MicrosoftLearning / PL-300-Microsoft-Power-BI-Data-Analyst Public

generated from MicrosoftLearning/INF99X-SampleCourse

Code Issues 6 Pull requests Actions Projects Security Insights

Main

PL-300-Microsoft-Power-BI-Data-Analyst / Instructions / 03-configure-data-model-in-power-bi-desktop.md

bpmoring Minor update to instructions ✓

3 contributors

645 lines (323 sloc) | 28.9 KB

lab	
title	**module**
Model Data in Power BI Desktop	Module 4 - Design a Data Model in Power BI

Model Data in Power BI Desktop

The estimated time to complete the lab is 45 minutes

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

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, 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**
4. Create DAX Calculations in Power BI Desktop, Part 1
5. Create DAX Calculations in Power BI Desktop, Part 2
6. Design a Report in Power BI Desktop, Part 1
7. Design a Report in Power BI Desktop, Part 2
8. Create a Power BI Dashboard
9. Perform Data Analysis in Power BI Desktop
10. Enforce Row-Level Security

Exercise 1: Create Model Relationships

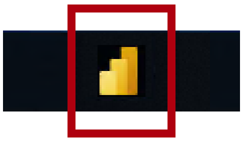
In this exercise you will create model relationships.

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.

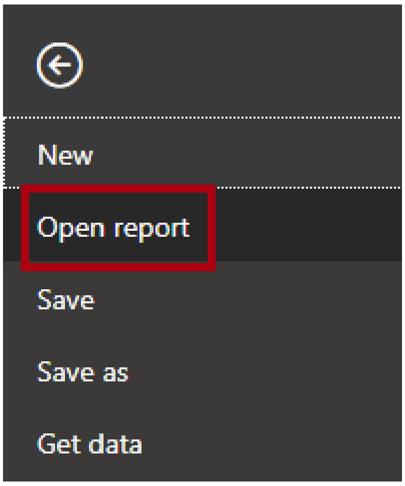
1. To open the Power BI Desