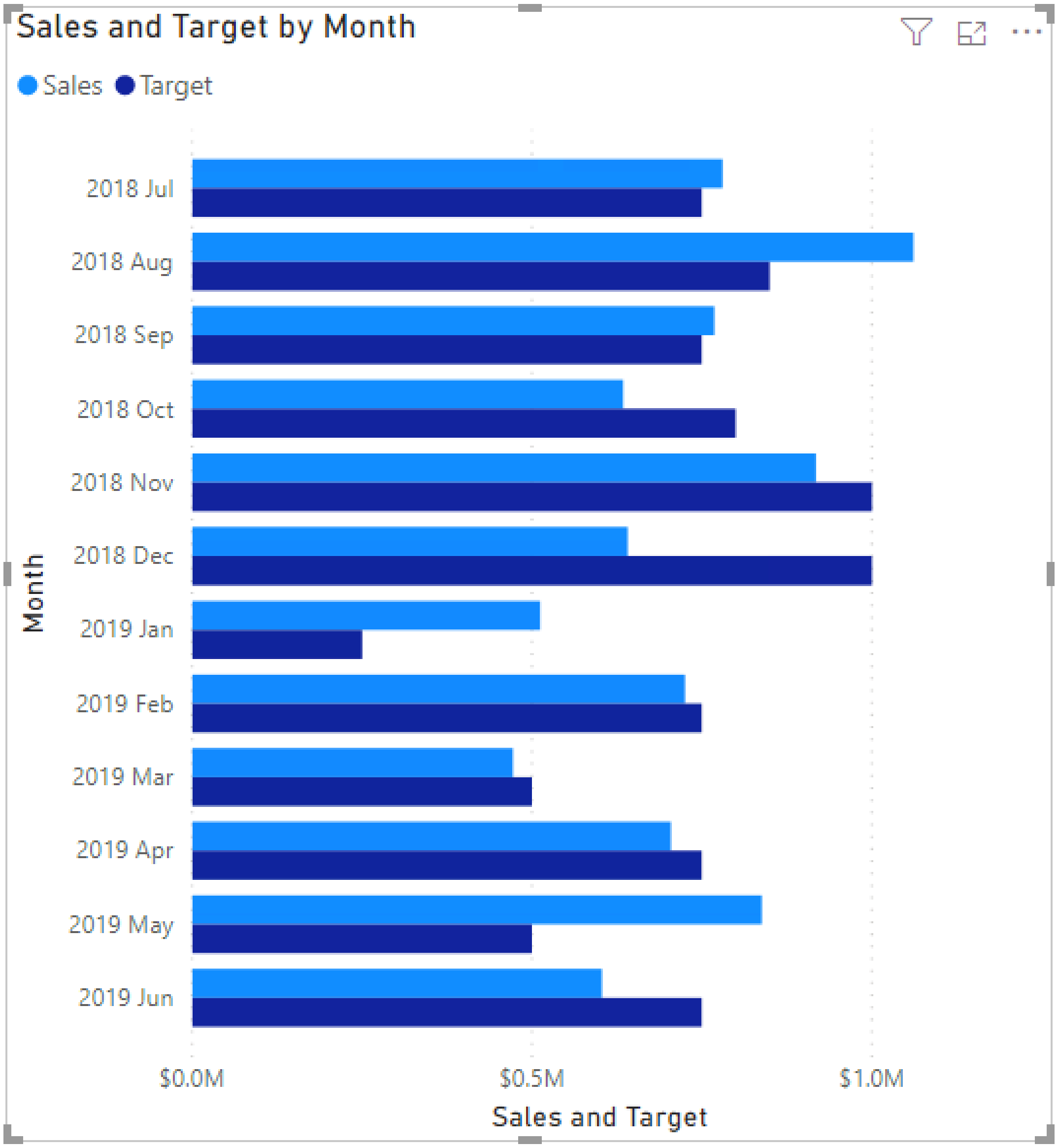
0.10 % Variance Margin

9. □ Add a **Clustered Bar Chart** visual, and then resize and position it so it sits beneath the multi-row card visual and fills the remaining height of the page, and half the width of the multi-row card visual.



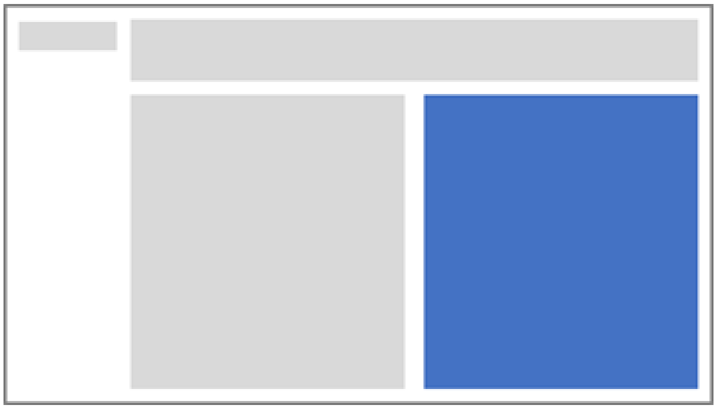


10. □ Add the following fields to the visual wells/areas:
    - Axis: **Date | Month**
    - Value: **Sales | Sales** and **Targets | Target**

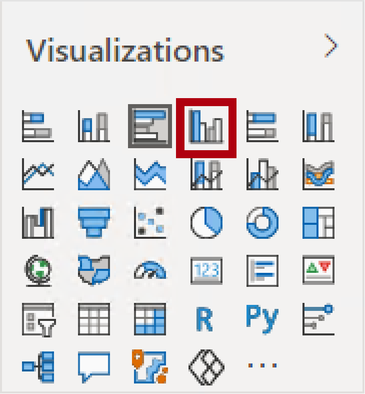


11. □ To create a copy of the visual, press **Ctrl+C**, and then press **Ctrl+V**.

12. □ Position the new visual to the right of the original visual.



13. □ To modify the visualization type, in the **Visualizations** pane, select **Clustered Column Chart**.



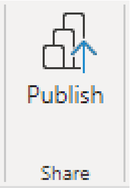
It's now possible to see the same data expressed by two different visualization types. This isn't a good use of the page layout, however, you'll improve it in the ***Design a Report in Power BI Desktop, Part 2*** *lab by superimposing the visuals. By adding buttons to the page, you'll allow the report user to determine which of the two visuals is visible.*

The design of the third—and final—page is now complete.

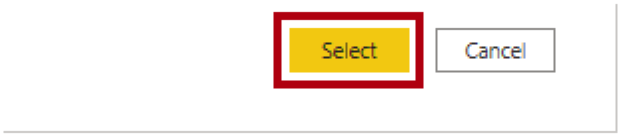
Task 6: Publish the report

In this task you will publish the report.

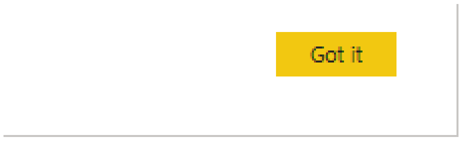
1. □ Select the **Overview** page.
2. □ Save the Power BI Desktop file.
3. □ On the **Home** ribbon tab, from inside the **Share** group, click **Publish**.



4. □ In the **Publish to Power** BI window, notice that **My Workspace** is selected.
5. □ To publish the report, click **Select**.



6. □ When the publication has succeeded, click **Got It**.



7. □ Leave Power BI Desktop open.

You'll explore the report in the Power BI service in the next exercise.

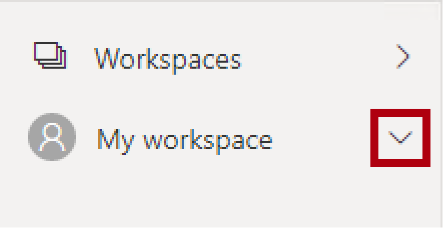
Exercise 2: Explore the Report

In this exercise you will explore the report that was published to Power BI.

Task 1: Explore the report

In this task you will explore the report that was published to Power BI.

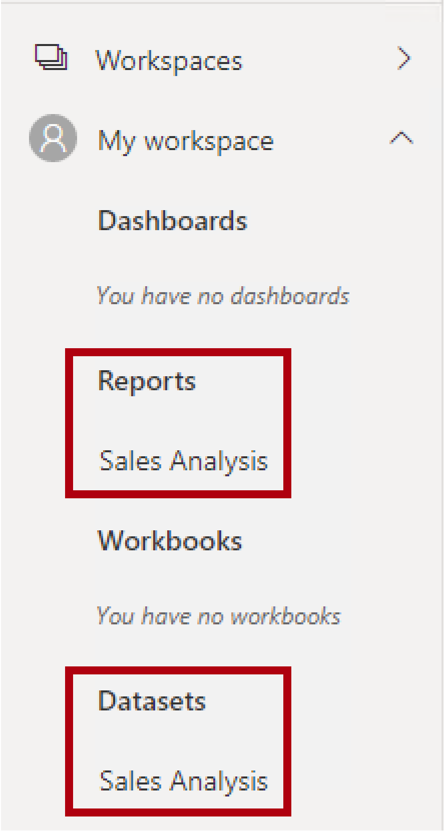
1. □ In the Microsoft Edge browser window, in the Power BI service, in the **Navigation** pane (located at the left, and it could be collapsed), expand **My Workspace**.



2. □ Review the contents of the workspace, noticing the **Sales Analysis** report and dataset.

When you published the Power BI Desktop file, the data model was published as a dataset.

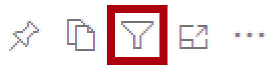
If you don't see it, press **F5** *to reload the browser, and then expand the workspace again.*



3. □ To open the report, click the **Sales Analysis** report.
4. □ At the left, in the **Pages** pane, select the **Overview** page.
5. □ In the **Regions** slicer, while pressing the **Ctrl** key, select multiple regions.
6. □ In the column/line chart, select any month column to cross filter the page.
7. □ While pressing the **Ctrl** key, select an additional month.

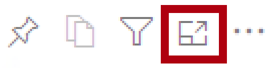
By default, cross filtering filters all other visuals on the page.

8. □ Notice that the bar chart is filtered and highlighted, with the bold portion of the bars representing the filtered months.
9. □ Hover the cursor over the bar chart visual, and then at the top-right, hover the cursor over the filter icon.



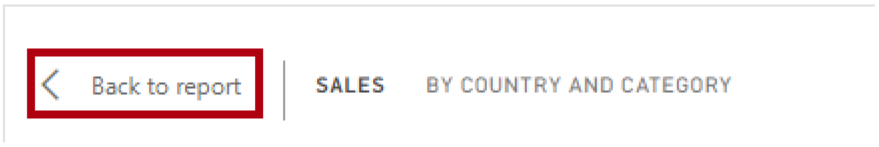
The filter icon allows you to understand all filters that are applied to the visual, including slicers and cross filters from other visual.

10. □ Hover the cursor over a bar, and then notice the tooltip information.
11. □ To undo the cross filter, in the column/line chart, click an empty area of the visual.
12. □ Hover the cursor over the map visual, and then at the top-right, click the **Focus mode** icon.

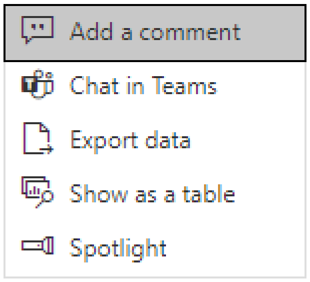


Focus mode zooms the visual to full page size.

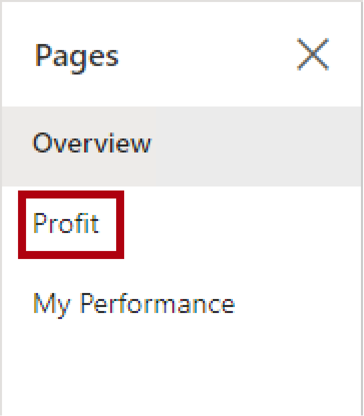
13. □ Hover the cursor over different segments of the pie charts to reveal tooltips.
14. □ To return to the report page, at the top-left, click **Back to Report**.



15. □ Hover the cursor over the map visual again, and then at the top-right, click the ellipsis (…), and then notice the menu options.



16. □ Try out each of the options, except **Chat in Teams**.
17. □ At the left, in the **Pages** pane, select the **Profit** page.



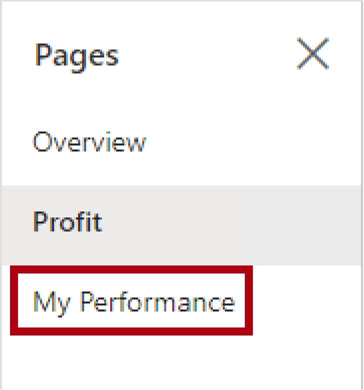
18. □ Notice that the **Region** slicer has a different selection to the **Region** slicer on the **Overview** page.

    *The slicers are not synchronized. You'll modify the report design to ensure they sync between pages in the **Design a Report in Power BI Desktop, Part 2** lab.*

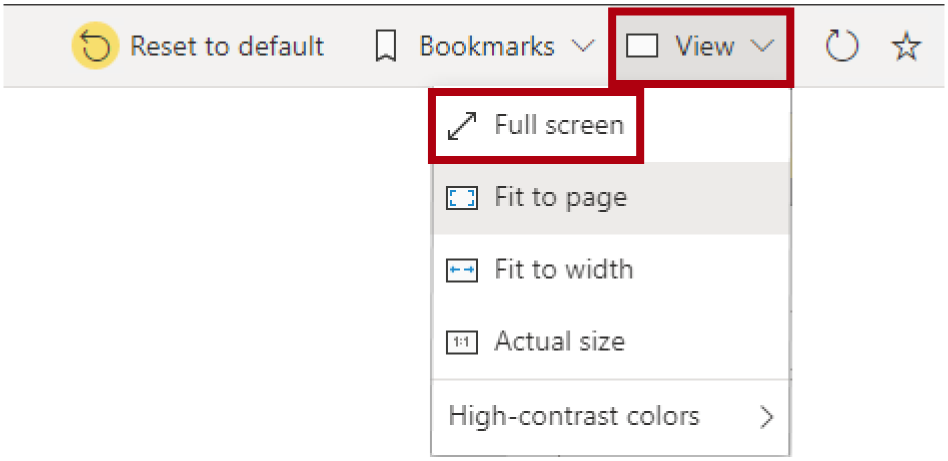
19. □ In the **Filters** pane (located at the right), expand a filter card, and apply some filters.

    *The **Filters** pane allows you to define more filters than could possibly fit on a page as slicers.*

20. □ In the matrix visual, use the plus (+) button to drill into the **Fiscal** hierarchy.
21. □ Select the **My Performance** page.



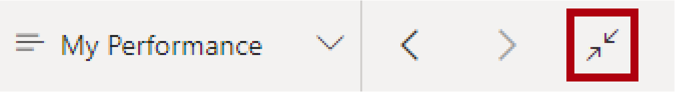
22. □ At the top-right on the menu bar, click **View**, and then select **Full Screen**.



23. □ Interact with the page by modifying the slicer, and cross filtering the page.

24. □ At the bottom of the window, notice the commands to change page, navigate backwards or forwards between pages, or to exit full screen mode.

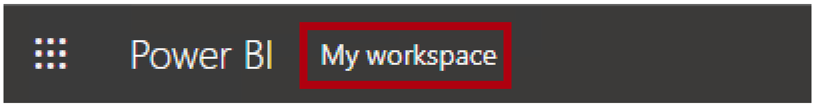
25. □ Click the left icon to exit full screen mode.



Task 2: Finish up

In this task you will complete the lab.

1. □ To return to your workspace, in the banner across the window web page, click **My Workspace**.



2. □ Leave the Microsoft Edge browser window open.

You will enhance the report design with advanced features in the ***Design a Report in Power BI Desktop, Part 2*** *lab.*

Congratulations

Design a Report in Power BI Desktop, Part 2

In this lab you will enhance the **Sales Analysis** with advanced design features.

In this lab you learn how to:

- Sync slicers
- Create a drillthrough page
- Apply conditional formatting
- Create and use bookmarks

Exercise 1: Configure Sync Slicers

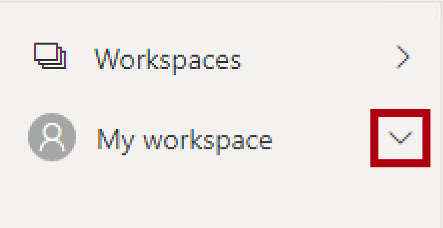
In this exercise you will sync the report page slicers.

Task 1: Get started – Sign in

In this task you will setup the environment for the lab by signing in to Power BI.

Important: If you have already signed in to Power BI, continue from the next task.

1. □ In the Microsoft Edge browser window, in the Power BI service, in the **Navigation** pane, expand **My Workspace**.



Task 2: Get started – Open report

In this task you will setup the environment for the lab by opening the starter report.

Important: If you are continuing on from the previous lab (and you completed that lab successfully), do not complete this task; instead, continue from the next task.

1. □ In the **Open** window, navigate to the **D:\DA100\Labs\08-design-report-in-power-bi-desktop-enhanced\Starter** folder.

Task 3: Sync slicers

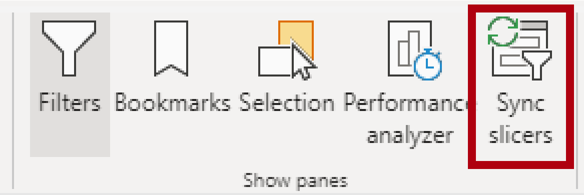
In this task you will sync the **Year** and **Region** slicers.

You will continue the development of the report created in the ***Design a Report in Power BI Desktop, Part 1*** *lab.*

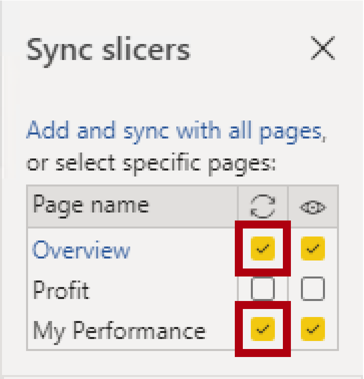
1. □ In Power BI Desktop, on the **Overview** page, set the **Year** slicer to **FY2018**.
2. □ Go to the **My Performance** page, and then notice that the **Year** slicer is a different value.

   When slicers aren't synced it can contribute to misrepresentation of data and frustration for report users. You'll now sync the report slicers.

3. □ Return to the **Overview** page, and then select the **Year** slicer.
4. □ On the **View** ribbon tab, from inside the **Show Panes** group, click **Sync Slicers**.



5. □ In the **Sync Slicers** pane (at the left of the **Visualizations** pane), in the second column (which represents syncing), check the checkboxes for the **Overview** and **My Performance** pages.

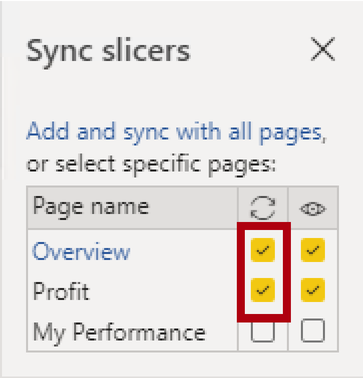


Sync slicers

Add and sync with all pages, or select specific pages:

Page name	⟳	👁
Overview	☑	☑
Profit	☐	☐
My Performance	☑	☑

6. □ On the **Overview** page, select the **Region** slicer.
7. □ Sync the slicer with the **Overview** and **Profit** pages.

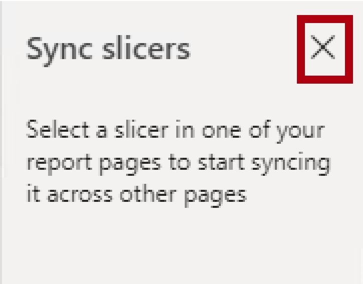


Sync slicers

Add and sync with all pages, or select specific pages:

Page name	⟳	👁
Overview	☑	☑
Profit	☑	☑
My Performance	□	□

8. □ Test the sync slicers by selecting different filter options, and then verifying that the synced slicers filter by the same selection.
9. □ To close the **Sync Slicer** page, click the **X** located at the top-right of the pane.

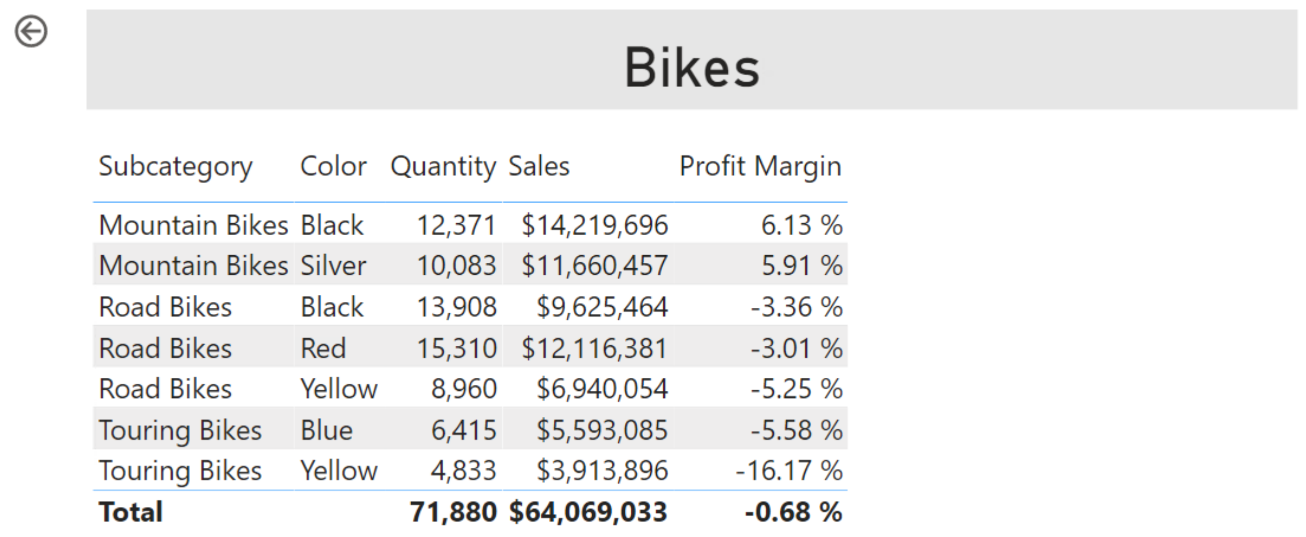


Sync slicers

Select a slicer in one of your report pages to start syncing it across other pages

Exercise 2: Configure Drill Through

In this exercise you will create a new page and configure it as a drill through page. When you've completed the design, the page will look like the following:



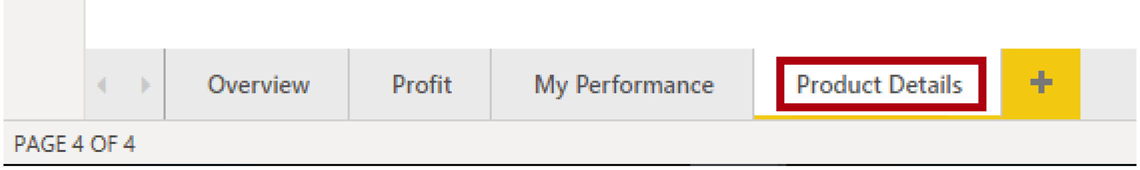
Bikes

Subcategory	Color	Quantity	Sales	Profit Margin
Mountain Bikes	Black	12,371	$14,219,696	6.13 %
Mountain Bikes	Silver	10,083	$11,660,457	5.91 %
Road Bikes	Black	13,908	$9,625,464	-3.36 %
Road Bikes	Red	15,310	$12,116,381	-3.01 %
Road Bikes	Yellow	8,960	$6,940,054	-5.25 %
Touring Bikes	Blue	6,415	$5,593,085	-5.58 %
Touring Bikes	Yellow	4,833	$3,913,896	-16.17 %
Total		**71,880**	**$64,069,033**	**-0.68 %**

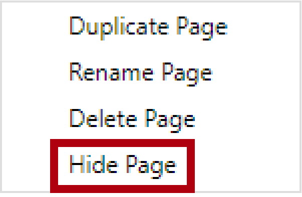
Task 1: Create a drill through page

In this task you will create a new page and configure it as a drill through page.

1. □ Add a new report page named **Product Details**.



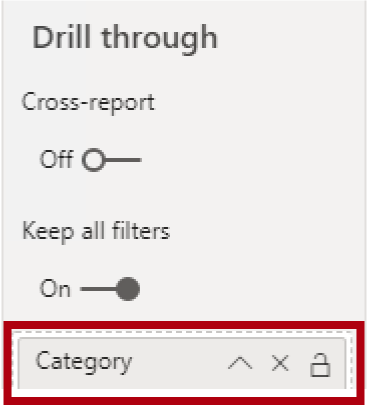
2. □ Right-click the **Product Details** page tab, and then select **Hide Page**.



Report users won't be able to go to the drill through page directly. They'll need to access it from visuals on other pages. You'll learn how to drill through to the page in the final exercise of this lab.

3. □ Beneath the **Visualizations** pane, in the **Drill Through** section, add the **Product | Category** field to the **Add Drill-Through Fields Here** box.

*The labs use a shorthand notation to reference a field. It will look like this: **Product | Category**. In this example, **Product** is the table name and **Category** is the field name.*



Drill through

Cross-report

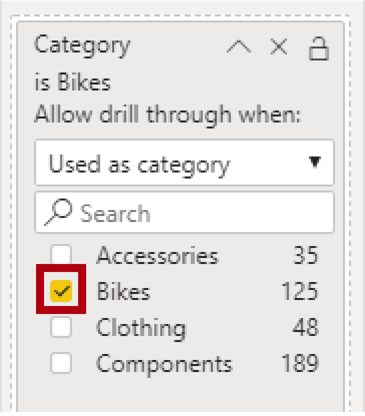
Off ○—

Keep all filters

On —●

Category

4. □ To test the drill through page, in the drill through filter card, select **Bikes**.



Category

is Bikes

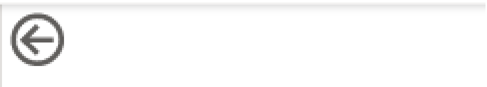
Allow drill through when:

Used as category ▾

Search

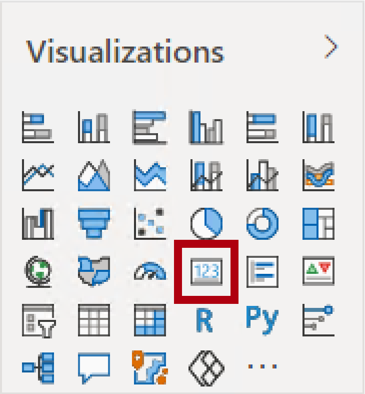
□	Accessories	35
☑	Bikes	125
□	Clothing	48
□	Components	189

5. □ At the top-left of the report page, notice the arrow button.



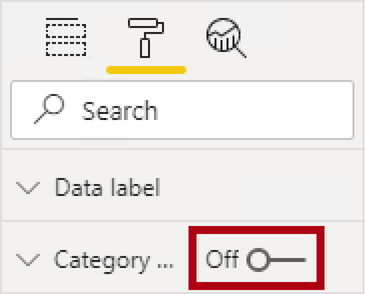
A button is added automatically when a field is added to the drill through well/area. It allows report users to navigate back to the page from which they drilled through.

6. □ Add a **Card** visual to the page, and then resize and position it so it sits to the right of the button and fills the remaining width of the page.



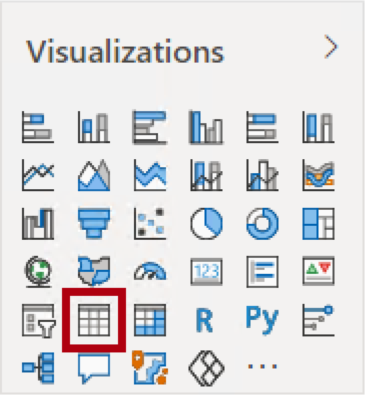
7. □ Drag the **Product | Category** field into the card visual.

8. □ Configure the format options for the visual, and then turn the **Category Label** property to **Off**.



9. □ Set the **Background Color** property to a light shade of gray.

10. □ Add a **Table** visual to the page, and then resize and position it so it sits beneath the card visual and fills the remaining space on the page.

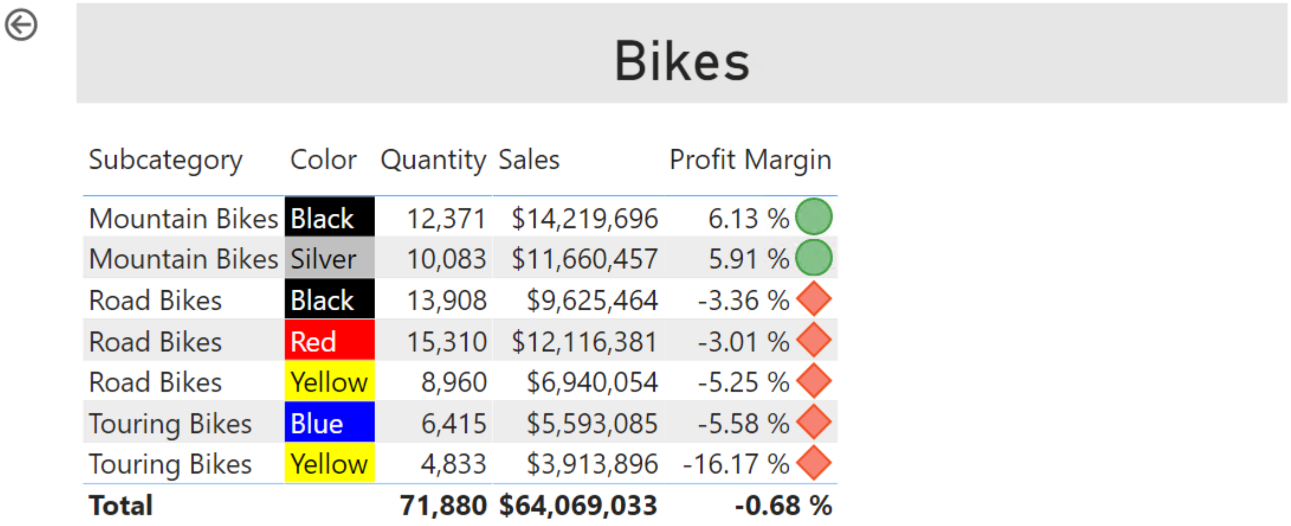


11. □ Add the following fields to the visual:
    - ○ Product | Subcategory
    - ○ Product | Color
    - ○ Sales | Quantity
    - ○ Sales | Sales
    - ○ Sales | Profit Margin
12. □ Configure the format options for the visual, and in the **Grid** section, set the **Text Size** property to **20pt**.

    The design of the drill through page is almost complete. You'll enhance the page with conditional formatting in the next exercise.

Exercise 3: Add Conditional Formatting

In this exercise you will enhance the drill through page with conditional formatting. When you've completed the design, the page will look like the following:



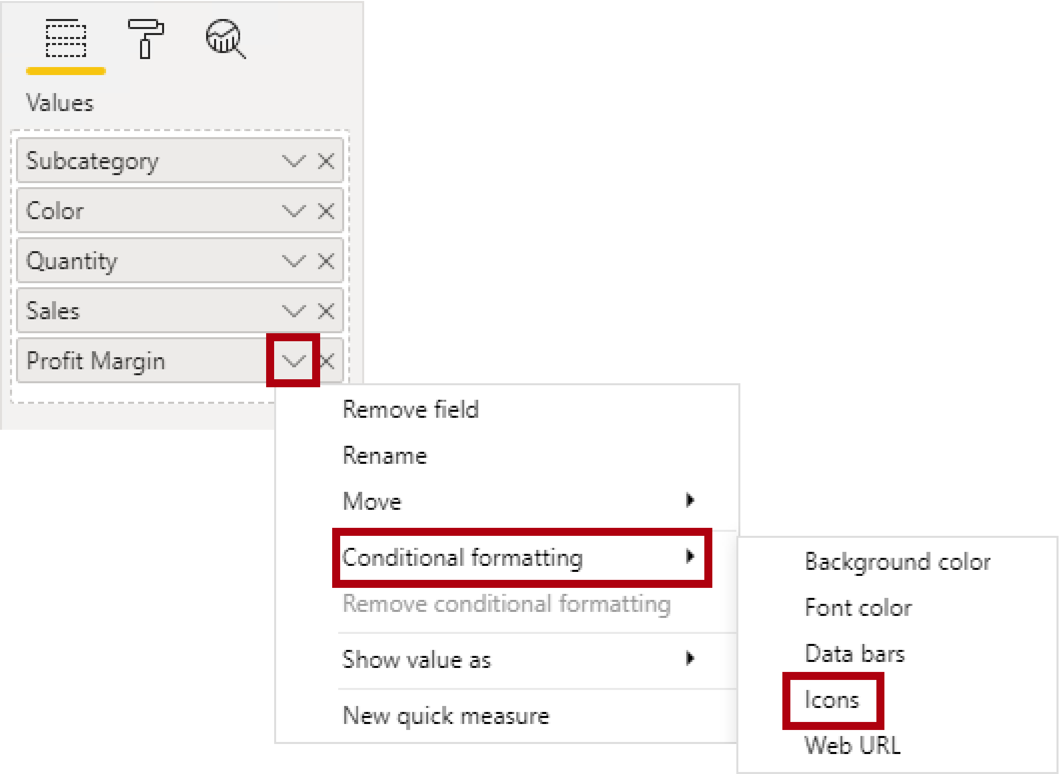
Bikes

Subcategory	Color	Quantity	Sales	Profit Margin
Mountain Bikes	Black	12,371	$14,219,696	6.13 % ●
Mountain Bikes	Silver	10,083	$11,660,457	5.91 % ●
Road Bikes	Black	13,908	$9,625,464	-3.36 % ◆
Road Bikes	Red	15,310	$12,116,381	-3.01 % ◆
Road Bikes	Yellow	8,960	$6,940,054	-5.25 % ◆
Touring Bikes	Blue	6,415	$5,593,085	-5.58 % ◆
Touring Bikes	Yellow	4,833	$3,913,896	-16.17 % ◆
Total		**71,880**	**$64,069,033**	**-0.68 %**

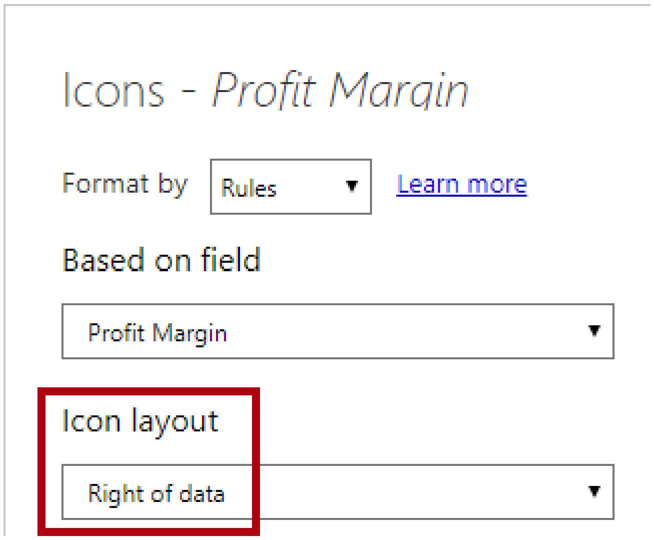
Task 1: Add conditional formatting

In this task you will enhance the drill through page with conditional formatting.

1. □ Select the table visual.
2. □ In the visualization pane, click the down-arrow on the **Profit Margin** value, and then select **Conditional Formatting | Icons**.



3. □ In the **Icons – Profit Margin** window, in the **Icon Layout** dropdown list, select **Right of Data**.



Icons - *Profit Margin*

Format by Rules

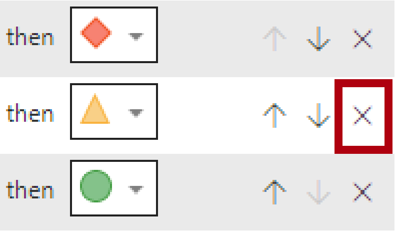
Based on field

Profit Margin

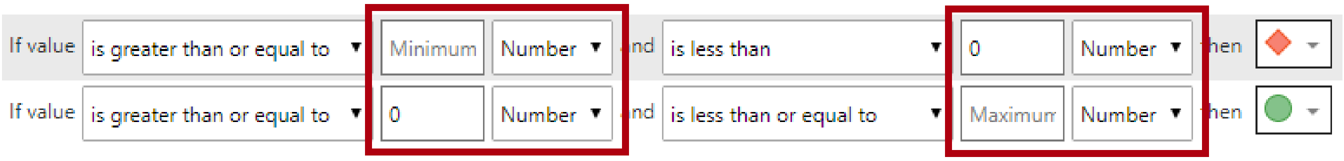
Icon layout

Right of data

4. □ To delete the middle rule, at the left of the yellow triangle, click **X**.



5. □ Configure the first rule (red diamond) as follows:
   - In the second control, remove the value
   - In the third control, select **Number**
   - In the fifth control, enter **0**
   - In the sixth control, select **Number**
6. □ Configure the second rule (green circle) as follows:
   - In the second control, enter **0**
   - In the third control, select **Number**
   - In the fifth control, remove the value
   - In the sixth control, select **Number**

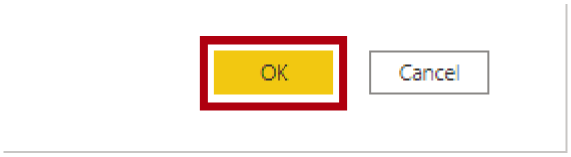


If value | is greater than or equal to ▾ | Minimum | Number ▾ | nd | is less than ▾ | 0 | Number ▾ | hen

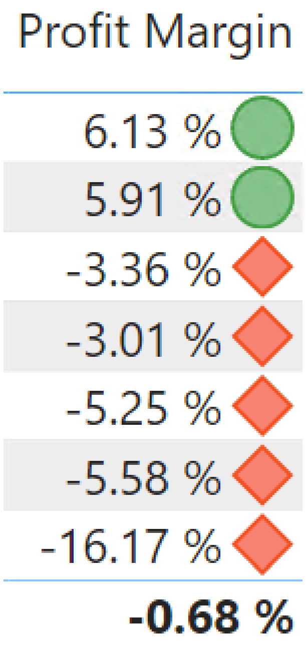
If value | is greater than or equal to ▾ | 0 | Number ▾ | nd | is less than or equal to ▾ | Maximum | Number ▾ | hen

The rules can be interpreted as follows: display a red diamond if the profit margin value is less than 0; otherwise if the value is great or equal to zero, display a green circle.

7. □ Click **OK**.



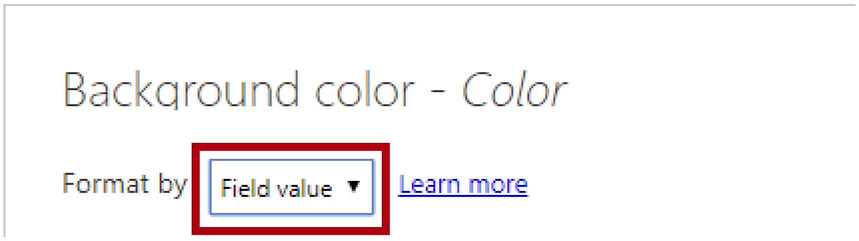
8. □ In the table visual, verify that the that the correct icons are displayed.



Profit Margin
6.13 % ●
5.91 % ●
-3.36 % ◆
-3.01 % ◆
-5.25 % ◆
-5.58 % ◆
-16.17 % ◆
-0.68 %

9. □ Configure background color conditional formatting for the **Color** field.

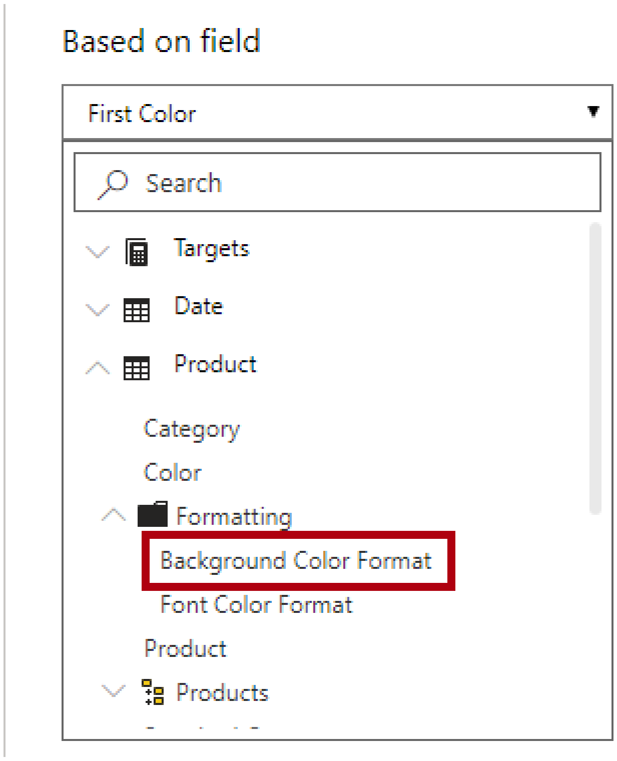
10. □ In the **Background Color – Color** window, in the **Format By** dropdown list, select **Field Value**.



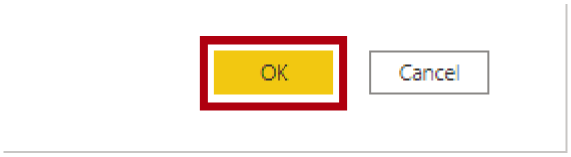
Background color - *Color*

Format by: Field value ▼

11. □ In the **Based on Field** dropdown list, select **Product | Formatting | Background Color Format**.



12. □ Click **OK**.

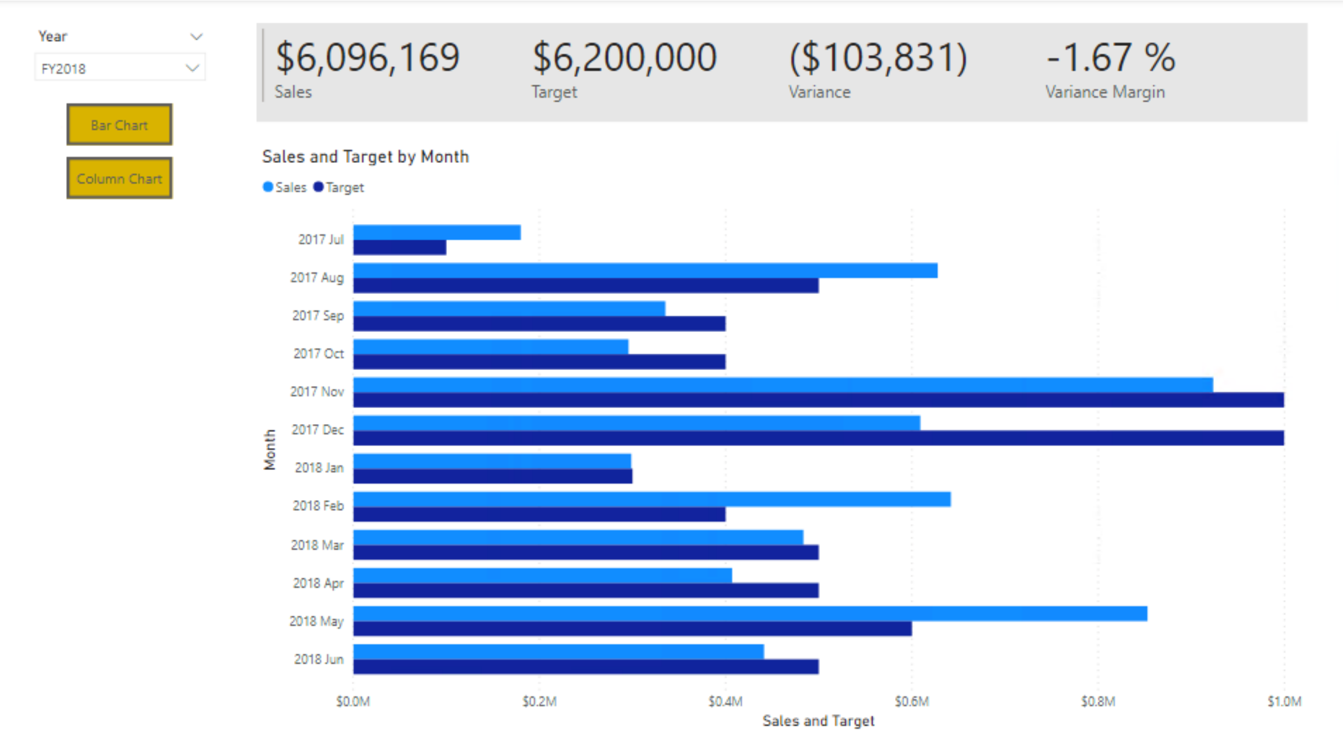


13. □ Repeat the previous steps to configure font color conditional formatting for the **Color** field, using the **Product | Formatting | Font Color Format** field

*You may recall that the background and font colors were source from the **ColorFormats.csv** file in the **Prepare Data in Power BI Desktop** lab, and then integrated with the **Product** query in the **Load Data in Power BI Desktop** lab.*

Exercise 4: Add Bookmarks and Buttons

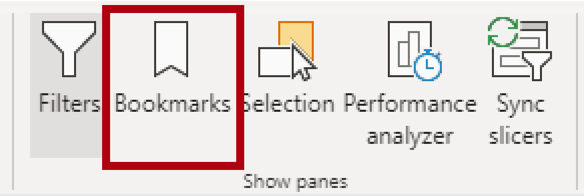
In this exercise you will enhance the **My Performance** page with buttons, allowing the report user to select the visual type to display. When you've completed the design, the page will look like the following:



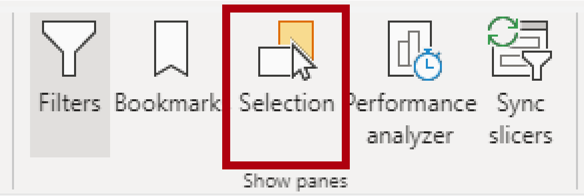
Task 1: Add bookmarks

In this task you will add two bookmarks, one to display each of the monthly sales/targets visuals.

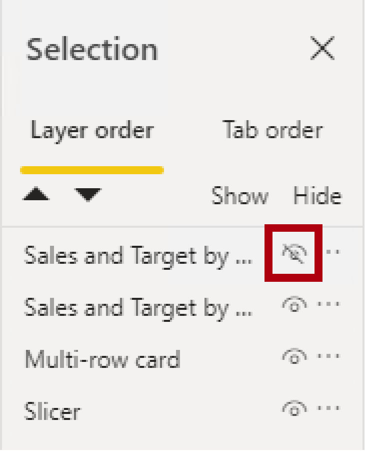
1. □ Go to the **My Performance** page.
2. □ On the **View** ribbon tab, from inside the **Show Panes** group, click **Bookmarks**.



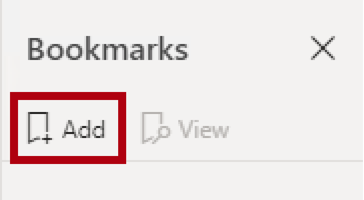
3. □ On the **View** ribbon tab, from inside the **Show Panes** group, click **Selection**.



4. □ In the **Selection** pane, beside one of the **Sales and Target by Month** items, to hide the visual, click the eye icon.



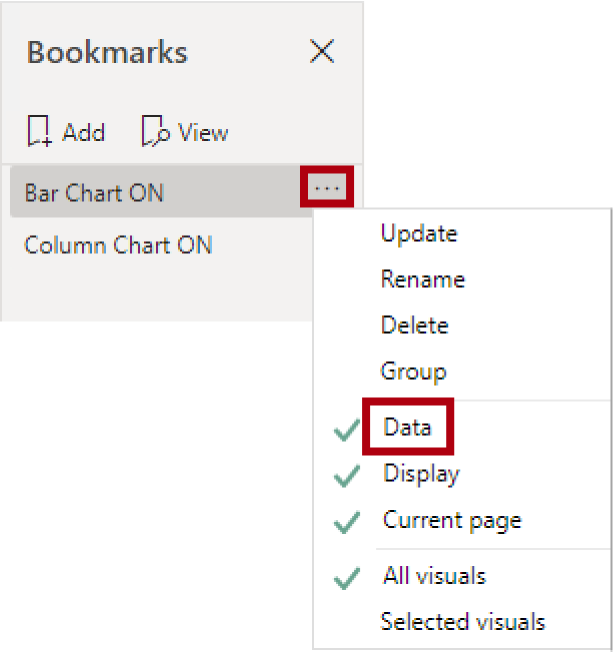
5. □ In the **Bookmarks** pane, click **Add**.



6. □ To rename the bookmark, double-click the bookmark.

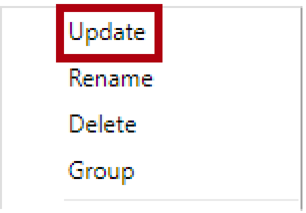
7. □ If the visible chart is the bar chart, rename the bookmark as **Bar Chart ON**, otherwise rename the bookmark as **Column Chart ON**.

8. □ To edit the bookmark, in the **Bookmarks** pane, hover the cursor over the bookmark, click the ellipsis, and then select **Data**.



*Disabling the **Data** option means the bookmark won't use the current filter state. That's important because otherwise the bookmark would permanently lock in the filter currently applied by the **Year** slicer.*

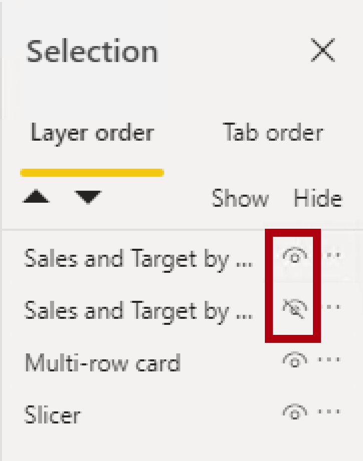
9. □ To update the bookmark, click the ellipsis again, and then select **Update**.



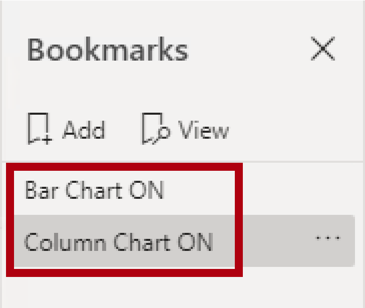
In the following steps, you'll create and configure a second bookmark to show the second visual.

10. □ In the **Selection** pane, toggle the visibility of the two **Sales and Target by Month** items.

In other words, make the visible visual hidden, and make the hidden visual visible.

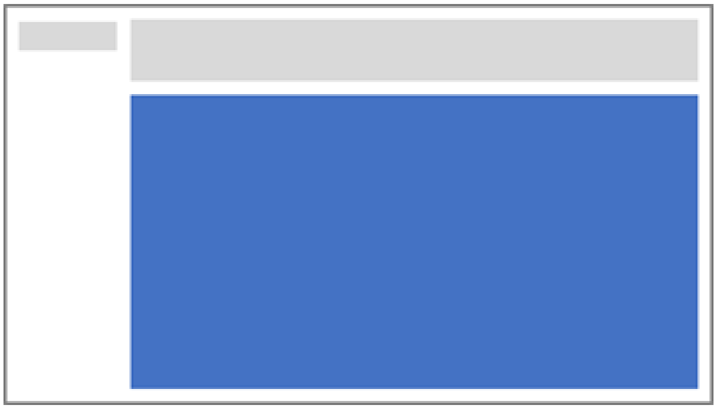


11. □ Create a second bookmark, and name it appropriately (either **Column Chart ON** or **Bar Chart ON).**



12. □ Configure the second bookmark to ignore filters (**Data** option off), and update the bookmark.
13. □ In the **Selection** pane, to make both visuals visible, simply show the hidden visual.
14. □ Resize and reposition both visuals so they fill the page beneath the multi-card visual, and completely overlap one another.

*Tip: To select the visual that is covered up, select it in the **Selection** pane.*



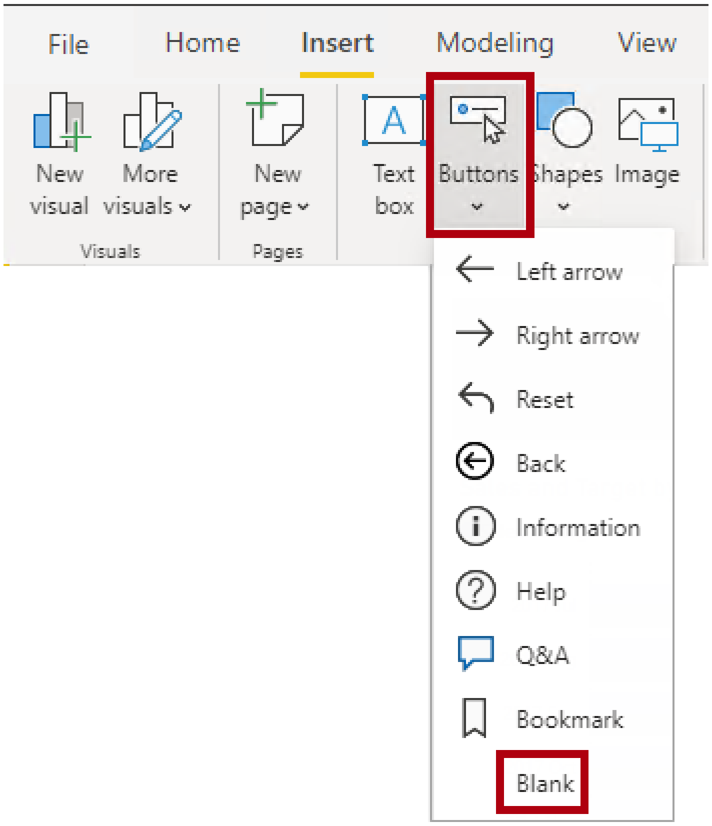
15. □ In the **Bookmarks** pane, select each of the bookmarks, and notice that only one of the visuals is visible.

The next stage of design is to add two buttons to the page, which will allow the report user to select the bookmarks.

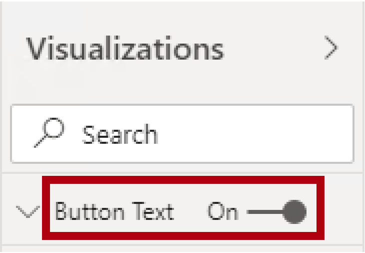
Task 2: Add buttons

In this task you will add two buttons, and assign bookmark actions to each.

1. □ On the **Insert** ribbon, from inside the **Elements** group, click **Button**, and then select **Blank**.



2. □ Position the button directly beneath the **Year** slicer.
3. □ Select the button, and then in the **Visualizations** pane, turn the **Button Text** property to **On**.

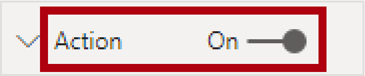


Visualizations

Search

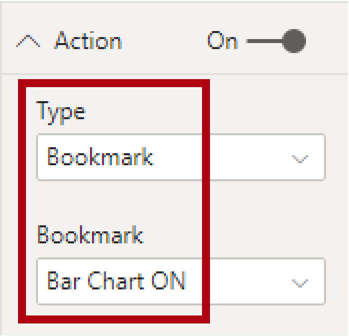
Button Text On

4. ☐ Expand the **Button Text** section, and then in the **Button Text** box, enter **Bar Chart**.
5. ☐ Expand the **Background** section, and then set a background color using a complementary color.
6. ☐ Turn the **Action** property to **On** (located near the bottom of the list).



Action On

7. ☐ Expand the **Action** section, and then set the **Type** dropdown list to **Bookmark**.
8. ☐ In the **Bookmark** dropdown list, select **Bar Chart ON**.



Action On

Type: Bookmark

Bookmark: Bar Chart ON

9. ☐ Create a copy of the button by using copy and paste, and then configure the new button as follows:

   *Tip: The shortcut commands for copy and paste are **Ctrl+C** followed by **Ctrl+V**.*

   - Set the **Button Text** property to **Column Chart**
   - In the **Action** section, set the **Bookmark** dropdown list to **Column Chart ON**

   The design of the Sales Analysis report is now complete.

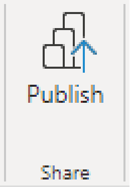
Task 3: Publish the report

In this task you will publish the report.

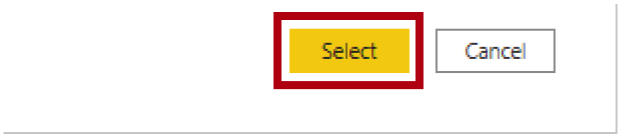
1. □ Select the **Overview** page.
2. □ In the **Year** slicer, select **FY2020**.
3. □ In the **Region** slicer, select **Select All**.
4. □ Save the Power BI Desktop file.

The file must always be saved prior to publishing to the Power BI service.

5. □ On the **Home** ribbon tab, from inside the **Share** group, click **Publish**.



6. □ In the **Publish to Power** BI window, notice that **My Workspace** is selected.
7. □ To publish the report, click **Select**.



8. □ If prompted to replace the dataset, click **Replace**.
9. □ When the publication has succeeded, click **Got It**.

10. □ Close Power BI Desktop.

You'll explore the report in the Power BI service in the next exercise.

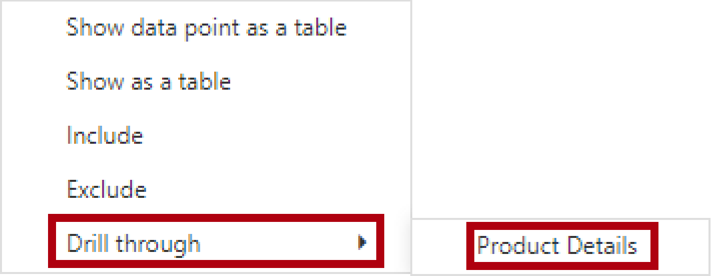
Exercise 5: Explore the Report

In this exercise you will explore the report in the Power BI service.

Task 1: Explore the report

In this task you will explore the report in the Power BI service.

1. □ In the Microsoft Edge browser window, in the Power BI service, in the **Navigation** pane, click the **Sales Analysis** report.
2. □ To test the drill through report, in the **Overview** page, in the **Quantity by Category** visual, right-click the **Clothing** bar, and then select **Drill Through | Product Details**.

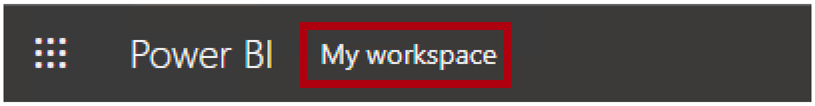


3. □ Notice that the **Product Details** page is for **Clothing**.
4. □ To return to the source page, at the top-left corner of the page, click the arrow button.
5. □ Select the **My Performance** page.
6. □ Click each of the buttons, and then notice that a different visual is displayed.

Task 2: Finish up

In this task you will complete the lab.

1. □ To return to your workspace, in the banner across the window web page, click **My Workspace**.



2. □ Leave the Microsoft Edge browser window open.

Congratulations!

Create a Power BI Dashboard

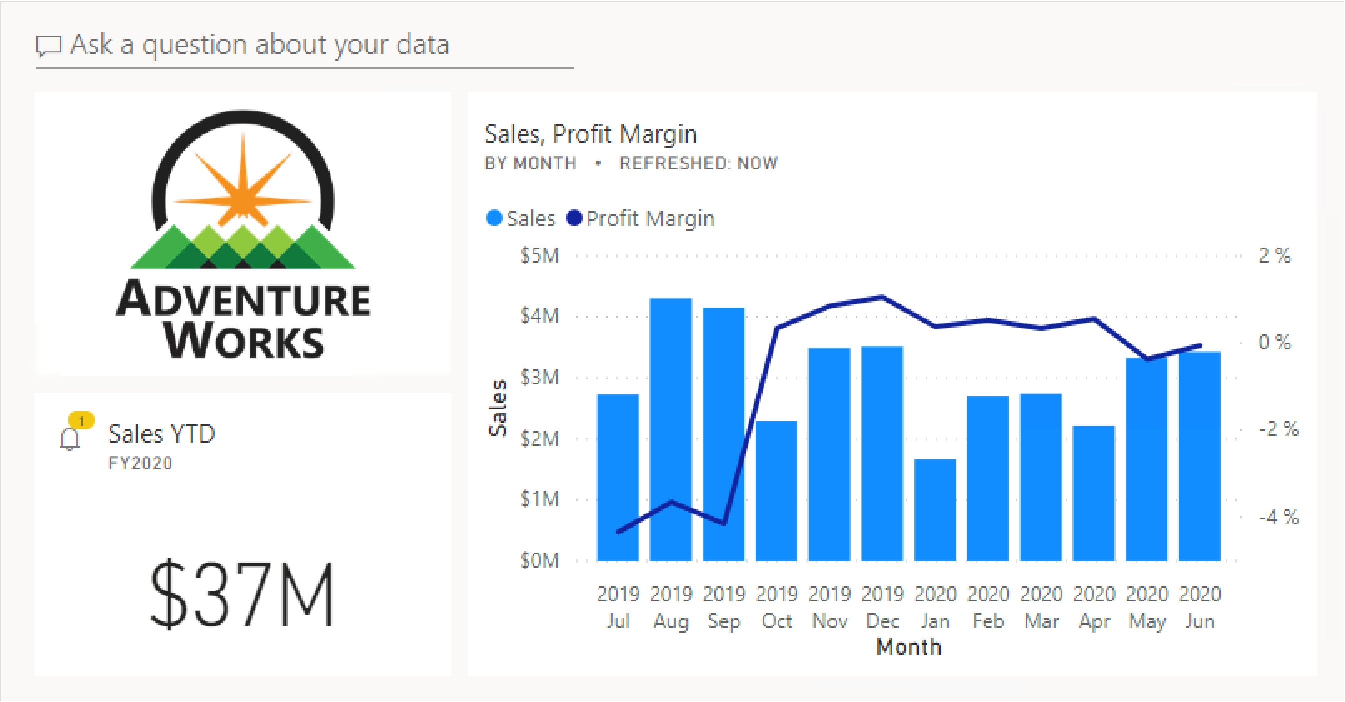
In this lab you will create the **Sales Monitoring** dashboard.

In this lab you learn how to:

- Pin visuals to a dashboard
- Use Q&A to create dashboard tiles
- Configure a dashboard tile alert

Exercise 1: Create a Dashboard

In this exercise you will create the **Sales Monitoring** dashboard. The completed dashboard will look like the following:



Task 1: Get started – Sign in

In this task you will setup the environment for the lab by signing in to Power BI.

Important: If you have already signed in to Power BI in a previous lab, continue from the next task.

1. □ In the Microsoft Edge browser window, in the Power BI service, in the **Navigation** pane, expand **My Workspace**.

Task 2: Get started – Open report

In this task you will setup the environment for the lab by opening the starter report.

Important: If you are continuing on from the previous lab (and you completed that lab successfully), do not complete this task; instead, continue from the next task.

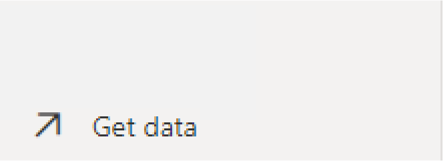
1. □ In the **Open** window, navigate to the **D:\DA100\Labs\09-create-power-bi-dashboard\Starter** folder.
2. □ Select the **Sales Analysis** file.

Task 3: Get started – Publish the report

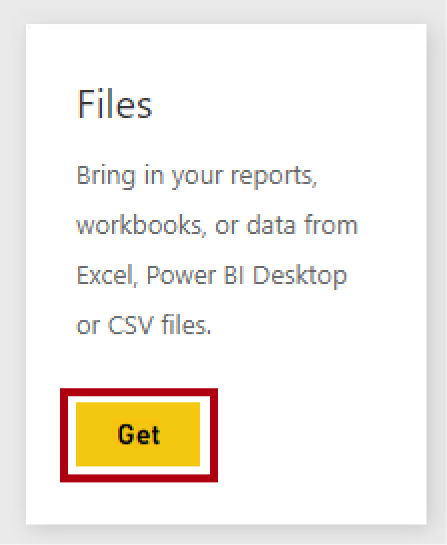
In this task you will setup the environment for the lab by creating a dataset.

Important: If you have already published the report in the ***Design a Report in Power BI Desktop, Part 2*** *lab, continue from the next task.*

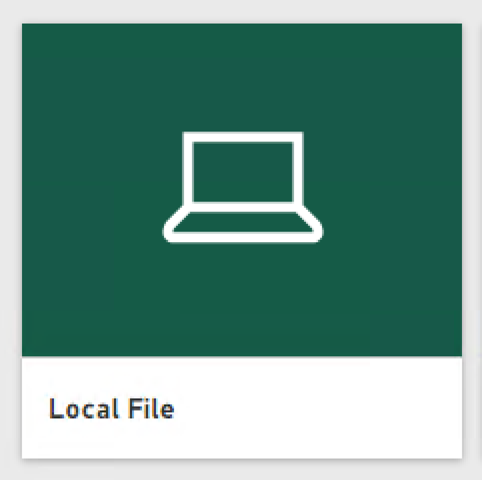
1. □ In the Microsoft Edge browser window, in the Power BI service, in the **Navigation** pane, at the bottom, click **Get Data**.



2. □ In the **Files** tile, click **Get**.


Files
Bring in your reports, workbooks, or data from Excel, Power BI Desktop or CSV files.
Get

3. □ Click the **Local File** tile.

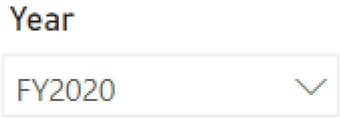


4. □ In the **Open** window, navigate to the **D:\DA100\Labs\08-design-report-in-power-bi-desktop-enhanced\Solution** folder.
5. □ Select the **Sales Analysis.pbix** file, and then click **Open**.
6. □ If prompted to replace the dataset, click **Replace**.

Task 4: Create a dashboard

In this task you will create the **Sales Monitoring** dashboard. You will pin a visual from the report, and add a tile based on an image data URI, and use Q&A to create a tile.

1. □ In the Microsoft Edge browser window, in the Power BI service, open the **Sales Analysis** report.
2. □ In the **Overview** page, set the **Year** slicer to **FY2020**.



Year

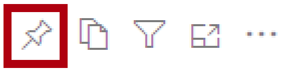
FY2020

3. □ Set the **Region** slicer to **Select All**.

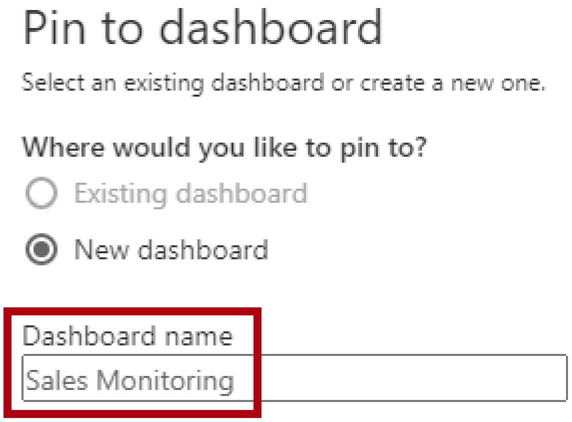
When pinning visuals to a dashboard, they will use the current filter context. Once pinned, the filter context cannot be changed. For time-based filters, it's a better idea to use a relative date slicer (or, Q&A using a relative time-based question).

4. □ To create a dashboard and pin a visual, hover the cursor over the **Sales and Profit Margin by Month** (column/line) visual.

5. □ At the top-right corner, click the pushpin.



6. □ In the **Pin to Dashboard** window, in the **Dashboard Name** box, enter **Sales Monitoring**.



Pin to dashboard

Select an existing dashboard or create a new one.

Where would you like to pin to?

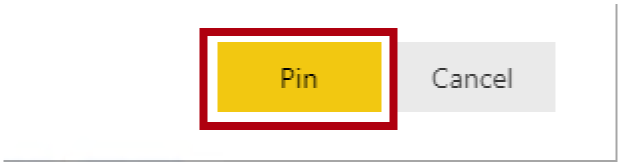
○ Existing dashboard

● New dashboard

Dashboard name

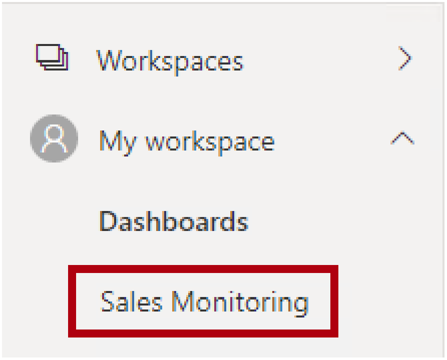
Sales Monitoring

7. □ Click **Pin**.

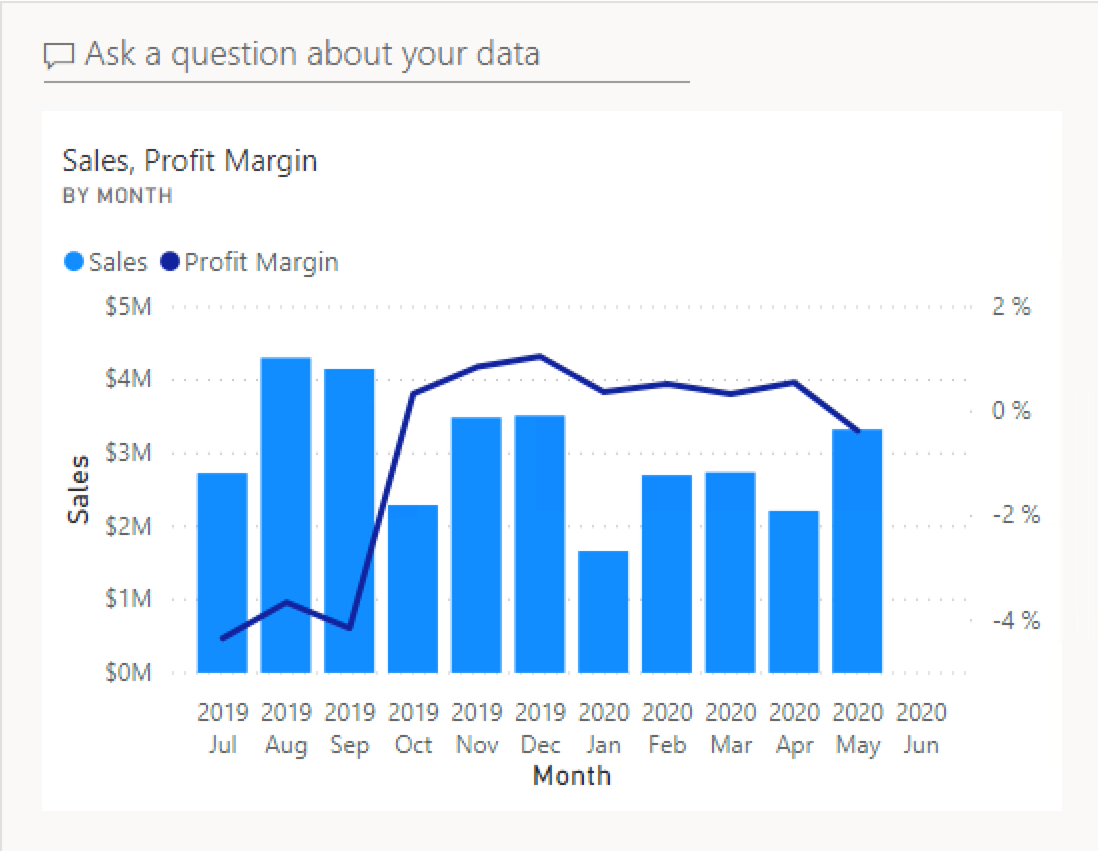


8. □ Open the **Navigation** pane, and then open the **Sales Monitoring** dashboard.

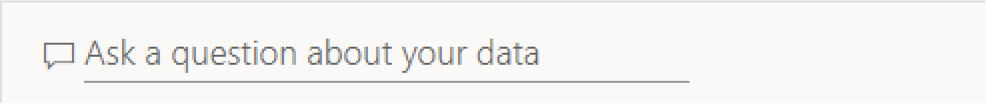
If prompted, click save and continue.



9. □ Notice that the dashboard has a single tile.

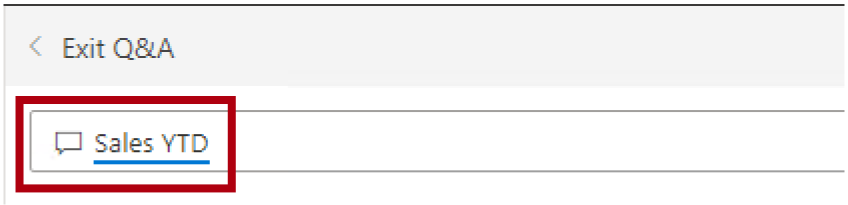


10. ☐ To add a tile based on a question, at the top-left of the dashboard, click **Ask a Question About Your Data**.

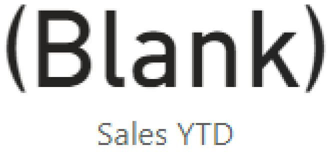


You can use the Q&A feature to ask a question, and Power BI will respond with a visual.

11. ☐ Click any one of the suggested questions beneath the Q&A box, in blue boxes.
12. ☐ Review the response.
13. ☐ Remove all text from the Q&A box.
14. ☐ In the Q&A box, enter the following: **Sales YTD**



15. □ Notice the response of **(Blank)**.

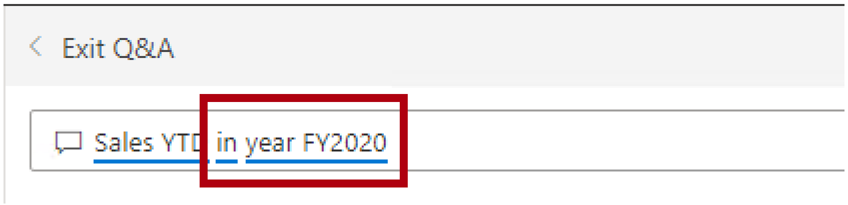


(Blank)

Sales YTD

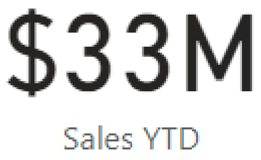
*You may recall you added the **Sales YTD** measure in the **Create DAX Calculations in Power BI Desktop, Part 2** lab. This measure is a Time Intelligence expression and it so requires a filter on the **Date** table to produce a result.*

16. □ Extend the question with: **in year FY2020**.



Sales YTD in year FY2020

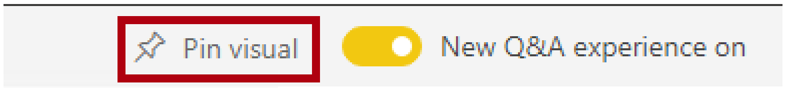
17. □ Notice the response is now **$33M**.



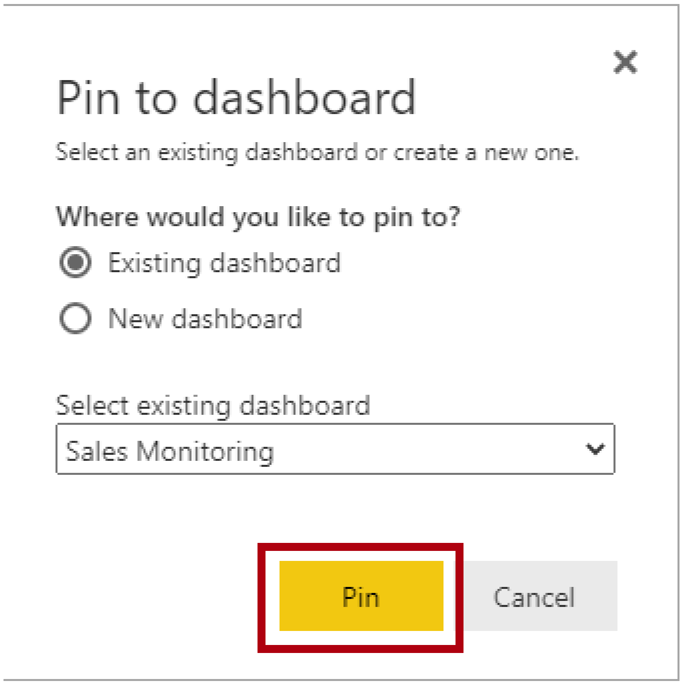
$33M

Sales YTD

18. □ To pin the response to the dashboard, at the top-right corner, click **Pin Visual**.



19. □ When prompted to pin the tile to the dashboard, click **Pin**.



Pin to dashboard

Select an existing dashboard or create a new one.

Where would you like to pin to?

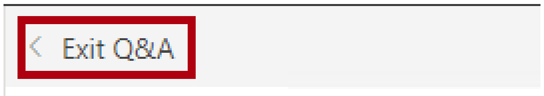
● Existing dashboard

○ New dashboard

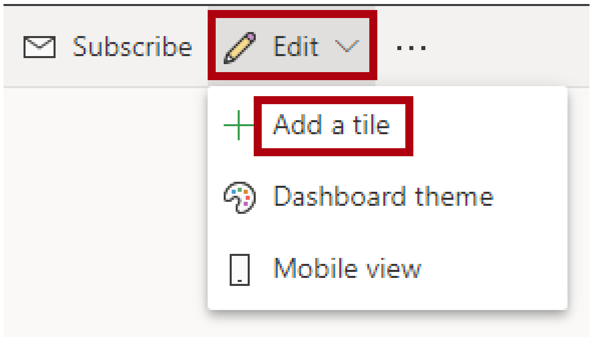
Select existing dashboard

Sales Monitoring

20. □ To return to the dashboard, at the top-left corner, click **Exit Q&A**.

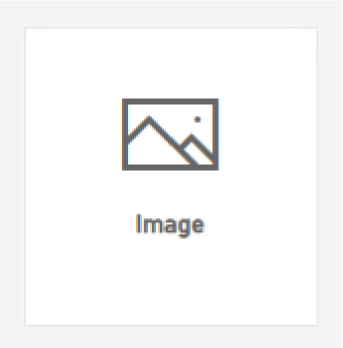


21. □ To add the company logo, on the menu bar, click **Edit**, and then select **Add a Tile**.

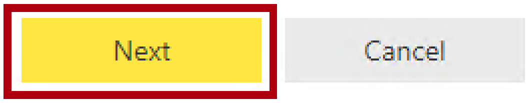


Using this technique to add a dashboard tile lets you embellish your dashboard with media, including web content, images, richly-formatted text boxes, and video (using YouTube or Vimeo links).

22. □ In the **Add a Tile** pane (located at the right), select the **Image** tile.



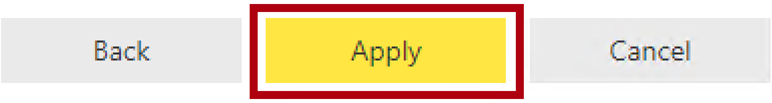
23. □ Click **Next**.



24. □ In the **Add Image Tile** pane, in the **URL** box, enter the complete URL found in the **D:\DA100\Resources\AdventureWorksLogo_DataURL.txt** file.

You can embed an image by using its URL, or you can use a data URL, which embeds content inline.

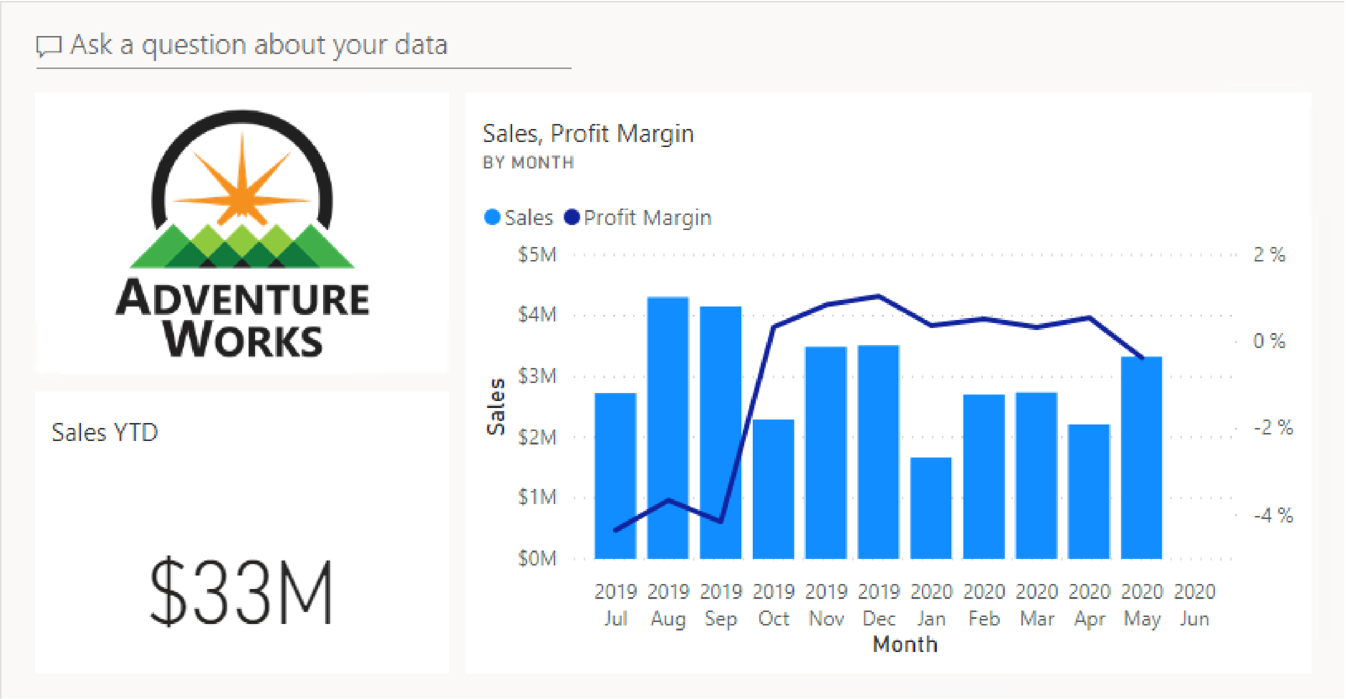
25. □ At the bottom of the pane, click **Apply**.



26. □ To resize the logo tile, drag the bottom-right corner, and resize the tile to become one unit wide, and two units high.

Tile sizes are constrained into a rectangular shape. It's only possible to resize into multiples of the rectangular shape.

27. □ Organize the tiles so that the logo appears at the top-left, with the **Sales YTD** tile beneath it, and the **Sales, Profit Margin** tile at the right.



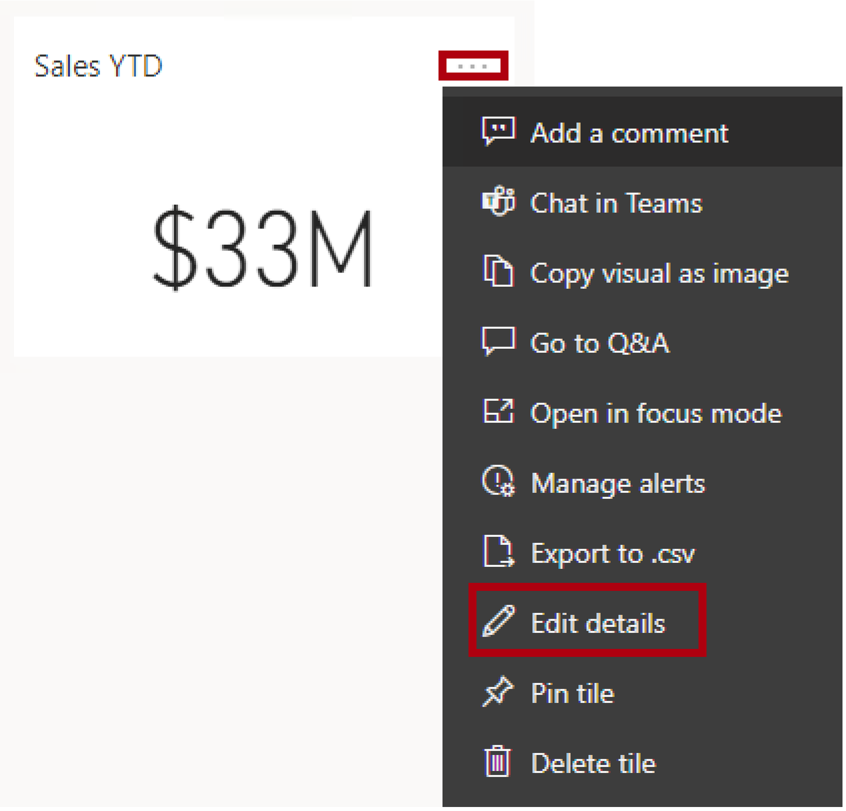
Sales YTD

$33M

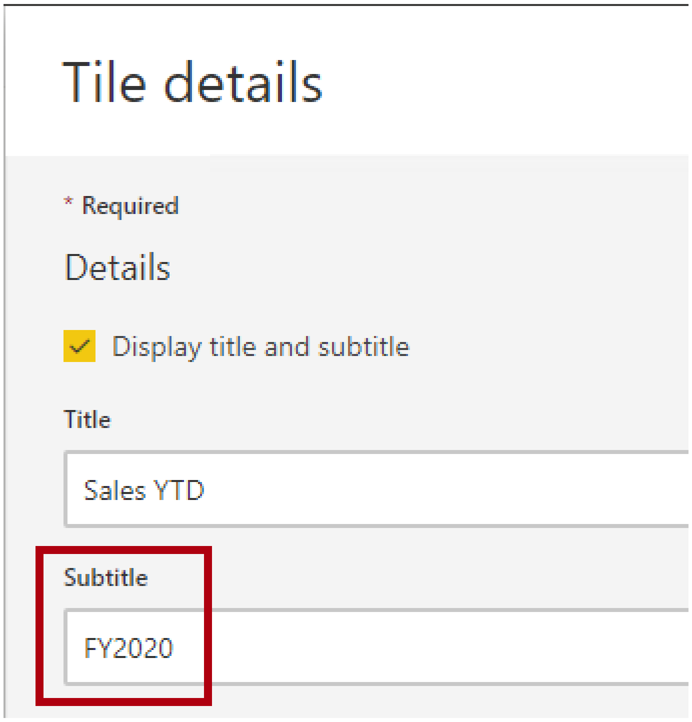
Task 5: Edit tile details

In this task you will edit the details of two tiles.

1. □ Hover the cursor over the **Sales YTD** tile, and then at the top-right of the tile, click the ellipsis, and then select **Edit Details**.



2. □ In the **Tile Details** pane (located at the right), in the **Subtitle** box, enter **FY2020**.



Tile details

* Required

Details

☑ Display title and subtitle

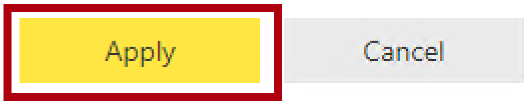
Title

Sales YTD

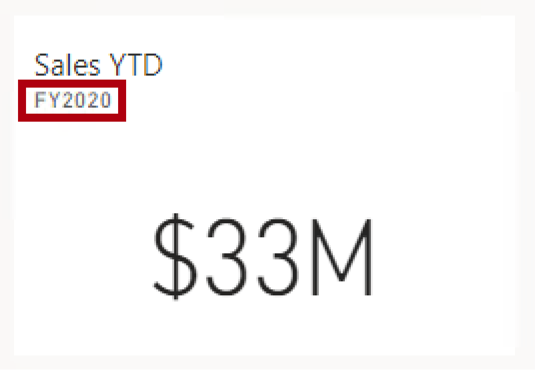
Subtitle

FY2020

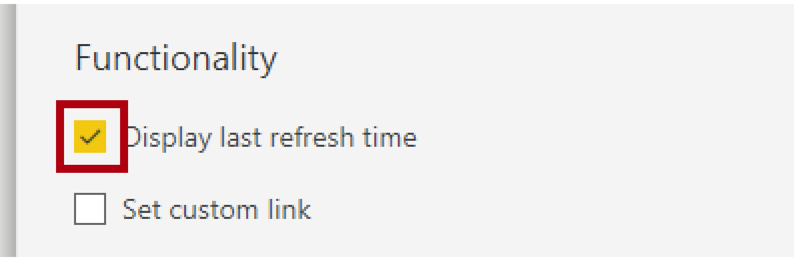
3. □ Click **Apply**.



4. □ Notice that the **Sales YTD** tile displays a subtitle.



5. □ Edit the tile details for the **Sales, Profit Margin** tile.
6. □ In the **Tile Details** pane, in the **Functionality** section, check **Display Last Refresh Time**.

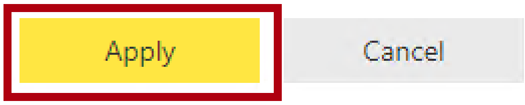


Functionality

☑ Display last refresh time

□ Set custom link

7. □ Click **Apply**.



8. □ Notice that the tile describes the last refresh time (which done when loading the data model in Power BI Desktop).

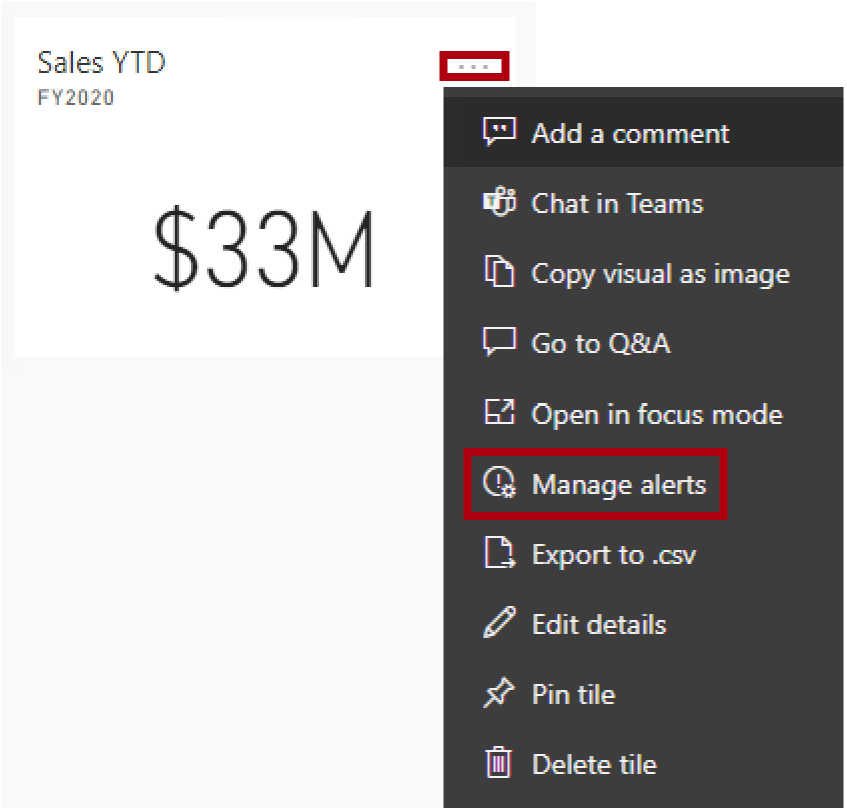
You'll simulate a data refresh later in this lab and see that the refresh time updates.

Task 6: Configure an alert

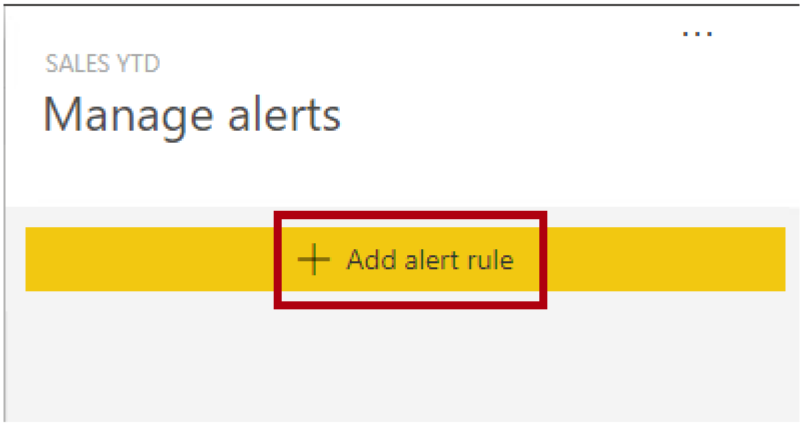
In this task you will configure a data alert.

Data alerts can only be configured on dashboard tiles, and specifically tiles that display a single numeric value.

1. □ Hover the cursor over the **Sales YTD** tile, click the ellipsis, and then select **Manage Alerts**.

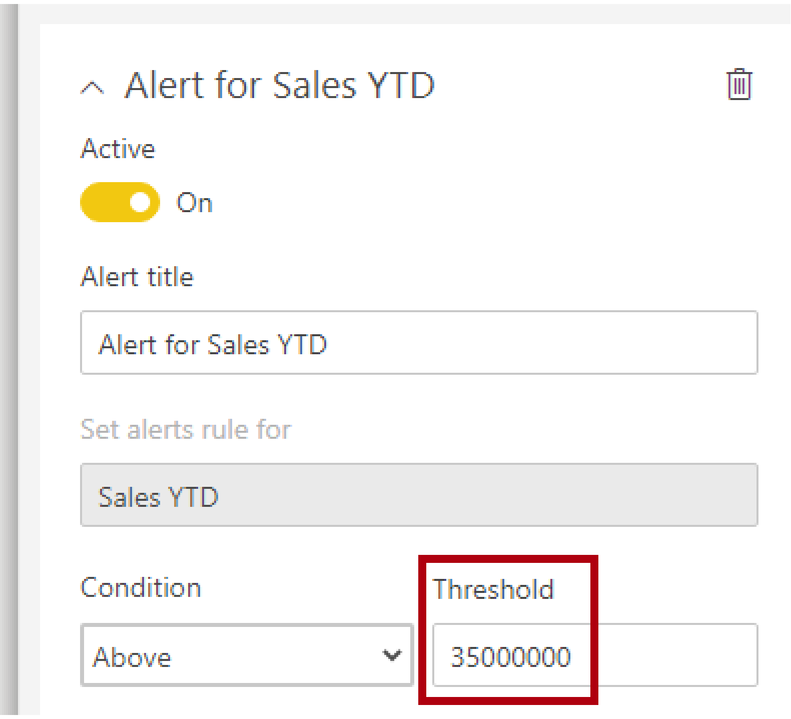


2. □ In the **Manage Alerts** pane (located at the right), click **Add Alert Rule**.


SALES YTD

Manage alerts

3. □ In the **Threshold** box, replace the value with **35000000** (35 million).


Alert for Sales YTD

Active: On

Alert title: Alert for Sales YTD

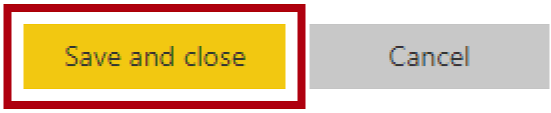
Set alerts rule for: Sales YTD

Condition: Above

Threshold: 35000000

This configuration ensures you'll be notified whenever the tile updates to a value above 35 million.

4. □ At the bottom of the pane, click **Save and Close**.



You'll refresh the dataset in the next exercise. Typically, this would be achieved by using scheduled refresh, in which case Power BI would use a gateway to connect to the SQL Server database. However, due to constraints in the classroom setup, there is no gateway. So, you'll open Power BI Desktop, perform a manual data refresh, and then upload the file to your workspace.

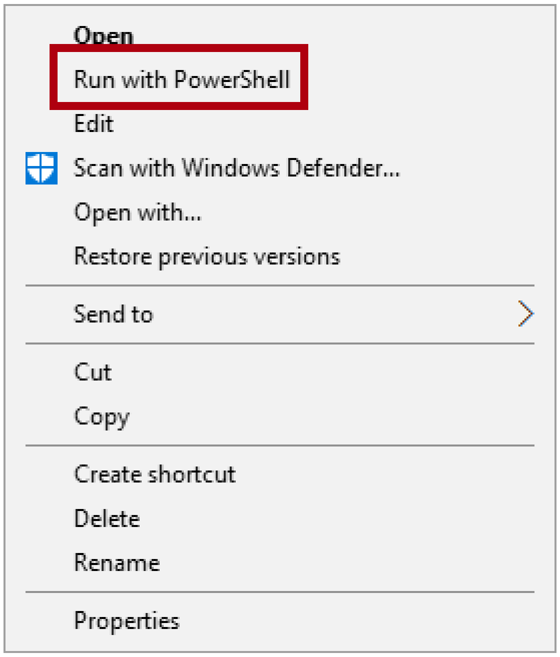
Exercise 2: Refresh the Dataset

In this exercise you will first load sales order data for June 2020 into the **AdventureWorksDW2020** database. You will then open your Power BI Desktop file, perform a data refresh, and then upload the file to your workspace.

Task 1: Update the lab database

In this task you will run a PowerShell script to update data in the **AdventureWorksDW2020** database.

1. □ In File Explorer, inside the **D:\DA100\Setup** folder, right-click the **UpdateDatabase-2-AddSales.ps1** file, and then select **Run with PowerShell**.



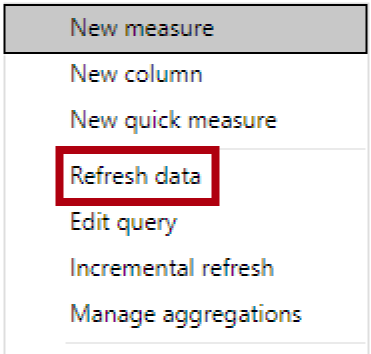
2. □ If prompted to change the execution policy, press **A**.
3. □ When prompted to press any key to close, press **Enter** again.

*The **AdventureWorksDW2020** database now includes sales orders made in June 2020.*

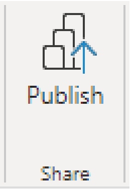
Task 2: Refresh the Power BI Desktop file

In this task you will open the **Sales Analysis** Power BI Desktop file, perform a data refresh, and then upload the file to your **Sales Analysis** workspace.

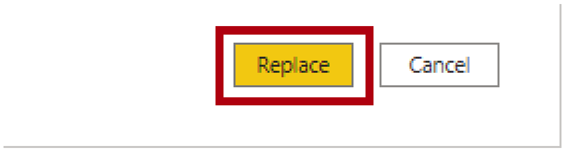
1. □ In Power BI Desktop file, in the **Fields** pane, right-click the **Sales** table, and then select **Refresh Data**.



2. □ When the refresh completes, save the Power BI Desktop file.
3. □ To publish the file to your workspace, on the **Home** ribbon tab, from inside the **Share** group, click **Publish**.



4. □ When prompted to replace the dataset, click **Replace**.



The dataset in the Power BI service now has June 2020 sales data.

5. □ Close Power BI Desktop.

Exercise 3: Review the Dashboard

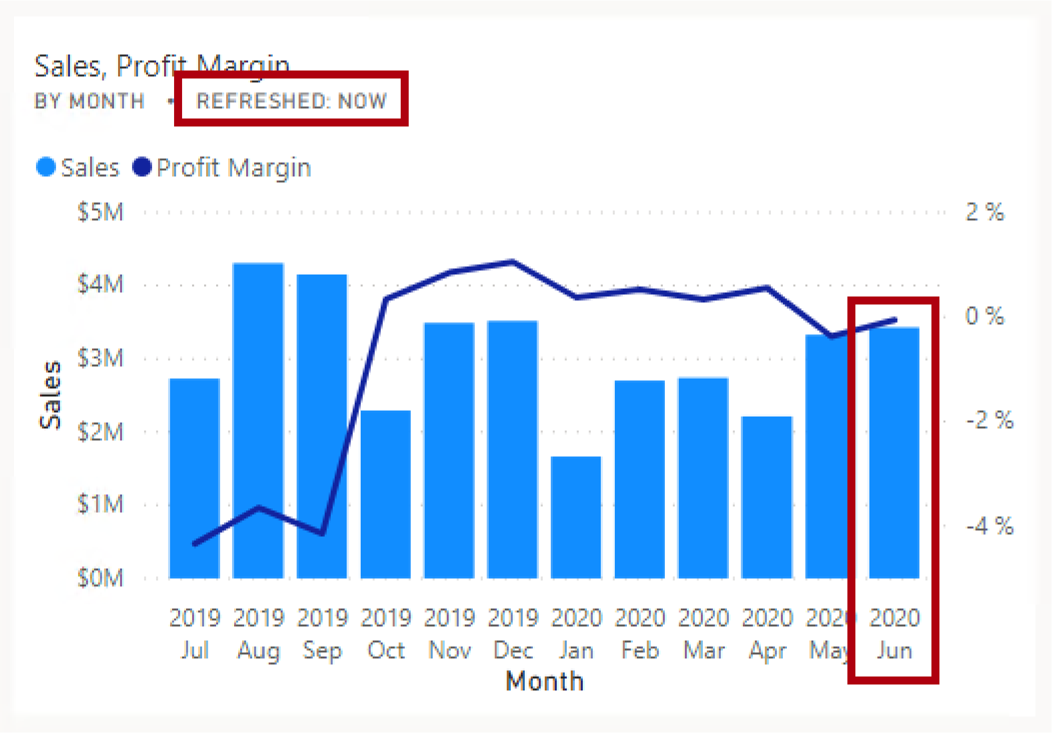
In this exercise you will review the dashboard to notice updated sales, and that the alert was triggered.

Task 1: Review the dashboard

In this task you will review the dashboard to notice updated sales, and that the alert was triggered.

1. □ In the Microsoft Edge browser window, in the Power BI service, review the **Sales Monitoring** dashboard.
2. □ In the **Sales, Profit Margin** tile, in the subtitle, notice that the data was refreshed **NOW**.
3. □ Notice also that there is now a column for **2020 Jun**.

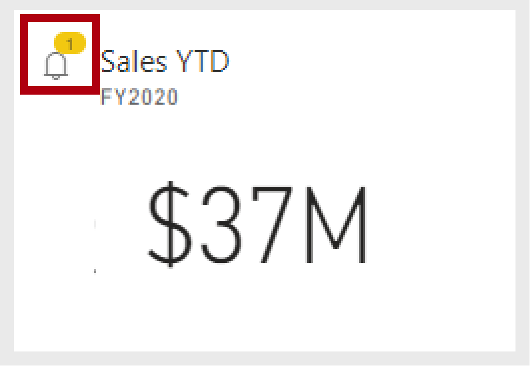
If you don't see the June 2020 data, you might need to press ***F5*** *to reload the web browser.*



The alert on the ***Sales YTD*** *tile should have triggered also. After a short while, the alert should notify you that sales now exceeds the configured threshold value.*

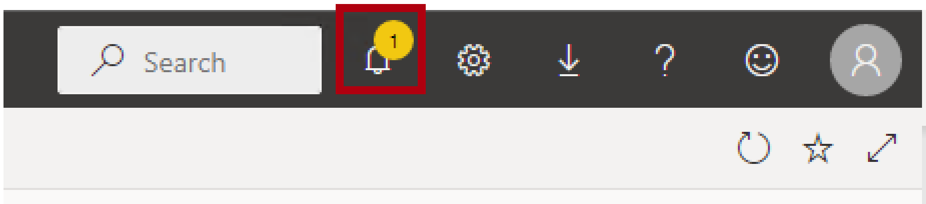
4. □ Notice that the **Sales YTD** tile has updated to **$37M**.
5. □ Verify that the **Sales YTD** tile displays an alert notification icon.

If you don't see the notification, you might need to press ***F5*** *to reload the browser. If you still don't see the notification, wait some minutes longer.*



Alert notifications appear on the dashboard tile, and can be delivered by email, and push notifications to mobile apps including the Apple Watch.

6. □ At the top-right corner of the web page, click the **Notifications** icon.



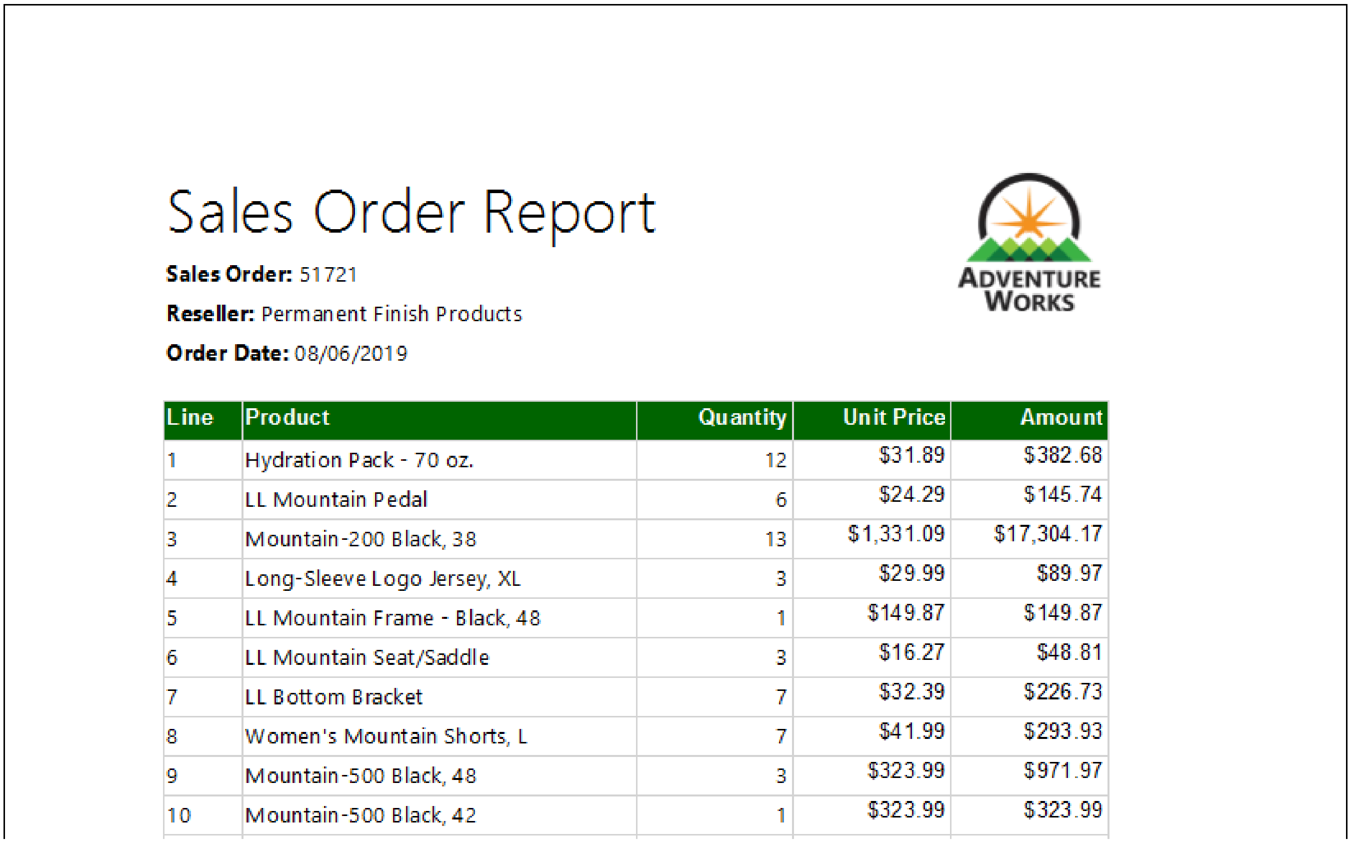
7. □ In the **All Notifications** pane, review the details of the alert notification.
8. □ To close the pane, click **Close**.

Congratulations!

Create a Power BI Paginated Report

In this lab you will use Power BI Report Builder to develop a pixel-perfect paginated report layout that sources data from the **AdventureWorksDW2020** SQL Server database. You will create a data source and dataset, and also configure a report parameter. The report layout will allow data to be rendered over multiple pages, and to be exported in PDF and other formats.

The final report will look like the following:



Sales Order Report

Sales Order: 51721

Reseller: Permanent Finish Products

Order Date: 08/06/2019

ADVENTURE WORKS

Line	Product	Quantity	Unit Price	Amount
1	Hydration Pack - 70 oz.	12	$31.89	$382.68
2	LL Mountain Pedal	6	$24.29	$145.74
3	Mountain-200 Black, 38	13	$1,331.09	$17,304.17
4	Long-Sleeve Logo Jersey, XL	3	$29.99	$89.97
5	LL Mountain Frame - Black, 48	1	$149.87	$149.87
6	LL Mountain Seat/Saddle	3	$16.27	$48.81
7	LL Bottom Bracket	7	$32.39	$226.73
8	Women's Mountain Shorts, L	7	$41.99	$293.93
9	Mountain-500 Black, 48	3	$323.99	$971.97
10	Mountain-500 Black, 42	1	$323.99	$323.99

In this lab you learn how to:

- Use Power BI Report Builder
- Design a multi-page report layout
- Define a data source
- Define a dataset
- Create a report parameter
- Export a report to PDF

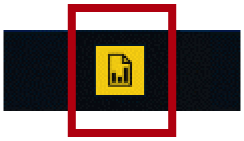
Exercise 1: Get Started

In this exercise you will open Power BI Report Builder to create and then save a report.

Task 1: Create the report

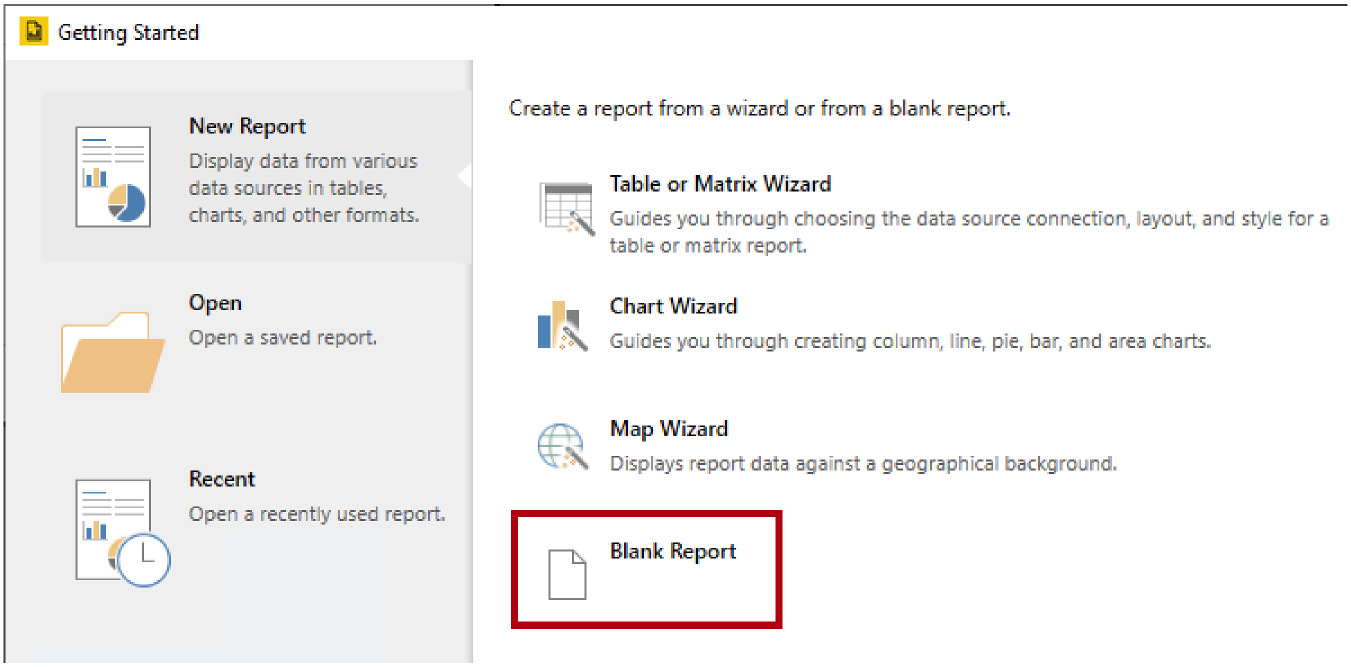
In this task you will open Power BI Report Builder to create and then save a report.

1. □ To open Power BI Report Builder, on the desktop, click the **Power BI Report Builder** shortcut.



*Note: if prompted to update to the latest version of Power BI Report Builder, click **Cancel** and proceed to step 2.

2. □ In the Power BI Report Builder window, to create a new report, in the **Getting Started** window, click **Blank Report**.



Getting Started

New Report
Display data from various data sources in tables, charts, and other formats.

Open
Open a saved report.

Recent
Open a recently used report.

Create a report from a wizard or from a blank report.

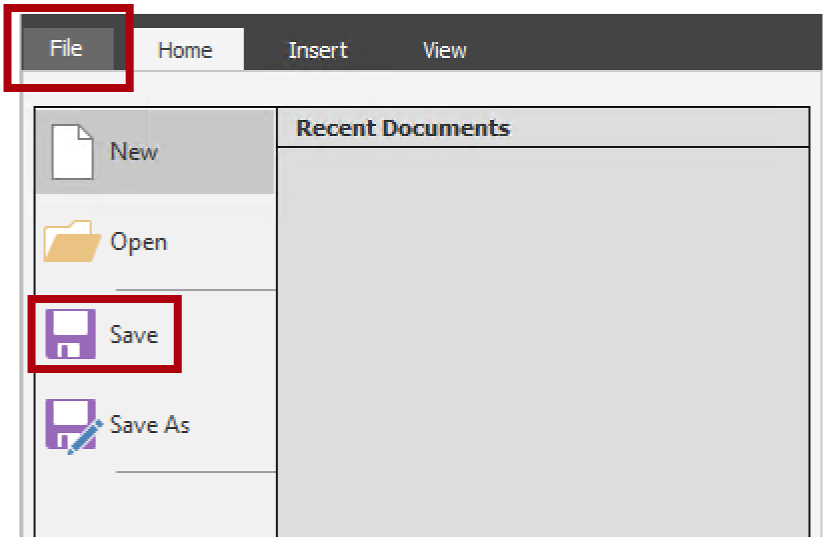
Table or Matrix Wizard
Guides you through choosing the data source connection, layout, and style for a table or matrix report.

Chart Wizard
Guides you through creating column, line, pie, bar, and area charts.

Map Wizard
Displays report data against a geographical background.

Blank Report

3. □ To save the report, click the **File** tab (located at the top-left), and then select **Save**.



4. □ In the **Save As Report** window, navigate to the **D:\DA100\MySolution** folder.
5. □ In the **Name** box, enter **Sales Order Report**.
6. □ Click **Save**.

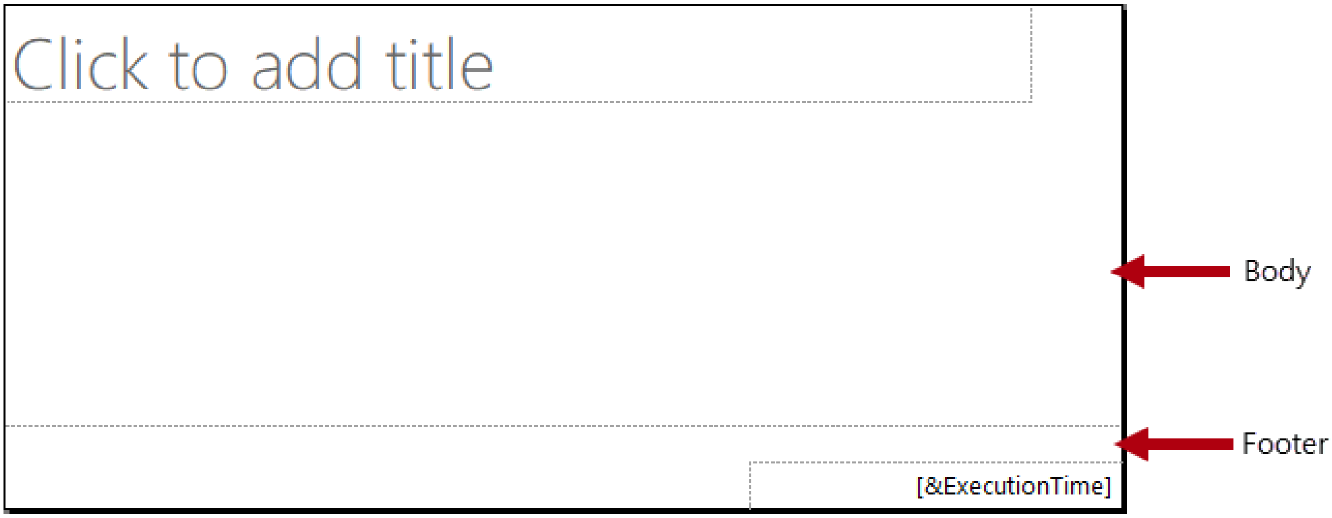
Exercise 2: Design the Report Layout

In this exercise you will design the report layout, and explore the final report design.

Task 1: Configure the report header

In this task you will configure the report header.

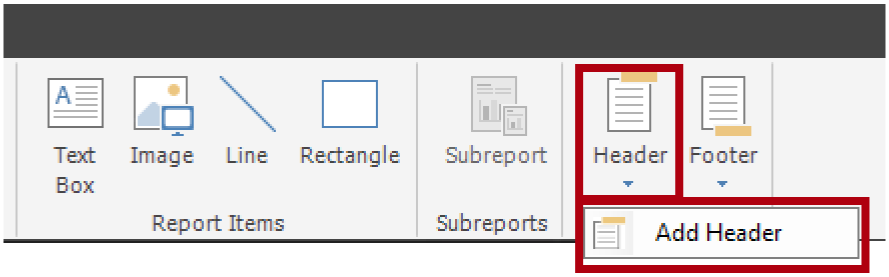
1. □ In the report designer, notice the default report layout, which consists of a body region and a report footer region.



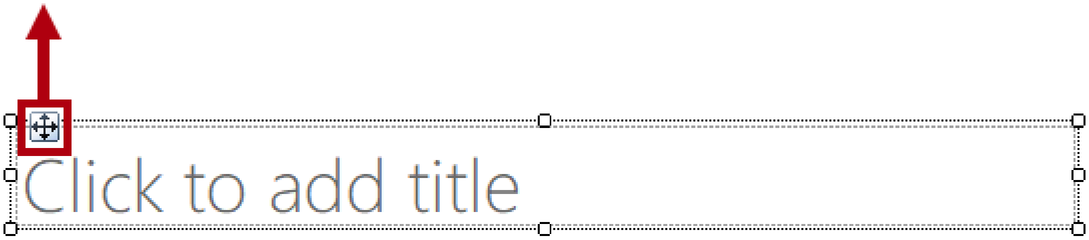
The body contains a single textbox ready for a report title, and the report footer contains a single textbox describing the report execution time.

The default design will render the report title once, in the body, on the first rendered page. However, you will now modify the report design by adding a report header region, and by moving the report title textbox into this region. This way, the report title will repeat on every page. You will also add an image of the company logo.

2. □ To add a report header region, on the **Insert** ribbon tab, from inside the **Header & Footer** group, click **Header**, and then select **Add Header**.



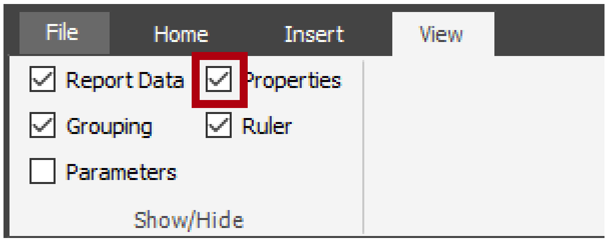
3. □ In the report designer, notice that a report header region has been added to the report layout.
4. □ To select the body textbox, click the "Click to add title" textbox.
5. □ To move the textbox, click the four-headed arrow icon, and then drag it into the header region to then drop it at the very top-left of the report header region.



6. □ To modify the report title textbox text, click inside the text box, and then enter: **Sales Order Report**

*To resize the textbox, you will first open the **Properties** pane. For fine-grained control of location and size properties, you will need use the **Properties** pane.*

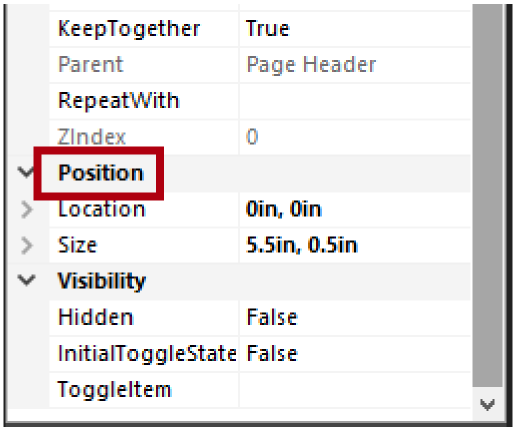
7. □ On the **View** ribbon tab, from inside the **Show/Hide** group, check **Properties**.



8. □ To select the report title textbox, first click an area outside the textbox, and then click the textbox again.

The textbox is selected when you see the border of the textbox highlighted and resizing handles (small circles) appear on the border.

9. □ In the **Properties** pane (located at the right), scroll down the list to locate the **Position** group.



KeepTogether	True
Parent	Page Header
RepeatWith	
ZIndex	0
Position	
Location	**0in, 0in**
Size	**5.5in, 0.5in**
Visibility	
Hidden	False
InitialToggleState	False
ToggleItem	

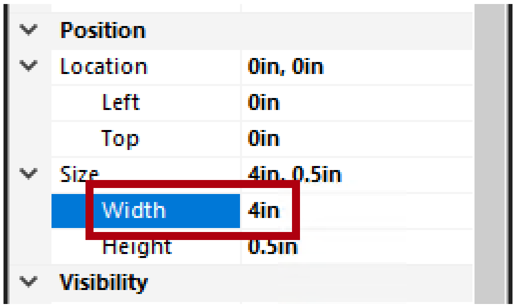
*The **Position** group allows setting exact values for the location and size of report items.*

It's important that you enter the values as directed in this lab. Pixel-perfect layout is required to achieve the page rendering at the end of the lab.

10. □ Within the **Position** group, expand the **Location** group, and ensure that the **Left** and **Top** properties are each set to **0in**.

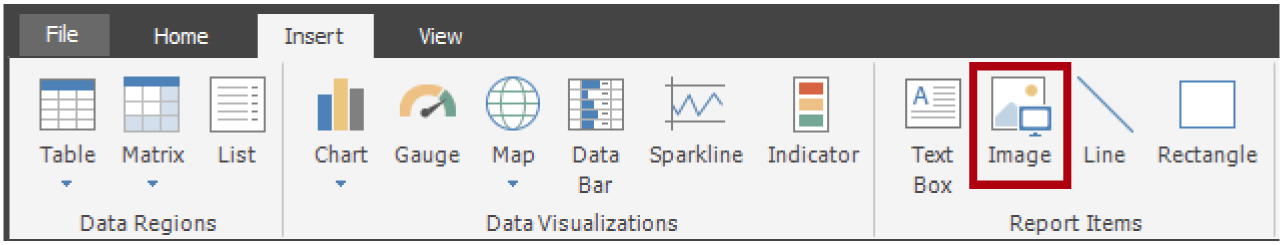
The location and size units are in inches because the regional settings of the lab virtual machine is set to the United States. If your region uses metric measurements, centimeters would be the default unit.

11. □ Within the **Position** group, expand the **Size** group, and then set the **Width** property to **4**.



Position	
Location	0in, 0in
Left	0in
Top	0in
Size	4in, 0.5in
Width	4in
Height	0.5in
Visibility	

12. □ To insert an image, on the **Insert** ribbon tab, from inside the **Report Items** group, click **Image**.



13. □ To add the image to the report design, click inside the report header region, to the right of the report title textbox.

14. □ In the **Image Properties** window, to import from an image file, click **Import**.

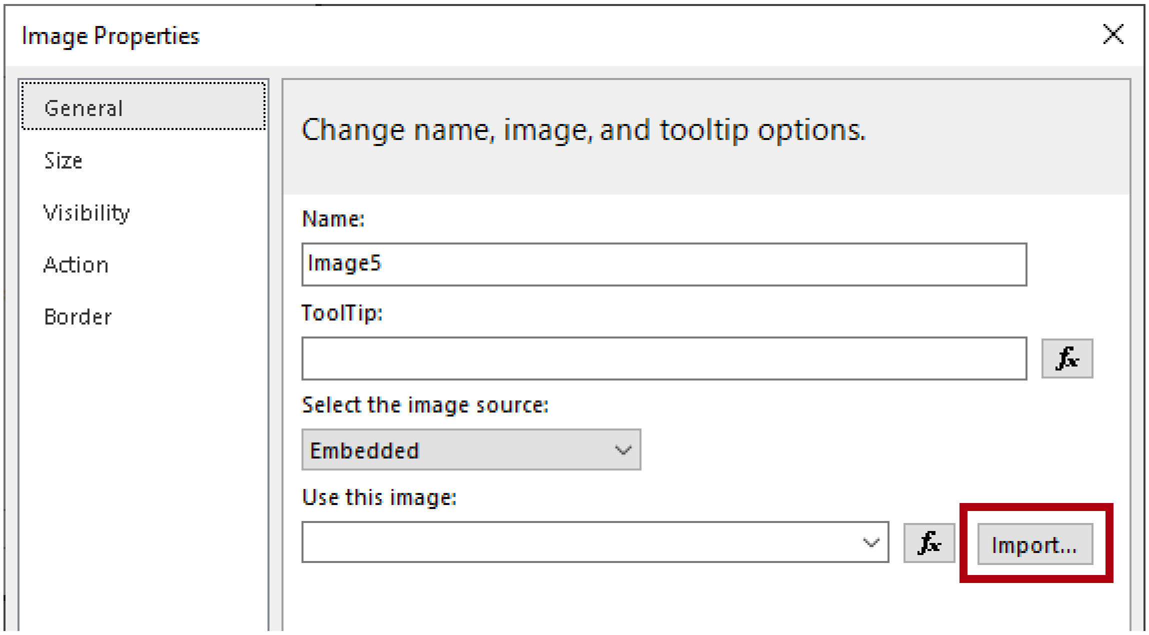


Image Properties

Change name, image, and tooltip options.

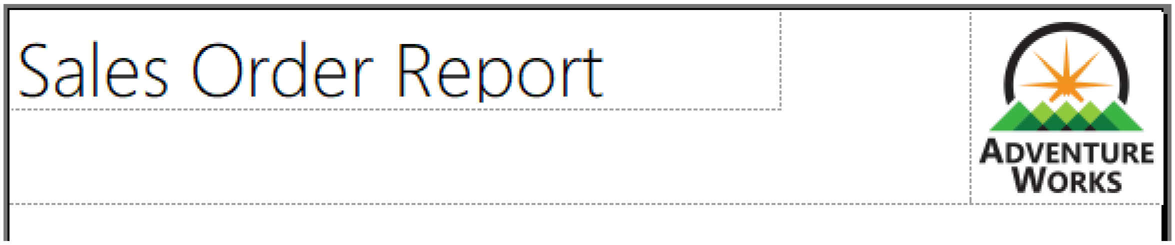
Name: Image5

ToolTip:

Select the image source: Embedded

Use this image:

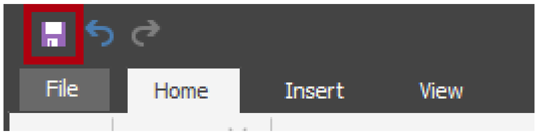
15. ☐ In the **Open** window, navigate to the **D:\DA100\Resources** folder, and then select the **AdventureWorksLogo.jpg** file.
16. ☐ Click **Open**.
17. ☐ In the **Image Properties** window, click **OK**.
18. ☐ In the report designer, notice that the image was added, and is selected.
19. ☐ To position and resize the image, in the **Properties** pane, configure the following properties:
    - **Position | Location | Left** : 5
    - **Position | Location | Top** : 0
    - **Position | Size | Width** : 1
    - **Position | Size | Height** : 1
20. ☐ To resize the report header region, first select the region by clicking a blank area of the region.
21. ☐ In the **Properties** pane, set the **General | Height** property to **1**.
22. ☐ Verify that the report header region contains a single textbox and image, and looks like the following:



Sales Order Report

23. □ To save the report, on the **File** tab, click **Save**.

Tip: You can also click the disk icon located at the top-left.

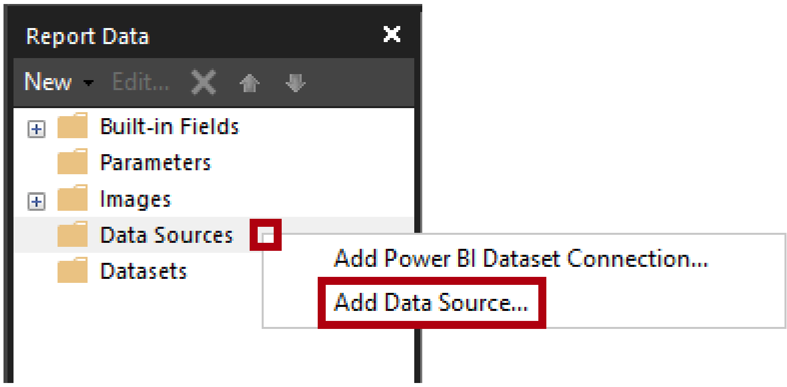


You are now ready to configure the report to retrieve a database query result.

Task 2: Retrieve data

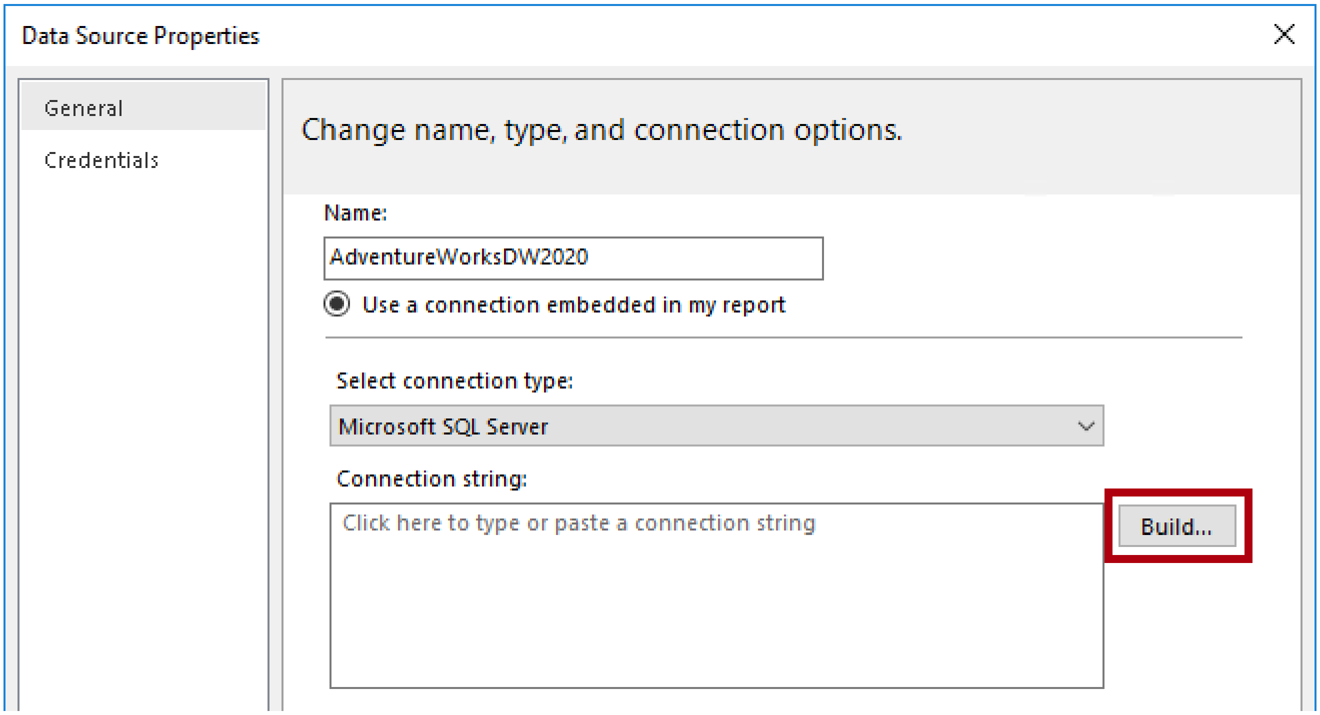
In this task you will create a data source and dataset to retrieve a query result from the **AdventureWorksDW2020** SQL Server database.

1. □ In the **Report Data** pane (located at the left), right-click the **Data Sources** folder, and then select **Add Data Source**.



It is possible to retrieve data from cloud or on-premises databases, or a Power BI dataset.

2. □ In the **Data Source Properties** window, in the **Name** box, replace the text with **AdventureWorksDW2020**.
3. □ In the **Select Connection Type** dropdown list, change the connection type from SQL Server Analysis Services to **Microsoft SQL Server**.
4. □ To build the connection string, click **Build**.



Data Source Properties

Change name, type, and connection options.

Name: AdventureWorksDW2020

● Use a connection embedded in my report

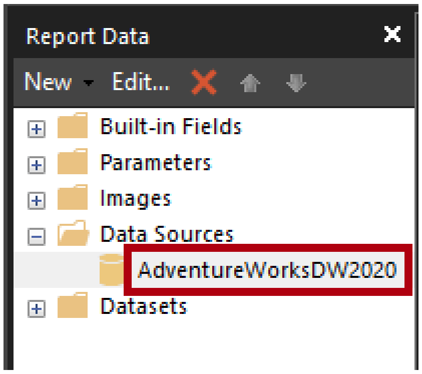
Select connection type: Microsoft SQL Server

Connection string: Click here to type or paste a connection string

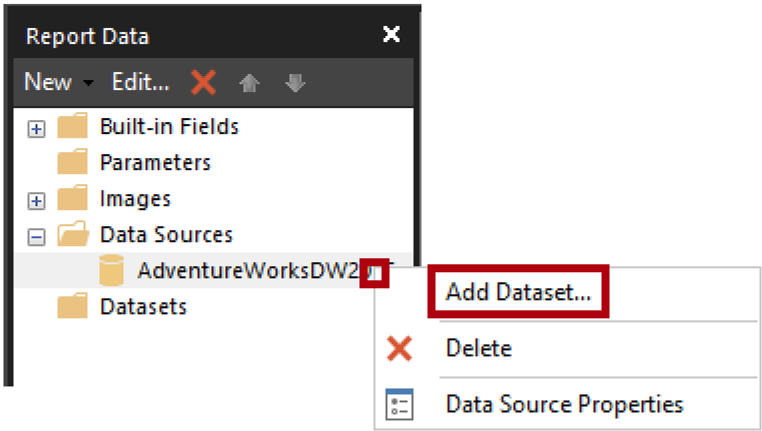
5. □ In the **Connection Properties** window, in the **Server Name** box, enter **localhost**.

In the labs, you will connect to the SQL Server database by using ***localhost****. This isn't a recommended practice, however, when creating your own solutions. It's because gateway data sources cannot resolve* ***localhost****.*

6. □ In the **Select or Enter a Database Name** dropdown list, select the **AdventureWorksDW2020**.
7. □ Click **OK**.
8. □ In the **Data Source Properties** window, click **OK**.
9. □ In the **Report Data** pane, notice the addition of the **AdventureWorksDW2020** data source.



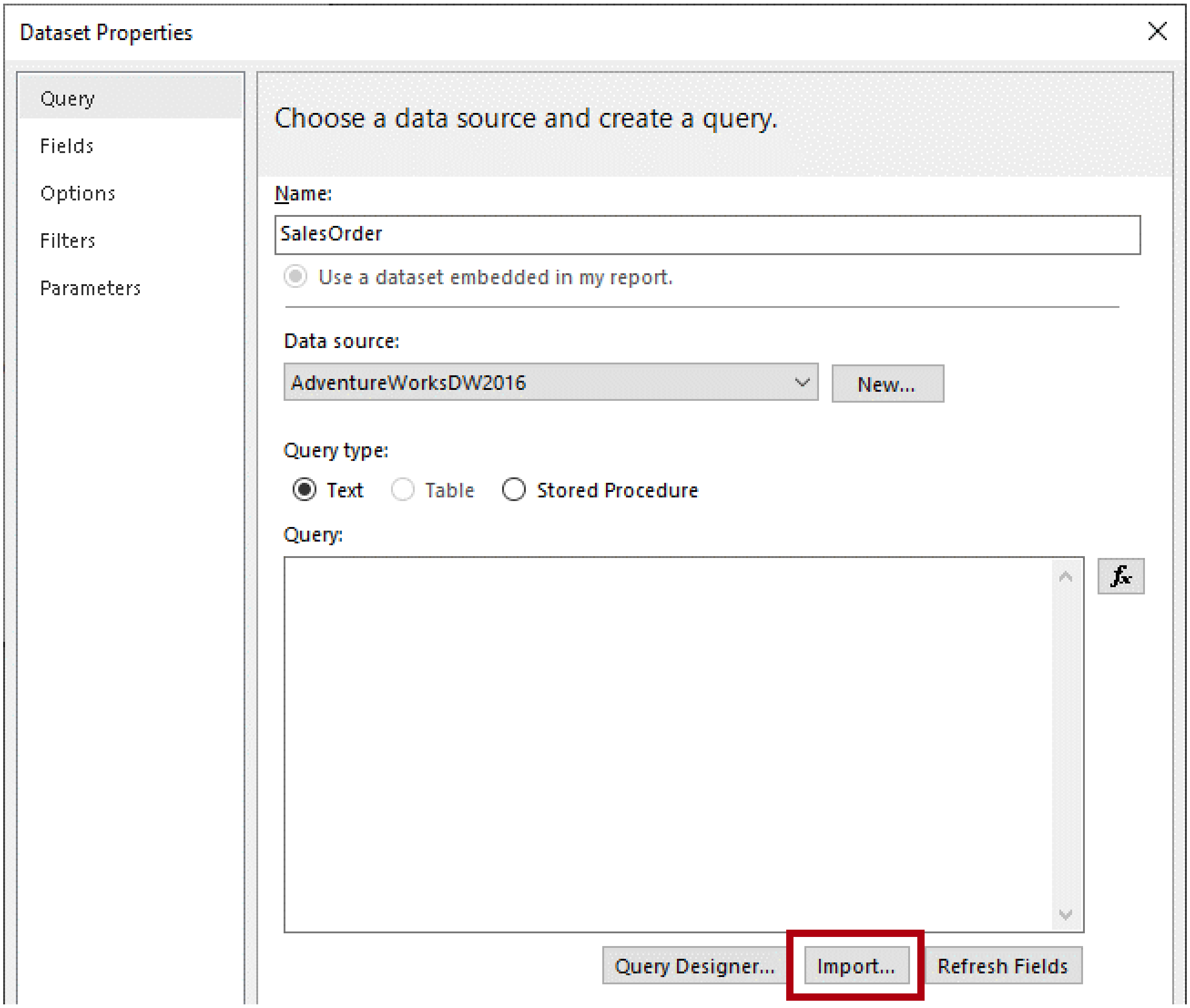
10. □ To create a dataset, in the **Report Data** pane, right-click the **AdventureWorksDW2020** data source, and then select **Add Dataset**.



A report dataset is a different in purpose and structure from a Power BI dataset.

11. □ In the **Dataset Properties** window, in the **Name** box, replace the text with **SalesOrder**.

12. □ To import a pre-defined query, click **Import**.



Name: SalesOrder

Data source: AdventureWorksDW2016

Query type: ● Text ○ Table ○ Stored Procedure

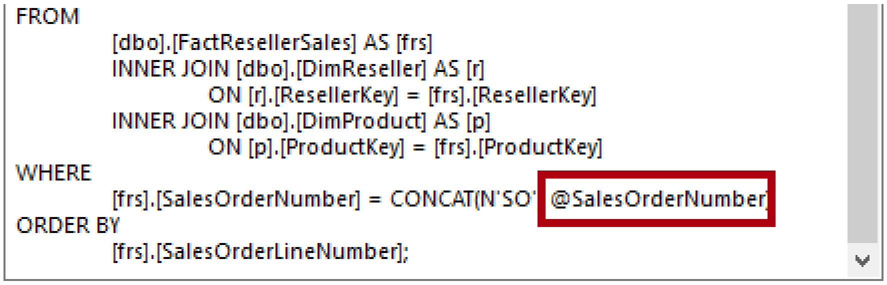
13. □ In the **Import Query** window, navigate to the **D:\DA100\Labs\10-create-power-bi-paginated-report\Assets** folder, and then select the **SalesOrder.sql** file.

14. □ Click **Open**.

15. □ In the **Query** box, review the query, and be sure to scroll down to the bottom of the query text.

    It is not important that you understand the details of the query statement. It has been designed to retrieve sales order line details. The WHERE clause includes a predicate to restrict the query result to a single sales order. The ORDER BY clause ensures the rows are returned by line number order.

16. □ Notice the use of **@SalesOrderNumber** in the WHERE clause, which represents a query parameter.

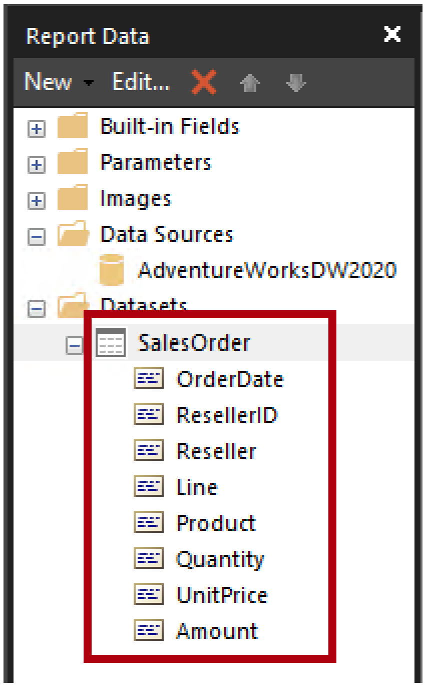


```
FROM
	[dbo].[FactResellerSales] AS [frs]
	INNER JOIN [dbo].[DimReseller] AS [r]
		ON [r].[ResellerKey] = [frs].[ResellerKey]
	INNER JOIN [dbo].[DimProduct] AS [p]
		ON [p].[ProductKey] = [frs].[ProductKey]
WHERE
	[frs].[SalesOrderNumber] = CONCAT(N'SO' @SalesOrderNumber
ORDER BY
	[frs].[SalesOrderLineNumber];
```

A query parameter is a placeholder for a value that will be passed in at query execution time. You will configure a report parameter to prompt the report user for a single sales order number which then will be passed to the query parameter.

17. □ Click **OK**.

18. □ In the **Report Data** pane, notice the addition of the **SalesOrder** dataset and its fields.



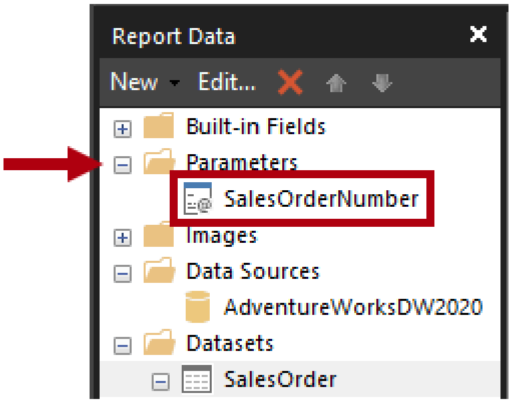
Fields are used to configure data regions in the report layout. They were derived from the dataset query columns.

19. □ Save the report.

Task 3: Configure the report parameter

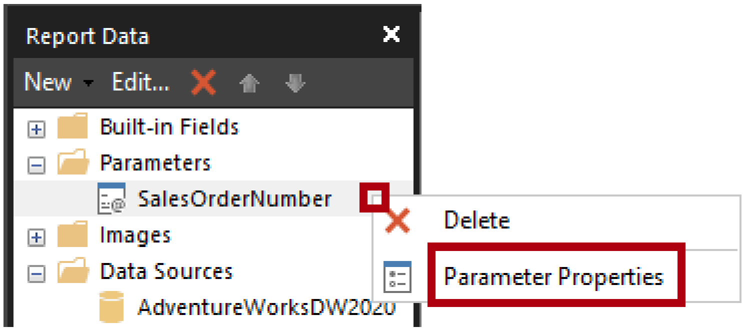
In this task you will configure the report parameter with a default value.

1. □ In the **Report Data** pane, expand the **Parameters** folder to reveal the **SalesOrderNumber** report parameter.

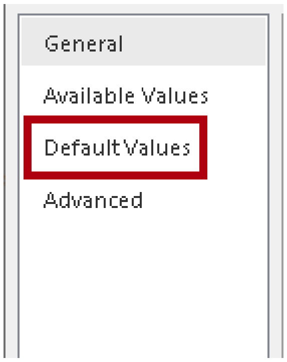


*The **SalesOrderNumber** report parameter was added automatically when the dataset was created. This is because the dataset query included the **@SalesOrderNumber** query parameter.*

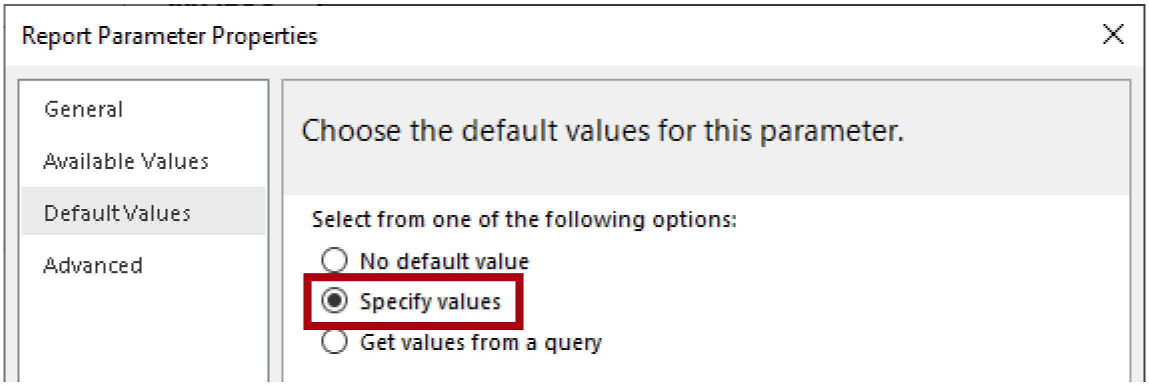
2. □ To edit the report parameter, right-click the **SalesOrderNumber** report parameter, and then select **Parameter Properties**.



3. □ In the **Report Parameter Properties** window, at the left, select the **Default Values** pages.



4. □ Select the **Specify Values** option.



Report Parameter Properties

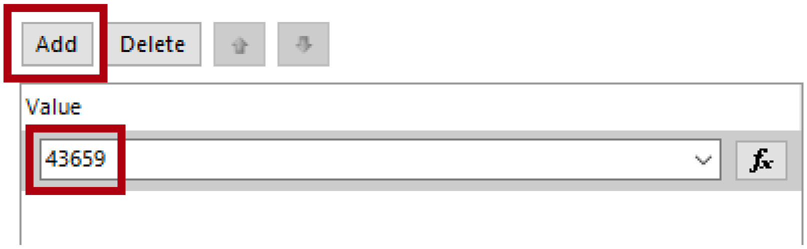
General
Available Values
Default Values
Advanced

Choose the default values for this parameter.

Select from one of the following options:

○ No default value
● Specify values
○ Get values from a query

5. □ To add a default value, click **Add**.
6. □ In the **Value** dropdown list, replace the text with **43659**.



Add | Delete

Value

43659

Sales order 43659 is the value you will initially use to test the report design.

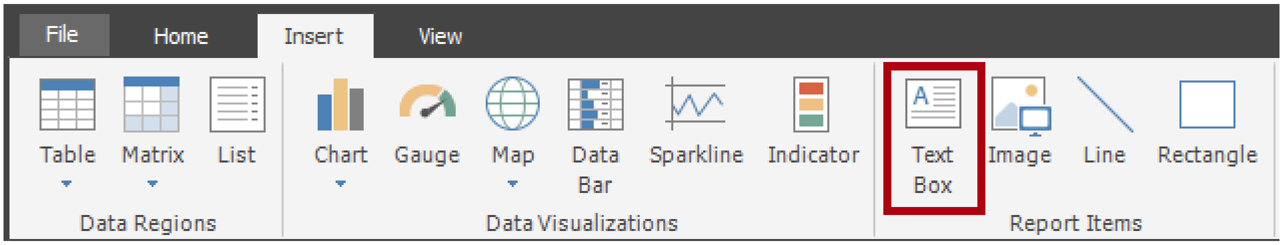
7. □ Click **OK**.
8. □ Save the report.

You will now complete the report header region design by adding textboxes to describe the sales order.

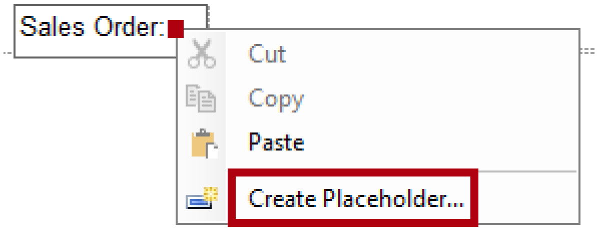
Task 4: Finalize the report header layout

In this task you will finalize the report header region design by adding textboxes.

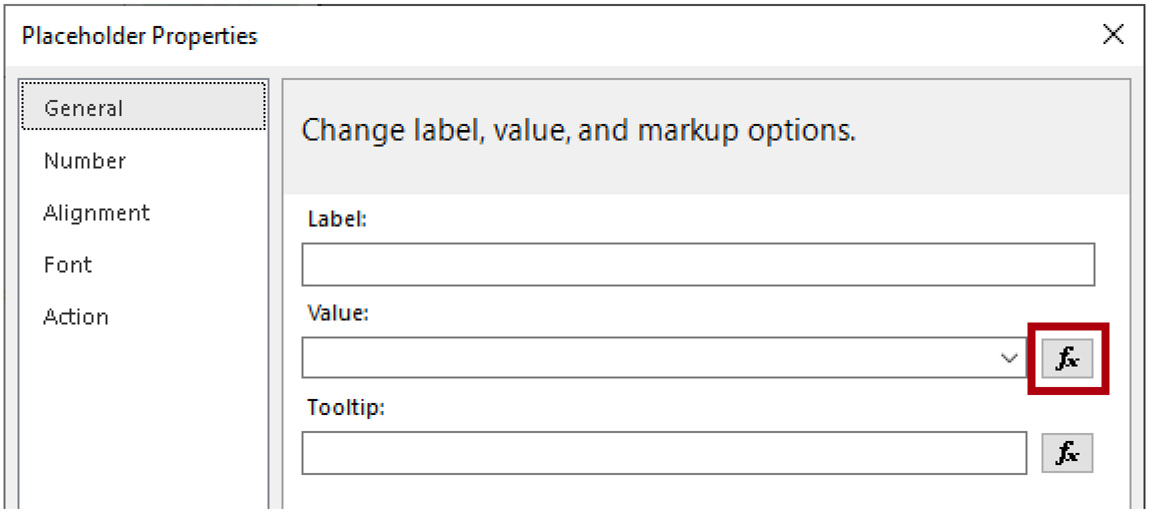
1. □ To add a textbox to the report header region, on the **Insert** ribbon tab, from inside the **Report Items** group, click **Text Box**.



2. □ Click inside the report header region, directly beneath the report title textbox.
3. □ Inside the textbox, enter **Sales Order:** followed by a space.
4. □ To insert a place holder, immediately after the space just entered, right-click and then select **Create Placeholder**.

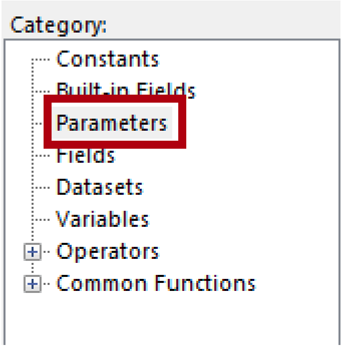


5. □ In the **Placeholder Properties** window, at the right of the **Value** dropdown list, click the **fx** button.

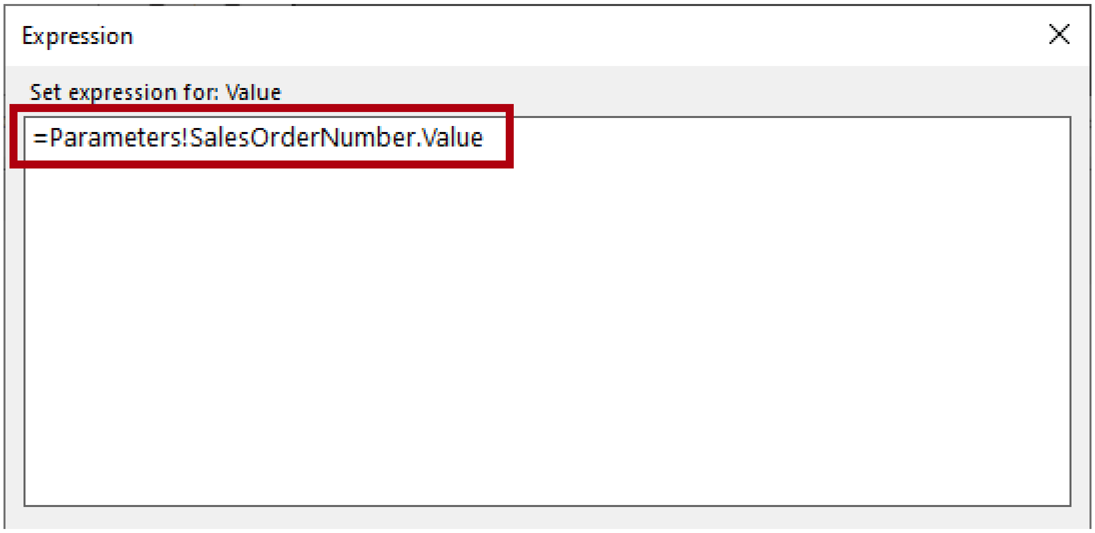


*The **fx** button allows entering a custom expression. This expression will be used to return the sales order number.*

6. □ In the **Expression** window, in the **Category** list, select **Parameters**.

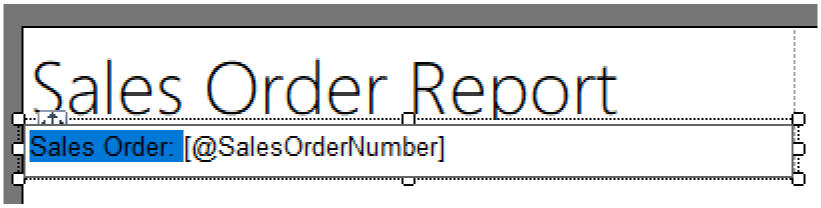


7. □ In the **Values** list, double-click the **SalesOrderNumber** parameter.
8. □ In the expression box, notice that a programmatic reference to the **SalesOrderNumber** report parameter was added.



```
=Parameters!SalesOrderNumber.Value
```

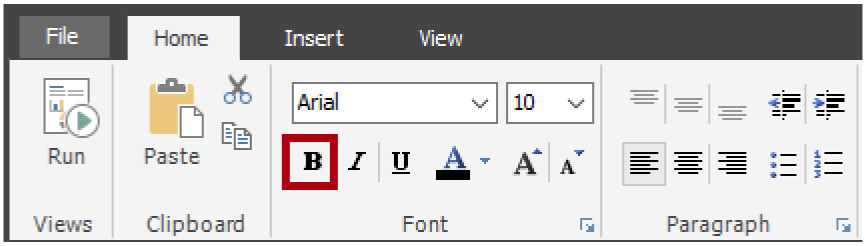
9. □ Click **OK**.
10. □ In the **Placeholder Properties** window, click **OK**.
11. □ Click a blank area of the report header region, and then select the new textbox.
12. □ In the **Properties** pane, configure the following position properties:
    - **Position | Location | Left** : 0
    - **Position | Location | Top** : 0.5
    - **Position | Size | Width** : 4
    - **Position | Size | Height** : 0.25
13. □ To format part of the textbox text, inside the new textbox, select only the **Sales Order:** text.



Sales Order Report

Sales Order: [@SalesOrderNumber]

14. □ On the **Home** ribbon tab, from inside the **Font** group, click the **Bold** command.

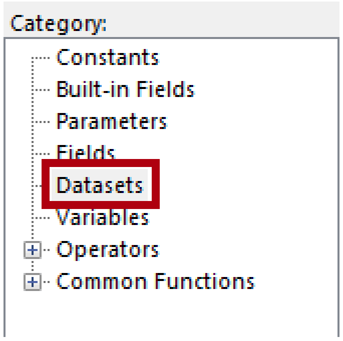


15. □ Add another textbox to the report header region, and then enter the text **Reseller:** followed by a space.

*Tip: You can also add a textbox by right-clicking the canvas, and then selected **Insert | Text Box**.*

16. □ After the space, insert a placeholder, and then set the value of the placeholder to use an expression.

17. □ In the **Expression** window, in the **Category** list, select **Datasets**.

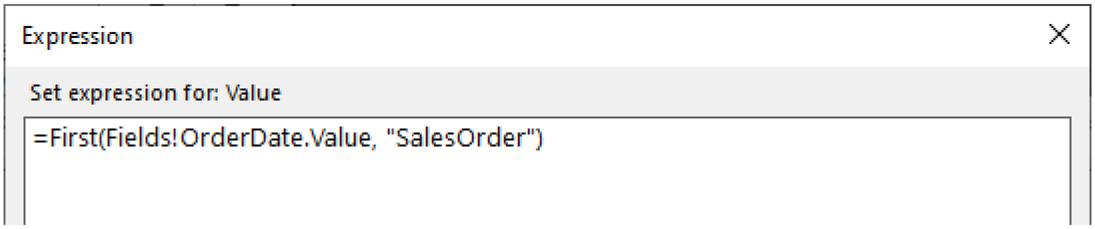


18. □ Base the expression value on **First(Reseller)** value then select **OK** in the Placeholder Properties box.

19. □ In the **Properties** pane, configure the following position properties:
    - **Position | Location | Left** : 0
    - **Position | Location | Top** : 0.75
    - **Position | Size | Width** : 4
    - **Position | Size | Height** : 0.25

20. □ Format the **Reseller:** text in bold.

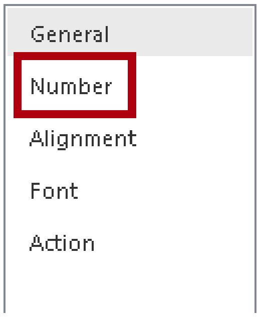
21. □ Add a third (and last) textbox to the report header region, and then enter the text **Order Date:** followed by a space.
22. □ After the space, insert a placeholder, and set the value of the placeholder to use an expression based on the **Datasets** category, **First(OrderDate)** value.



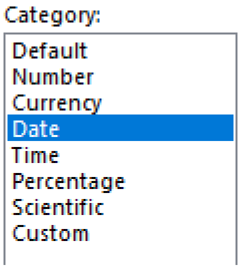
Set expression for: Value

```
=First(Fields!OrderDate.Value, "SalesOrder")
```

23. □ To format the date value, in the **Placeholder Properties** window, select the **Number** page.



24. □ In the **Category** list, select **Date**.

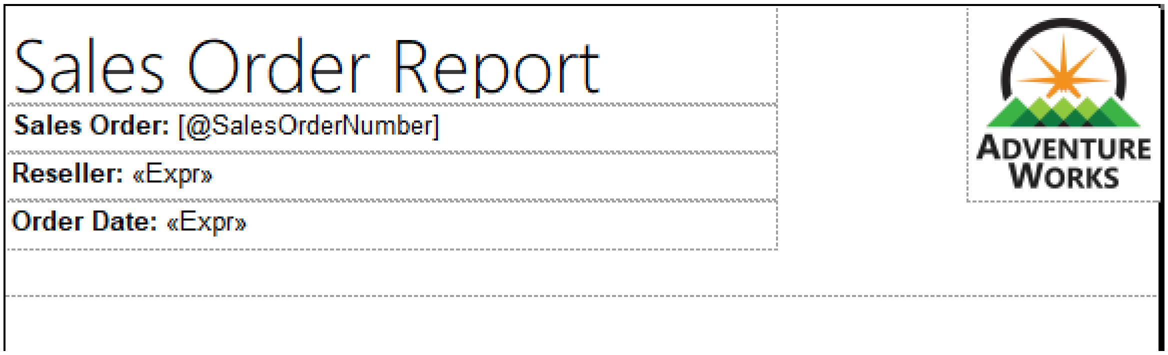


25. □ In the **Type** list, select a suitable date format type.
26. □ In the **Placeholder Properties** window , click **OK**.
27. □ In the **Properties** pane, configure the following position properties:
    - **Position | Location | Left** : 0
    - **Position | Location | Top** : 1
    - **Position | Size | Width** : 4
    - **Position | Size | Height** : 0.25
28. □ Format the **Order Date:** text in bold.

29. □ Finally, click a blank area of the report header region.

30. □ In the **Properties** pane, set the **Height** property to **1.5**.

31. □ Verify that the report header region looks like the following:



Sales Order Report

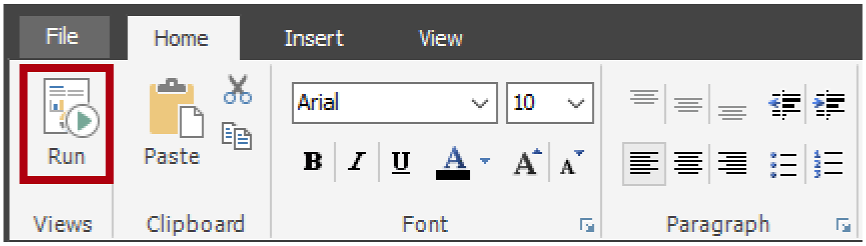
Sales Order: [@SalesOrderNumber]

Reseller: «Expr»

Order Date: «Expr»

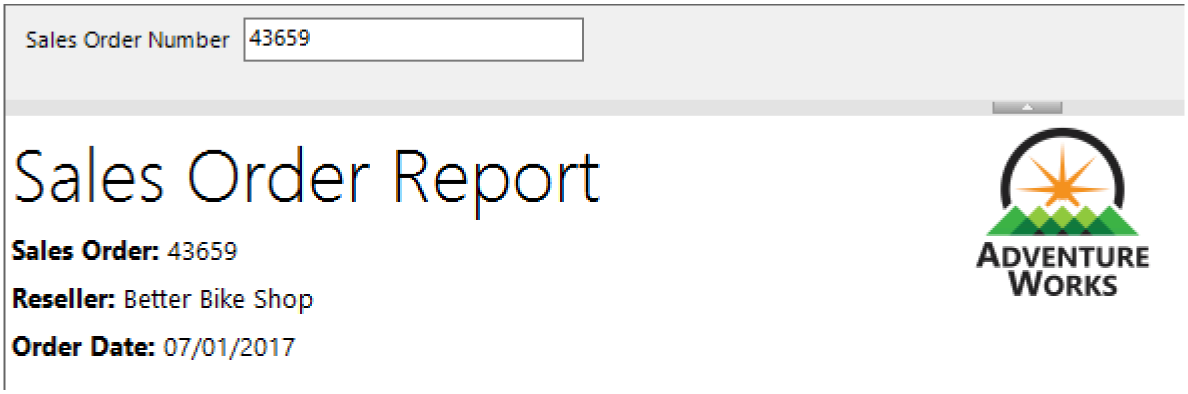
32. □ Save the report.

33. □ To preview the report, on the **Home** ribbon tab, from inside the **Views** group, click **Run**.



Running the report renders the report in HTML. As the only report parameter has a default value, the report will run automatically.

34. □ Verify that the rendered report looks like the following:



Sales Order Number 43659

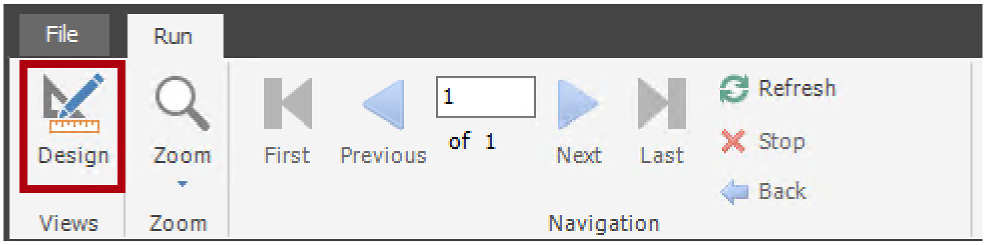
Sales Order Report

Sales Order: 43659

Reseller: Better Bike Shop

Order Date: 07/01/2017

35. □ To return to design view, on the **Run** ribbon tab, from inside the **Views** group, click **Design**.

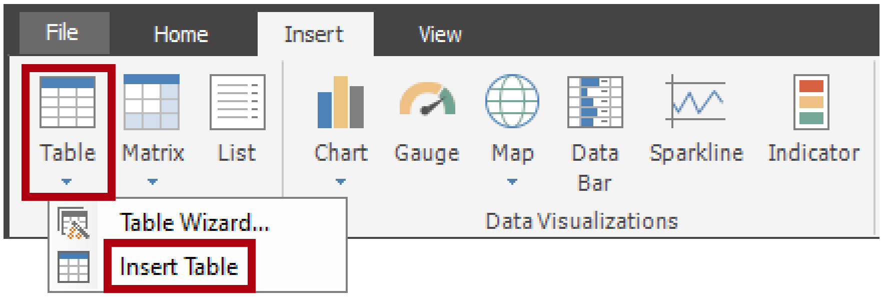


You will now add a table to the report body to display a formatted layout of the sales order lines.

Task 5: Add a table data region

In this task you will add a table data region to the report body.

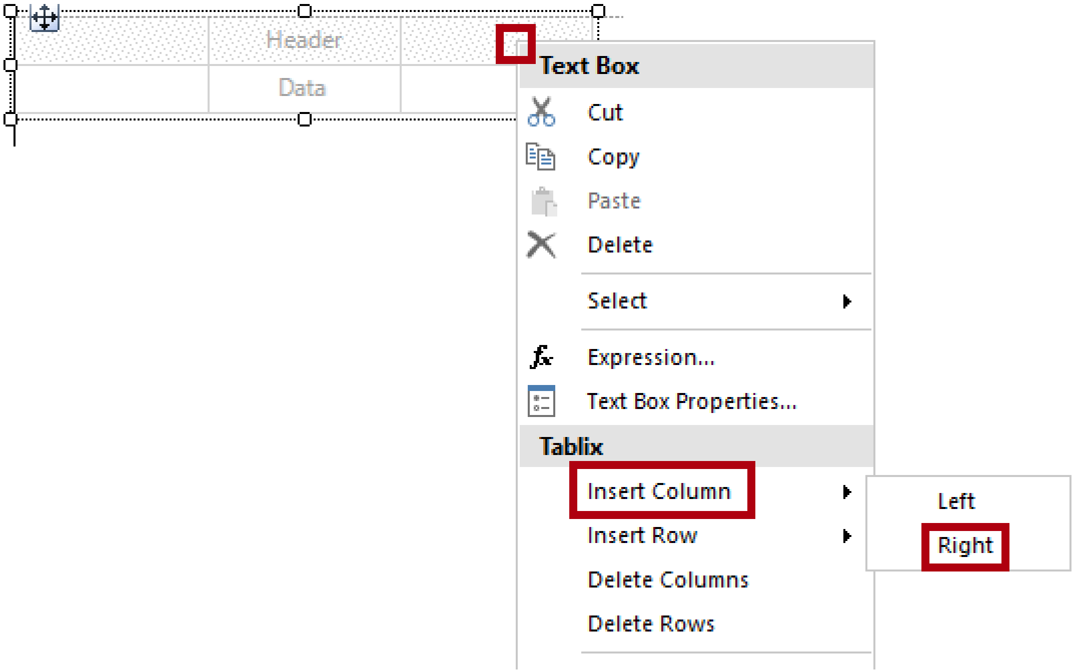
1. □ On the **Insert** ribbon tab, from inside the **Data Regions** group, click **Table**, and then select **Insert Table**.



2. □ To add the table, click a blank area inside the report body.
3. □ In the **Properties** pane, configure the following position properties:
   - ○ **Position | Location | Left** : 0
   - ○ **Position | Location | Top** : 0

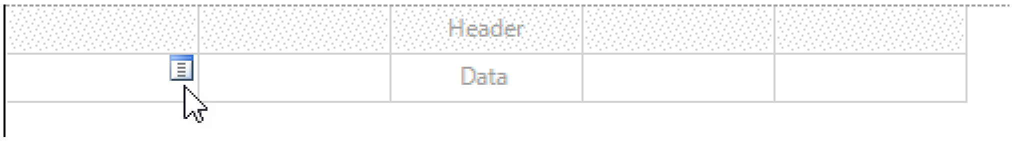
The table will display five columns. By default, the table template includes only three columns.

4. □ To add a column to the table, right-click inside any cell of the last column, and then select **Insert Column | Right**.

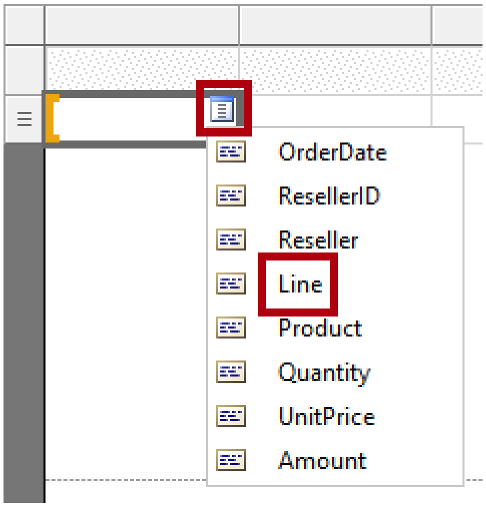


5. □ Repeat the last step to add a second new column.

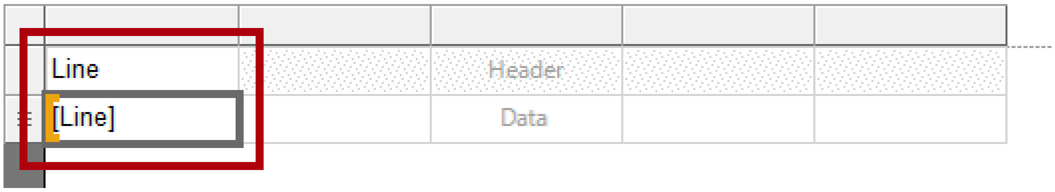
6. □ Hover the cursor over the cell in the second row of the first column to reveal the field picker icon.



7. □ Click the field picker icon, and then select the **Line** field.



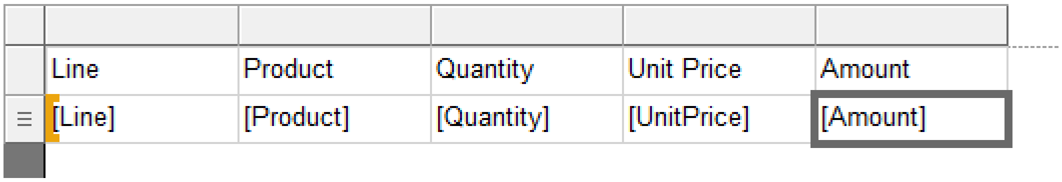
8. □ Notice that the table now includes a text value in the first row (header), and a field reference in the detail row.



Line		Header		
[Line]		Data		

9. □ Add fields to the next four columns, in order, as follows:
   - Product
   - Quantity
   - UnitPrice
   - Amount

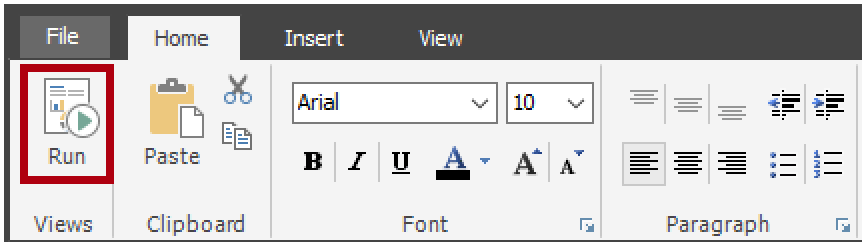
10. □ Verify that the table design looks like the following:

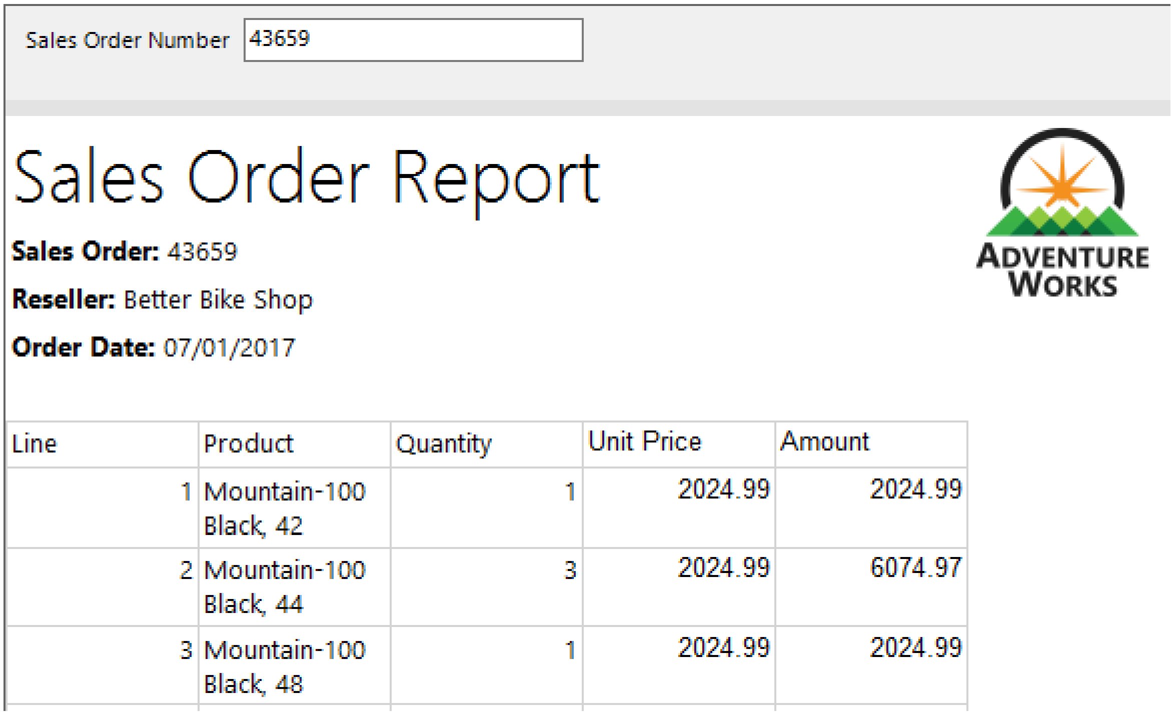


Line	Product	Quantity	Unit Price	Amount
[Line]	[Product]	[Quantity]	[UnitPrice]	[Amount]

11. □ Save the report.

12. □ Preview the report.





Sales Order Number 43659

Sales Order Report

Sales Order: 43659

Reseller: Better Bike Shop

Order Date: 07/01/2017

Line	Product	Quantity	Unit Price	Amount
1	Mountain-100 Black, 42	1	2024.99	2024.99
2	Mountain-100 Black, 44	3	2024.99	6074.97
3	Mountain-100 Black, 48	1	2024.99	2024.99

The table includes a header and 12 sales order line rows. There are many improvements that can be made by formatting the table layout.

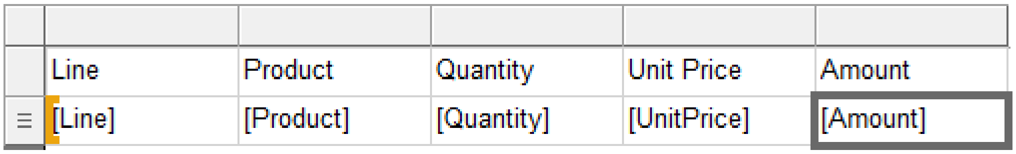
In the next task you will:

- Format the table header by using a background color and bold font style
- Modify column widths to remove redundant space and to prevent long text values from wrapping
- Left-justify the first column values
- Right-justify the last three column values
- Format currency values using a currency symbol (for USD)
- Add and format a total row for the table

Task 6: Format the table data region

In this task you will format the table data region.

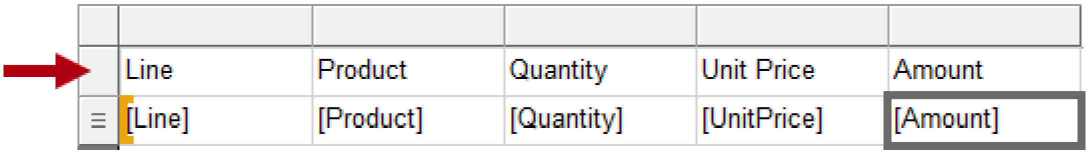
1. ☐ Return to design view.
2. ☐ Click any cell in the table to reveal the gray cell guides.



Line	Product	Quantity	Unit Price	Amount
[Line]	[Product]	[Quantity]	[UnitPrice]	[Amount]

The cell guides are there to help you configure entire rows or columns.

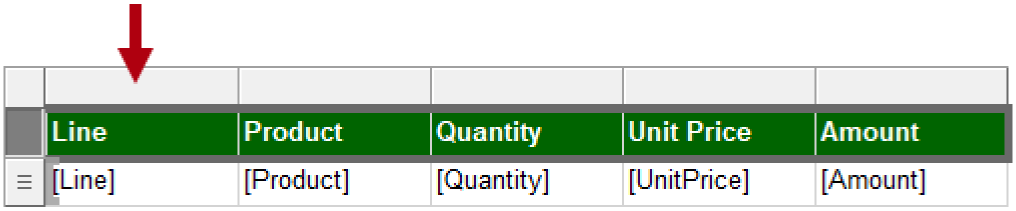
3. □ To format the table header, click the header row guide.



Line	Product	Quantity	Unit Price	Amount
[Line]	[Product]	[Quantity]	[UnitPrice]	[Amount]

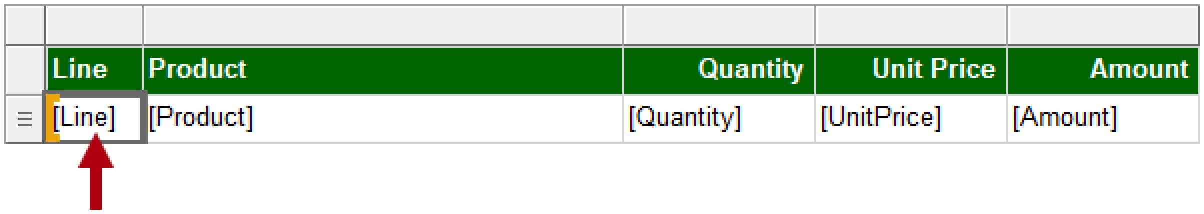
*Selecting a row or a column guide selects all cells in the row or column. Each cell is in fact a textbox. Formatting single textbox—or a multi-selection of textboxes—can then be achieved by using the **Properties** pane, or the ribbon commands.*

4. □ In the **Properties** pane (or the ribbon), configure the following properties:
   - **Fill | BackgroundColor** : DarkGreen (tip: hover the cursor over each color to reveal its name)
   - **Font | Color** : White
   - **Font | Font | FontWeight** : Bold
5. □ Select the first column guide.



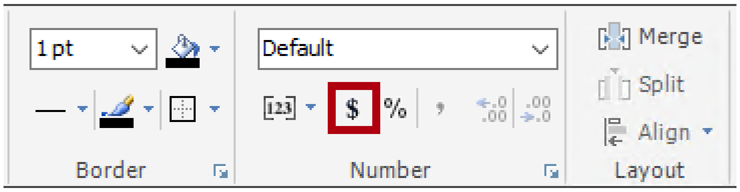
Line	Product	Quantity	Unit Price	Amount
[Line]	[Product]	[Quantity]	[UnitPrice]	[Amount]

6. □ In the **Properties** pane, set the **Position | Size | Width** property to **0.5**.
7. □ Set the width of the second column to **2.5**.
8. □ While pressing the **Ctrl** key, multi-select the last three column header textboxes (**Quantity**, **Unit Price** and **Amount**).
9. □ In the **Properties** pane (or ribbon), set the **Alignment | TextAlign** property to **Right**.
10. □ Set the **Line** detail textbox to left align.

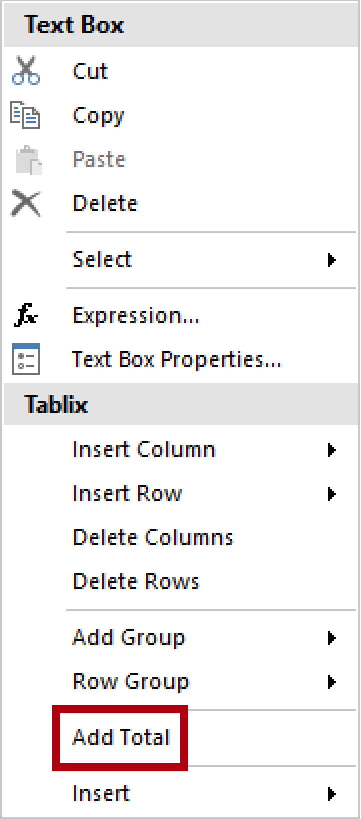


Line	Product	Quantity	Unit Price	Amount
[Line]	[Product]	[Quantity]	[UnitPrice]	[Amount]

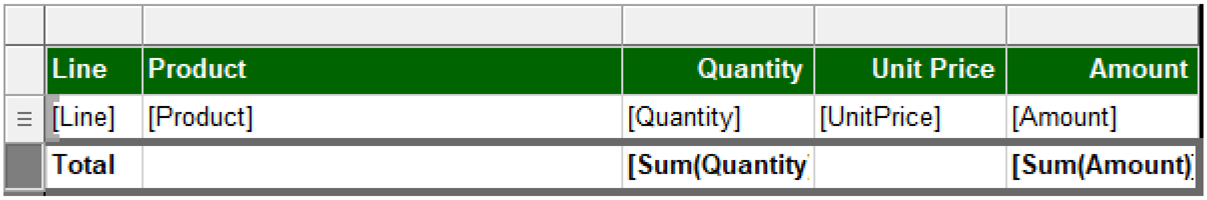
11. □ On the **Home** ribbon tab, from inside the **Number** group, set the last two detail (not header) textboxes (**UnitPrice** and **Amount**) to format with a currency symbol.



12. □ To add a total row to the table, right-click the **Quantity** detail textbox, and then select **Add Total**.

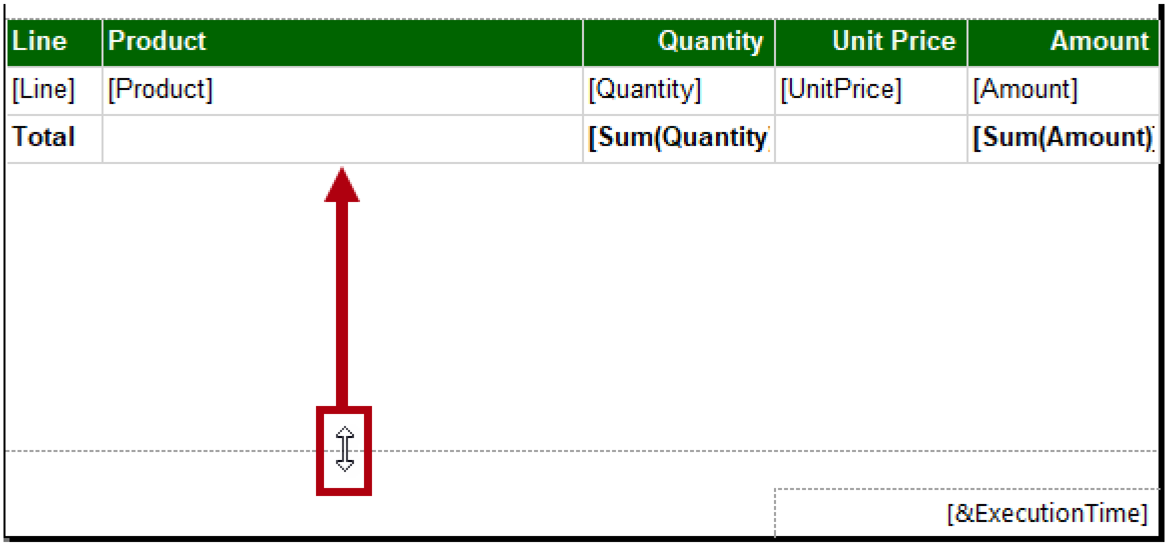


13. □ Notice that a new row, which represents the table footer, has been added, and that the expression will evaluate the sum of **Quantity** values.
14. □ Repeat the last step to add a total for the **Amount** detail textbox.
15. □ In the first cell of the table footer row, enter the word **Total**.
16. □ Format all textboxes in the footer row to format as bold.
17. □ Verify that the table design looks like the following:



Line	Product	Quantity	Unit Price	Amount
[Line]	[Product]	[Quantity]	[UnitPrice]	[Amount]
Total		[Sum(Quantity		[Sum(Amount)

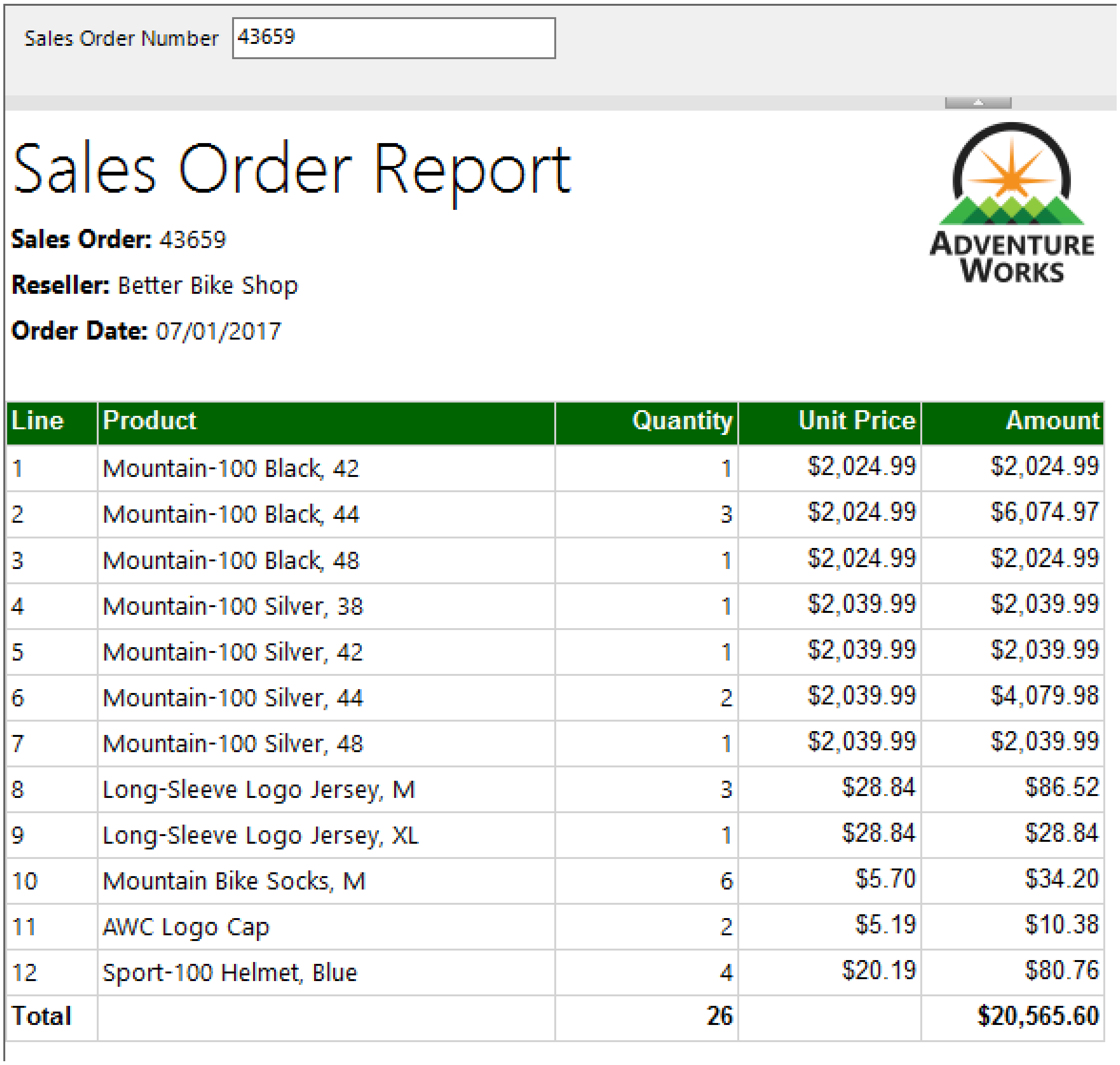
18. □ To remove any trailing space after the table, hover the cursor over the dashed line between the report body and report footer region, and then drag upwards to touch the bottom of the table.



Line	Product	Quantity	Unit Price	Amount
[Line]	[Product]	[Quantity]	[UnitPrice]	[Amount]
Total		[Sum(Quantity		[Sum(Amount)

[&ExecutionTime]

19. □ Save the report
20. □ Preview the report.
21. □ Verify that the rendered report looks like the following:



Sales Order Number 43659

Sales Order Report

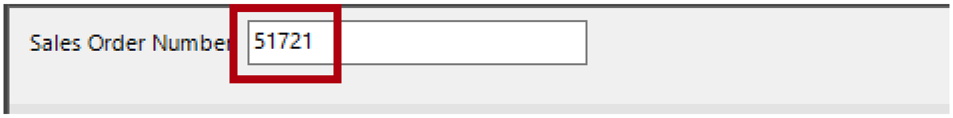
Sales Order: 43659

Reseller: Better Bike Shop

Order Date: 07/01/2017

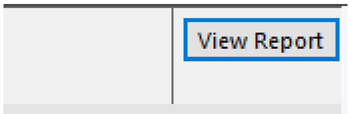
Line	Product	Quantity	Unit Price	Amount
1	Mountain-100 Black, 42	1	$2,024.99	$2,024.99
2	Mountain-100 Black, 44	3	$2,024.99	$6,074.97
3	Mountain-100 Black, 48	1	$2,024.99	$2,024.99
4	Mountain-100 Silver, 38	1	$2,039.99	$2,039.99
5	Mountain-100 Silver, 42	1	$2,039.99	$2,039.99
6	Mountain-100 Silver, 44	2	$2,039.99	$4,079.98
7	Mountain-100 Silver, 48	1	$2,039.99	$2,039.99
8	Long-Sleeve Logo Jersey, M	3	$28.84	$86.52
9	Long-Sleeve Logo Jersey, XL	1	$28.84	$28.84
10	Mountain Bike Socks, M	6	$5.70	$34.20
11	AWC Logo Cap	2	$5.19	$10.38
12	Sport-100 Helmet, Blue	4	$20.19	$80.76
Total		**26**		**$20,565.60**

22. □ In the **Sales Order Number** parameter box, replace the value with **51721**.



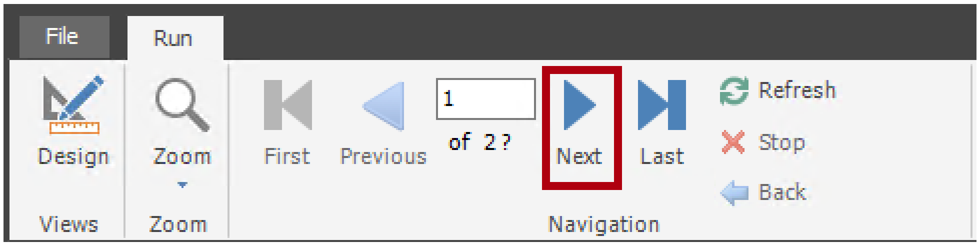
Sales Order Number 51721

23. □ To re-run the report, at the right, click **View Report**.



This sales order has 72 sales order lines, and so the data will render over many pages.

24. □ To navigate to the second page of the report, on the **Run** ribbon tab, from inside the **Navigation** group, click **Next**.



25. □ On page 2, notice that the table header does not appear.

You will address this issue in the next task.

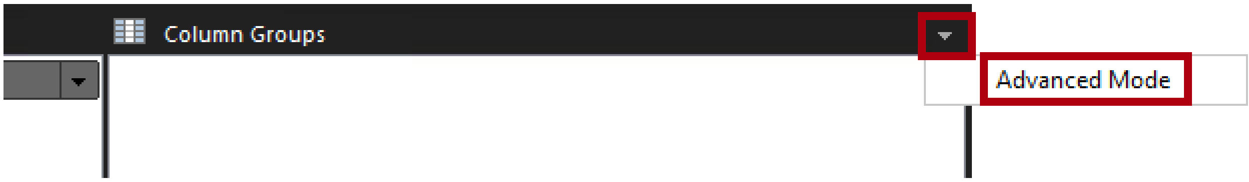
26. □ Scroll to the bottom of the page, and then notice that the report footer displays only the execution time.

In the next task you will improve the footer text by appending the page number.

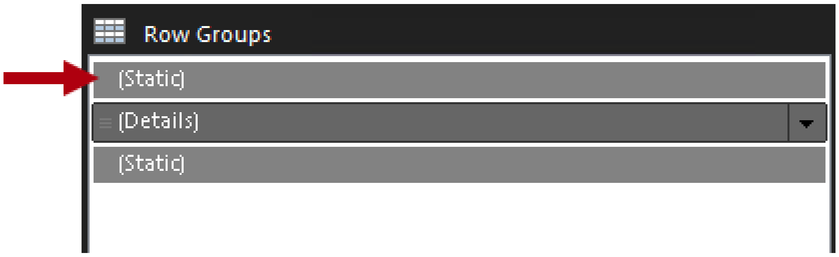
Task 7: Finalize the report design

In this task you will finalize the report design by ensuring multi-page reports render appropriately.

1. □ Switch to the design view.
2. □ To ensure the table header repeats on all pages, first select any textbox of the table.
3. □ In the **Grouping** pane (located along the bottom of the report designer), at the far right of the **Column Groups**, click the down-arrow, and then select **Advanced Mode**.



4. □ In the **Row Groups** section, select the first static group.

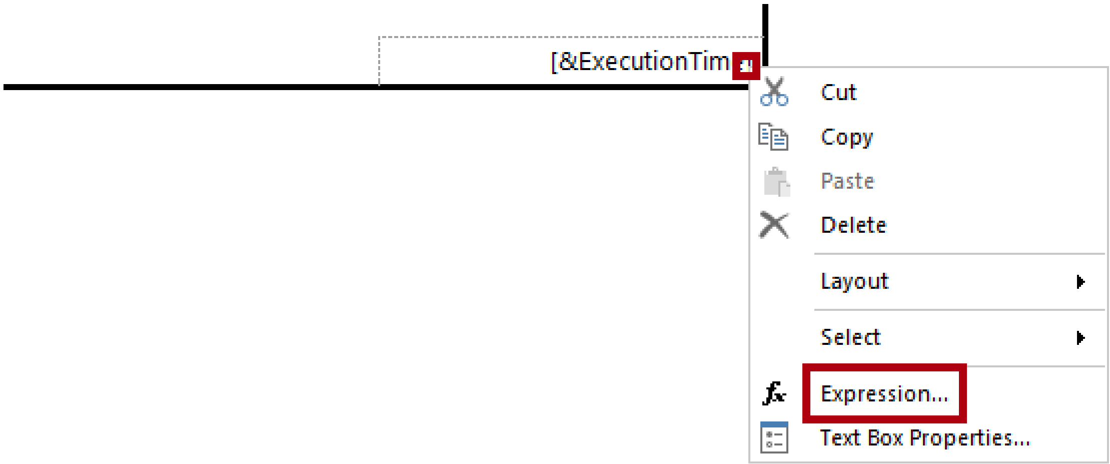


This selected the table header row.

5. □ In the **Properties** pane, set the **Other | RepeatOnNewPage** property to **True**.

   This ensures that the first static group (representing the table header) will repeat on all pages.

6. □ In the table footer region, right-click the **ExecutionTime** textbox, and then select **Expression**.

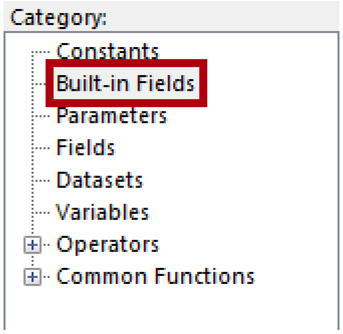


7. □ In the **Expression** window, in the expression box, append a space, followed by **& " | Page " &**, to produce the following:

   VB Script

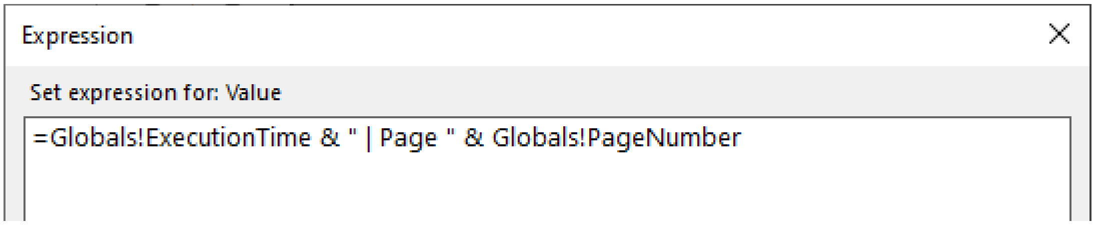
```
=Globals!ExecutionTime & " | Page " &
```

8. □ Ensure that a space follows the last ampersand (&).
9. □ In the **Category** list, select **Built-in Fields**.



10. ☐ To inject the page number value into the expression, in the **Item** list, double-click **PageNumber**.

11. ☐ Verify that the complete expression reads as follows:

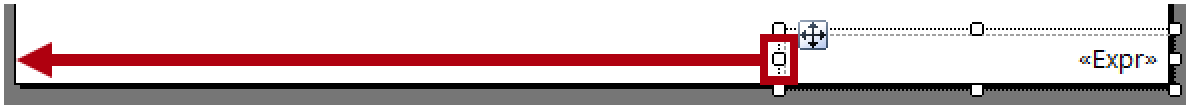


Set expression for: Value

```
=Globals!ExecutionTime & " | Page " & Globals!PageNumber
```

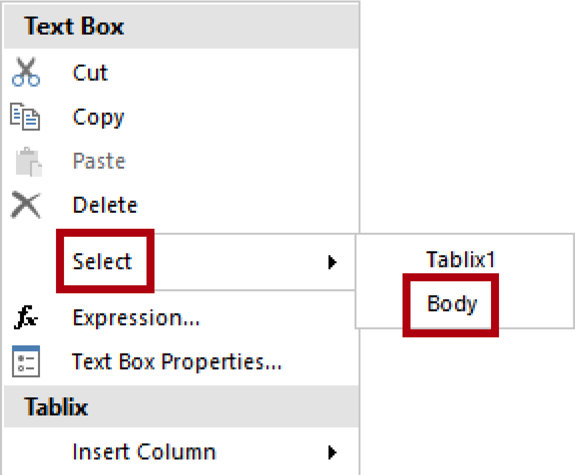
12. ☐ Click **OK**.

13. ☐ Drag the left side of the textbox to increase the width to the width of the report page.



The design of the report is now complete. Lastly, you will ensure that the page width is set to exactly six inches, and also remove the report parameter default value.

14. ☐ To select the report body, right-click any table textbox, and then select **Select | Body**.



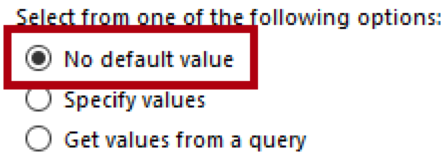
As the table fills the entire report body, this technique must be used to select the report body.

15. ☐ In the **Properties** pane, ensure that the **Position | Size | Width** property is set to **6**.

It is important the width is not greater than six inches, as rendering to print format would break the table up across multiple pages.

16. □ In the **Report Data** pane, open the **SalesOrderNumber** report parameter properties.

17. □ On the **Default Values** page, select the **No Default Value** option.



Select from one of the following options:

● No default value

○ Specify values

○ Get values from a query

18. □ Click **OK**.

19. □ Save the report.

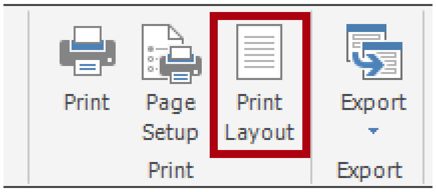
Task 8: Explore the final report

In this task you will view the report in print layout mode.

1. □ Preview the report.

2. □ In the **Sales Order Number** parameter box, enter the value with **51721**

3. □ On the **Run** ribbon tab, from inside the **Print** group, click **Print Layout**.



Print layout mode provides a preview of what the report will look like when printed to the strict page size.

4. □ Navigate to page 2.

In this lab you won't publish the report. Paginated reports can only be rendered in the Power BI service when they are stored in a workspace on dedicated capacity, and when that capacity has the paginated reports workload enabled. These requirements do not exist for the class.

Congratulations!

Perform Data Analysis in Power BI Desktop

In this lab you will create the **Sales Exploration** report.

In this lab you learn how to:

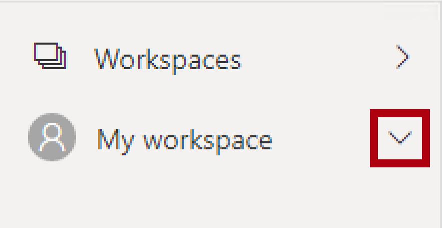
- Create animated scatter charts
- Use a visual to forecast values
- Work with the decomposition tree visual
- Work with the key influencers visual

Exercise 1: Create the Report

In this exercise you will create the **Sales Exploration** report.

Task 1: Get started – Sign in

1. □ In the Microsoft Edge browser window, in the Power BI service, in the **Navigation** pane, expand **My Workspace**.

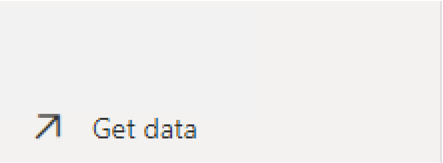


Task 2: Get started – Create a dataset

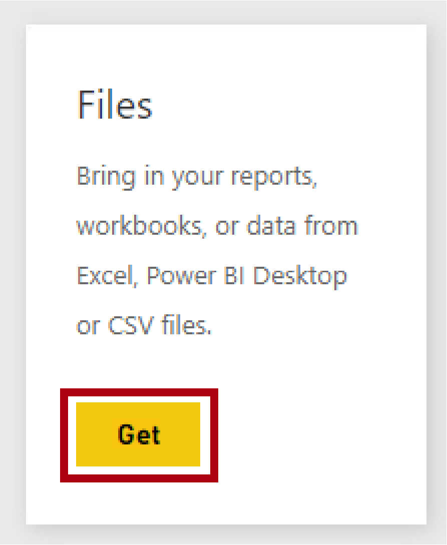
In this task you will setup the environment for the lab by creating a dataset.

Important: If you have already published the dataset in the ***Create a Power BI Dashboard*** *lab, continue from the next task.*

1. □ In the Microsoft Edge browser window, in the Power BI service, in the **Navigation** pane, at the bottom, click **Get Data**.



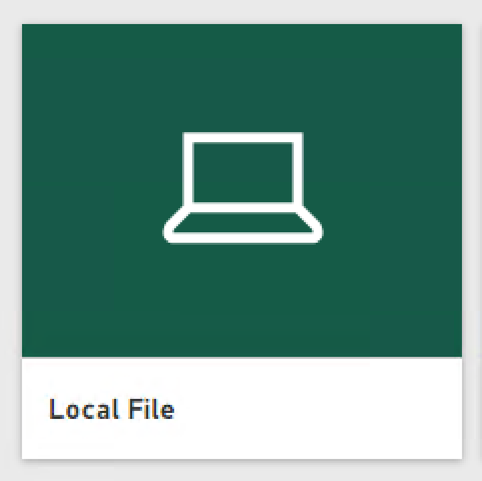
2. □ In the **Files** tile, click **Get**.



Files

Bring in your reports, workbooks, or data from Excel, Power BI Desktop or CSV files.

3. □ Click the **Local File** tile.



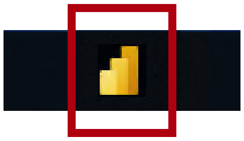
4. □ In the **Open** window, navigate to the **D:\DA100\Labs\09-create-power-bi-dashboard\Solution** folder.
5. □ Select the **Sales Analysis.pbix** file, and then click **Open**.
6. □ If prompted to replace the dataset, click **Replace**.

Task 3: Create the report

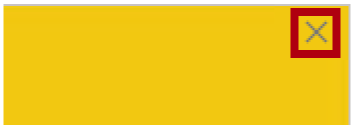
In this task you will create the **Sales Exploration** report.

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

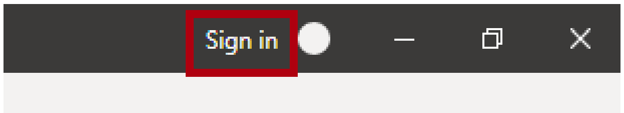
   Important: If you already have Power BI Desktop open (from a previous lab), close that instance.



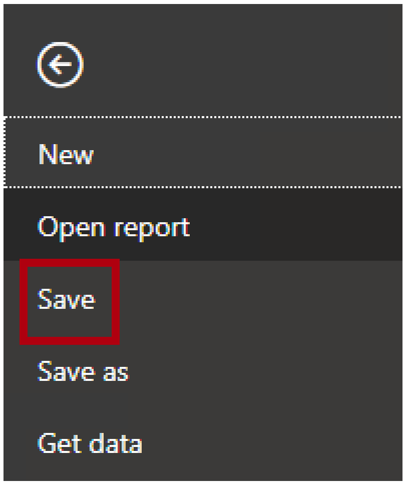
2. □ To close the getting started window, at the top-left of the window, click **X**.



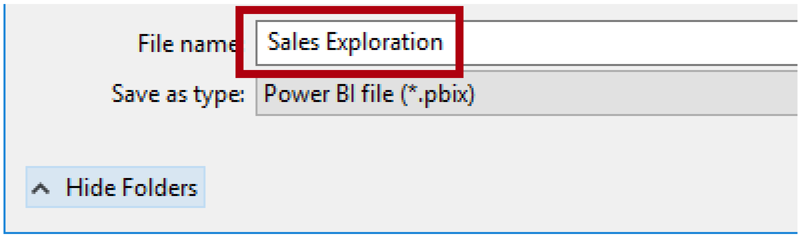
3. □ If Power BI Desktop is not signed in to the Power BI service, at the top-right, click **Sign In**.



4. □ Complete the sign in process using the same account used to sign in to the Power BI service.
5. □ To save the file, click the **File** ribbon tab to open the backstage view.
6. □ Select **Save**.



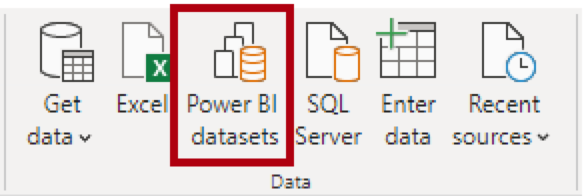
7. □ In the **Save As** window, navigate to the **D:\DA100\MySolution** folder.
8. □ In the **File Name** box, enter **Sales Exploration**.



File name: Sales Exploration

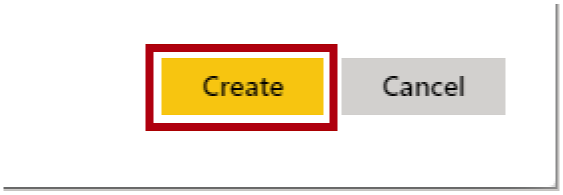
Save as type: Power BI file (*.pbix)

9. □ To create a live connection to the **Sales Analysis** dataset, on the **Home** ribbon tab, from inside the **Data** group, click **Power BI Datasets**.



10. □ In the **Select a Dataset to Create a Report** window, select the **Sales Analysis** dataset.

11. □ Click **Create**.



12. □ Save the Power BI Desktop file.

You'll now create four report pages, and on each page you'll work with a different visual to analyze and explore data.

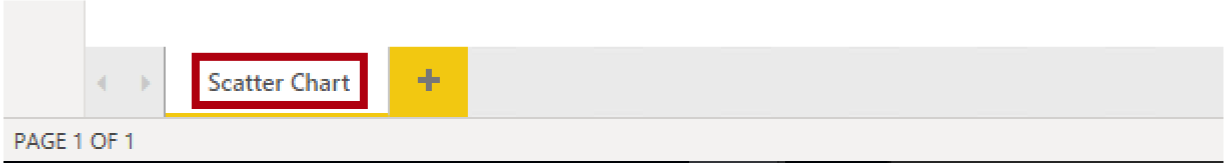
Exercise 2: Create a Scatter Chart

In this exercise you will create a scatter chart that can be animated.

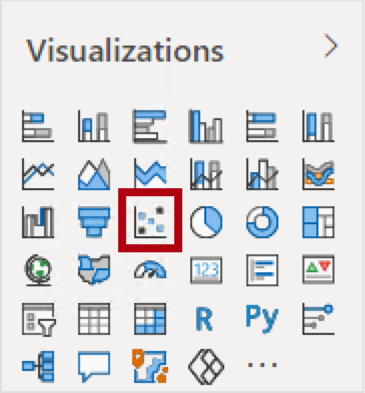
Task 1: Create an animated scatter chart

In this task you will create a scatter chart that can be animated.

1. □ Rename **Page 1** as **Scatter Chart**.



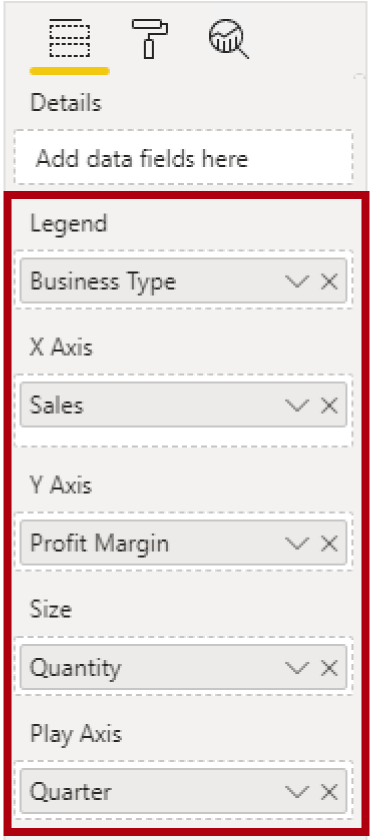
2. □ Add a **Scatter Chart** visual to the report page, and then position and resize it so it fills the entire page.



3. □ Add the following fields to the visual wells/areas:

The labs use a shorthand notation to reference a field. It will look like this: **Reseller | Business Type**. In this example, **Reseller** is the table name and **Business Type** is the field name.

- Legend: **Reseller | Business Type**
- X Axis: **Sales | Sales**
- Y Axis: **Sales | Profit Margin**
- Size: **Sales | Quantity**
- Play Axis: **Date | Quarter**



Details

Add data fields here

Legend

Business Type

X Axis

Sales

Y Axis

Profit Margin

Size

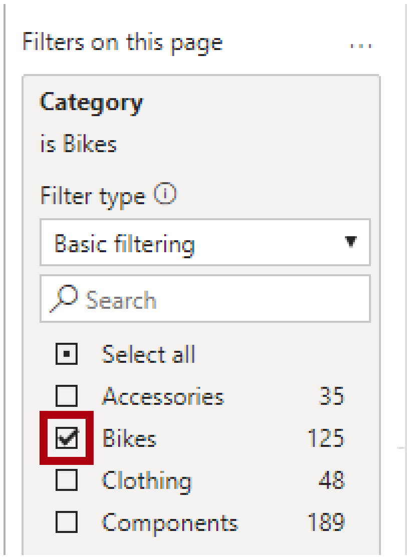
Quantity

Play Axis

Quarter

The chart can be animated when a field is added to the ***Play Axis*** *well/area.*

4. □ In the **Filters** pane, add the **Product | Category** field to the **Filters On This Page** well/area.
5. □ In the filter card, filter by **Bikes**.



Filters on this page ...

Category

is Bikes

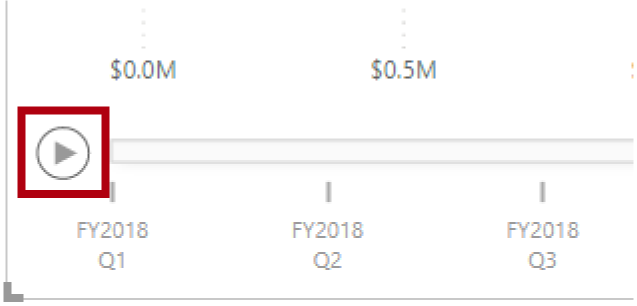
Filter type ⓘ

Basic filtering ▾

Search

⊡	Select all	
□	Accessories	35
☑	Bikes	125
□	Clothing	48
□	Components	189

6. □ To animate the chart, at the bottom left corner, click **Play**.



7. □ Watch the entire animation cycle from **FY2018 Q1** to **FY2020 Q4**.

The scatter chart allows understanding the measure values simultaneously: in this case, order quantity, sales revenue, and profit margin.

Each bubble represents a reseller business type. Changes in the bubble size reflect increased or decreased order quantities. While horizontal movements represent increases/decreases in sales revenue, and vertical movements represent increases/decreases in profitability.

8. □ When the animation stops, click one of the bubbles to reveal its tracking over time.
9. □ Hover the cursor over any bubble to reveal a tooltip describing the measure values for the reseller type at that point in time.
10. □ In the **Filters** pane, filter by **Clothing** only, and notice that it produces a very different result.
11. □ Save the Power BI Desktop file.

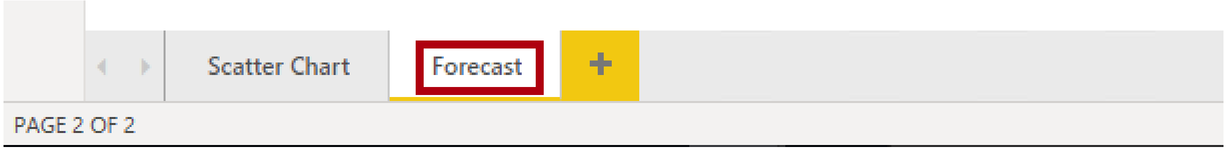
Exercise 3: Create a Forecast

In this exercise you will create a forecast to determine possible future sales revenue.

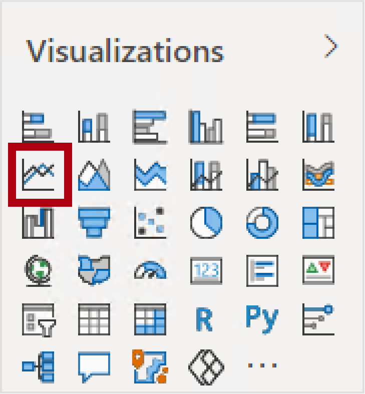
Task 1: Create a forecast

In this task you will create a forecast to determine possible future sales revenue.

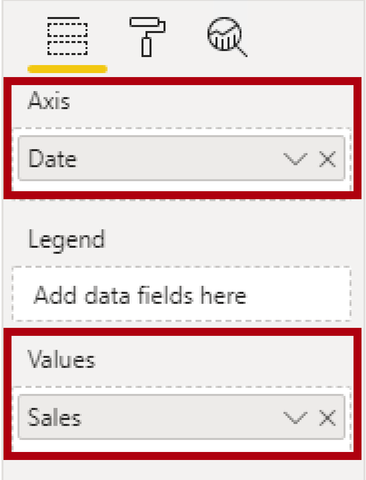
1. □ Add a new page, and then rename the page to **Forecast**.



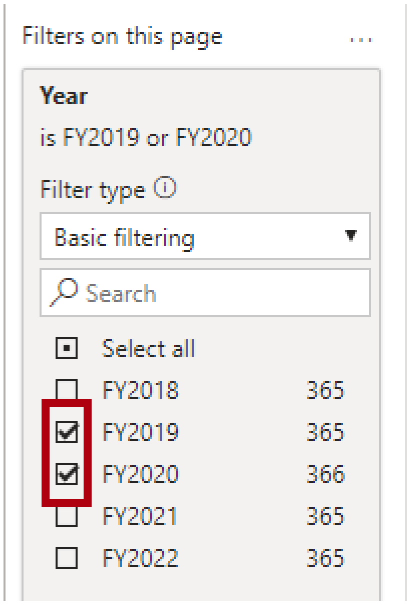
2. □ Add a **Line Chart** visual to the report page, and then position and resize it so it fills the entire page.



3. □ Add the following fields to the visual wells/areas:
   - ○ Axis: **Date | Date**
   - ○ Values: **Sales | Sales**



4. □ In the **Filters** pane, add the **Date | Year** field to the **Filters On This Page** well/area.
5. □ In the filter card, filter by two years: **FY2019** and **FY2020**.



Filters on this page ...

Year

is FY2019 or FY2020

Filter type ⓘ

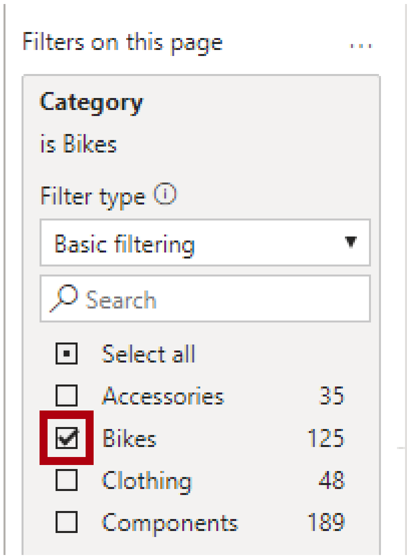
Basic filtering ▼

Search

▣	Select all	
□	FY2018	365
☑	FY2019	365
☑	FY2020	366
□	FY2021	365
□	FY2022	365

When forecasting over a time line, you will need at least two cycles (years) of data to produce an accurate and stable forecast.

6. □ Add also the **Product | Category** field to the **Filters On This Page** well/area, and filter by **Bikes**.



Filters on this page ...

Category

is Bikes

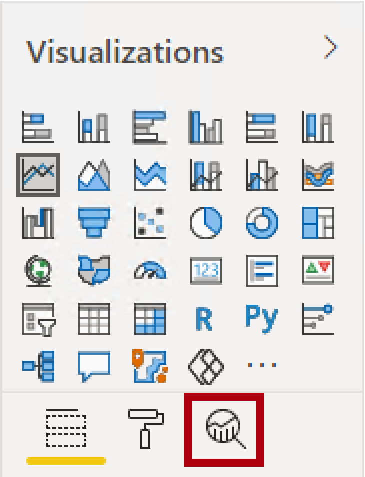
Filter type ⓘ

Basic filtering ▼

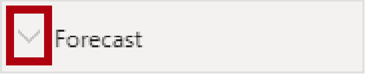
Search

▣	Select all	
□	Accessories	35
☑	Bikes	125
□	Clothing	48
□	Components	189

7. □ To add a forecast, beneath the **Visualizations** pane, select the **Analytics** pane.

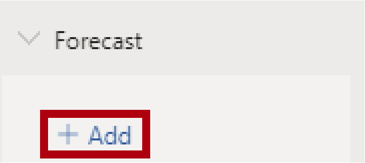


8. □ Expand the **Forecast** section.

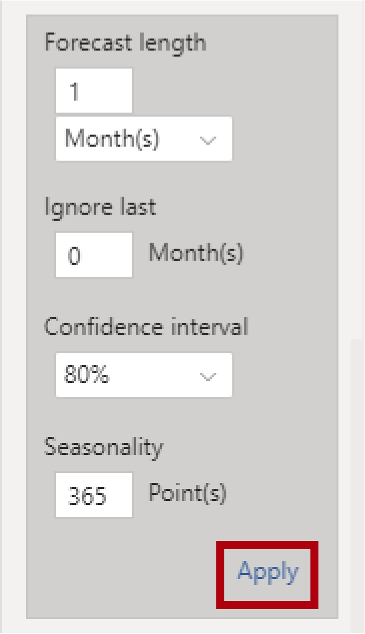


*If the **Forecast** section is not available, it's probably because the visual hasn't been correctly configured. Forecasting is only available when two conditions are met: the axis has a single field of type date, and there's only one value field.*

9. □ Click **Add**.



10. □ Configure the following forecast properties:
    - Forecast length: 1 month
    - Confidence interval: 80%
    - Seasonality: 365
11. □ Click **Apply**.



Forecast length
1
Month(s)

Ignore last
0 Month(s)

Confidence interval
80%

Seasonality
365 Point(s)

12. □ In the line visual, notice that the forecast has extended one month beyond the history data.

    The gray area represents the confidence. The wider the confidence, the less stable—and therefore the less accurate—the forecast is likely to be.

    When you know the length of the cycle, in this case annual, you should enter the seasonality points. Sometimes it could be weekly (7), or monthly (30).

13. □ In the **Filters** pane, filter by **Clothing** only, and notice that it produces a different result.
14. □ Save the Power BI Desktop file.

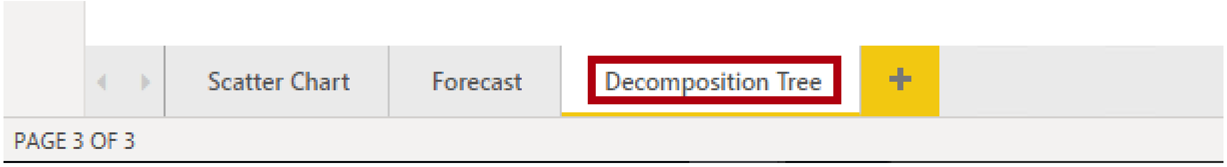
Exercise 4: Work with a Decomposition Tree

In this exercise you will create a decomposition tree to explore the relationships between reseller geography and profit margin.

Task 1: Work with a decomposition tree

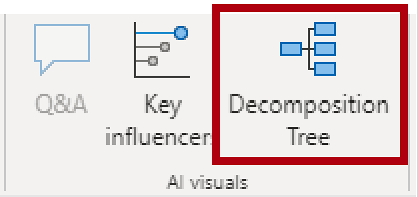
In this task you will create a decomposition tree to explore the relationships between reseller geography and profit margin.

1. □ Add a new page, and then rename the page to **Decomposition Tree**.



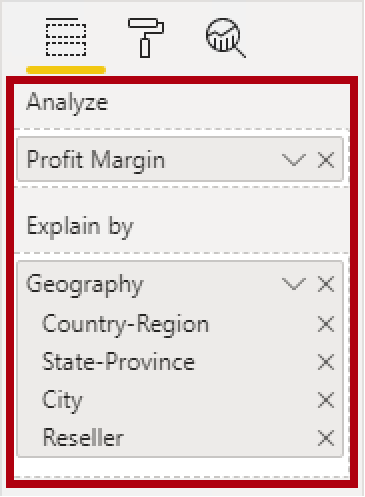
2. □ On the **Insert** ribbon, from inside the **AI Visuals** group, click **Decomposition Tree**.

*Tip: The AI visuals are also available in the **Visualizations** pane.*



3. □ Position and resize the visual so it fills the entire page.

4. □ Add the following fields to the visual wells/areas:
   - Analyze: **Sales | Profit Margin**
   - Explain By: **Reseller | Geography** (the entire hierarchy)



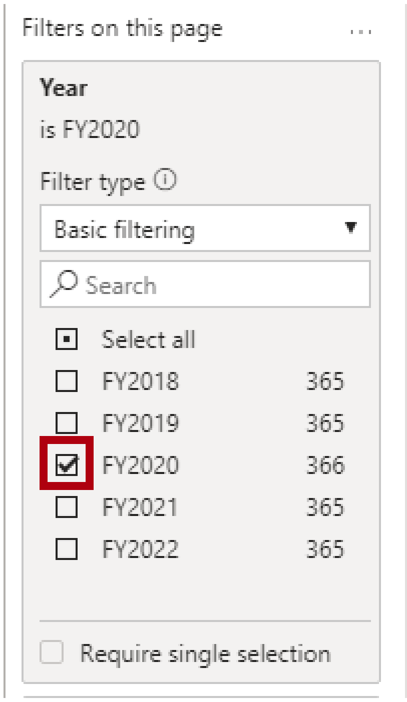
Analyze

Profit Margin

Explain by

Geography
- Country-Region
- State-Province
- City
- Reseller

5. □ In the **Filters** pane, add the **Date | Year** field to the **Filters On This Page**, and set the filter to **FY2020**.



Filters on this page ...

Year

is FY2020

Filter type ⓘ

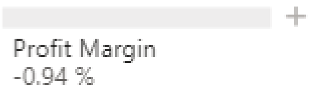
Basic filtering ▾

Search

⊡	Select all	
□	FY2018	365
□	FY2019	365
☑	FY2020	366
□	FY2021	365
□	FY2022	365

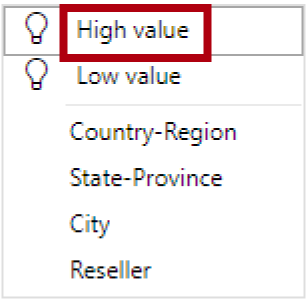
□ Require single selection

6. □ In the decomposition tree visual, notice the root of the tree: **Profit Margin** at -0.94%



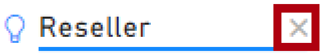
Profit Margin
-0.94 %

7. □ Click the plus icon, and in the context menu, select **High Value**.



8. □ Notice that the decomposition tree presents resellers, ordered from highest profit margin.

9. □ To remove the level, at the top of visual, beside the **Reseller** label, click **X**.



10. □ Click the plus icon again, and then expand to the **Country-Region** level.

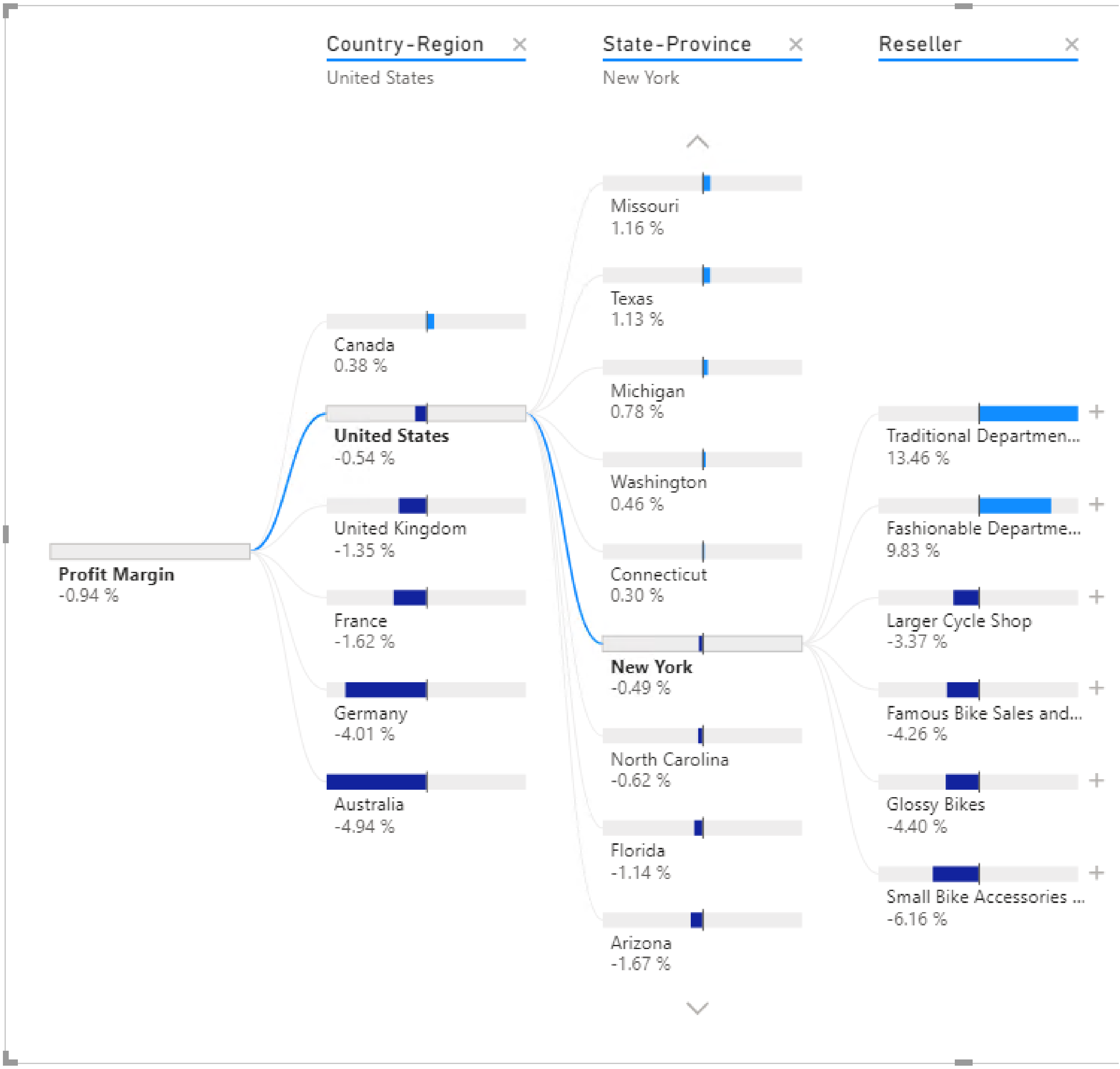
11. □ Expand from the **United States** to the **State-Province** level.

12. □ Use the down-arrow located at the bottom of the visual for **State-Province**, and then scroll to the lower profitable states.

13. □ Notice that **New York** state has negative profitability.

14. □ Expand from **New York** to the **Reseller** level.

15. □ Notice that it is easy to isolate root cause.



United States *is not producing profit in* ***FY2020***. ***New York*** *is one state that's not achieving positive profit, and it's due to four resellers paying less than standard costs for their goods.*

16. □ Save the Power BI Desktop file.

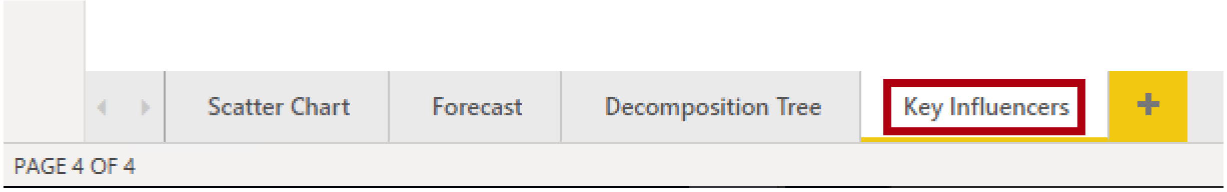
Exercise 5: Work with Key Influencers

In this exercise you will use the Key Influencers AI visual to determine what influences profitability within reseller business types and geography.

Task 1: Work with key influencers

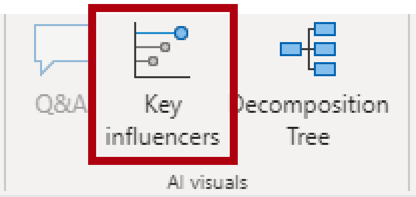
In this task you will use the Key Influencers AI visual to determine what influences profitability within reseller business types and geography.

1. □ Add a new page, and then rename the page to **Key Influencers**.



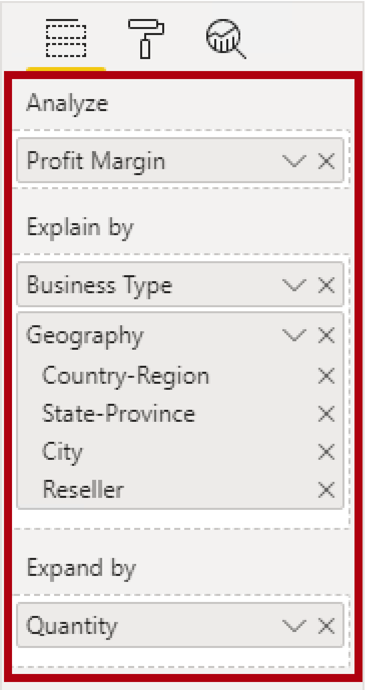
2. □ On the **Insert** ribbon, from inside the **AI Visuals** group, click **Key Influencers**.

*Tip: The AI visuals are also available in the **Visualizations** pane.*



3. □ Position and resize the visual so it fills the entire page.

4. □ Add the following fields to the visual wells:
   - Analyze: **Sales | Profit Margin**
   - Explain By: **Reseller | Business Type** and **Reseller | Geography** (the entire hierarchy)
   - Expand By: **Sales | Quantity**



Analyze

Profit Margin

Explain by

Business Type

Geography

Country-Region

State-Province

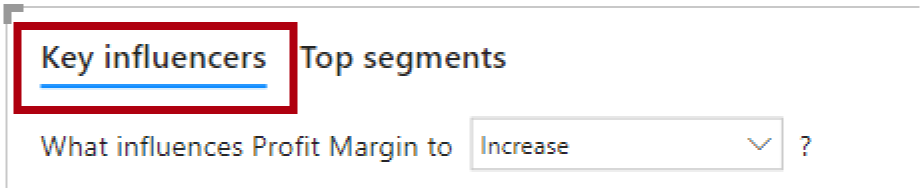
City

Reseller

Expand by

Quantity

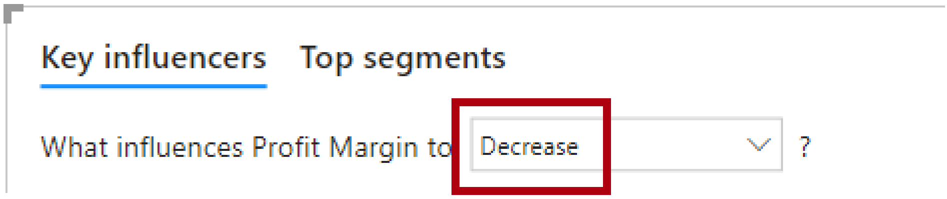
5. ☐ At the top-left of the visual, notice that **Key Influencers** is in focus, and the specific influence is set to understand what influences profit margin to increase.



Key influencers Top segments

What influences Profit Margin to Increase ?

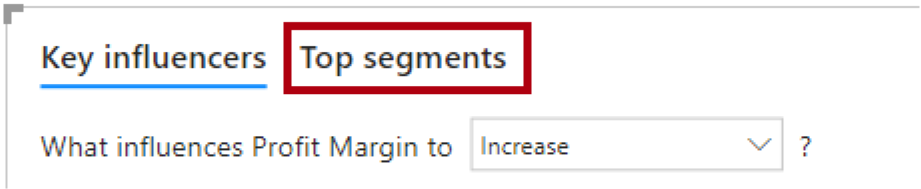
6. ☐ Review the result, which is that the city of **Bothel** is more likely to increase.
7. ☐ Modify the target to determine what influences profit margin to decrease.



Key influencers Top segments

What influences Profit Margin to Decrease ?

8. ☐ Review the result.
9. ☐ To detect segments, at the top-left, select **Top Segments**.



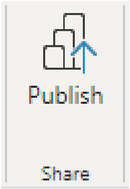
What influences Profit Margin to Increase ?

10. □ Notice that the target is now to determine segments when profit margin is likely to be high.
11. □ When the visual displays the segments (as circles), click one of them to reveal information about it.
12. □ Review the segment results.

Task 2: Finish up

In this task you will complete the lab.

1. □ Select the **Scatter Chart** page.
2. □ Save the Power BI Desktop file.
3. □ To publish the file to your workspace, on the **Home** ribbon tab, from inside the **Share** group, click **Publish**.



4. □ Close Power BI Desktop.

Congratulations!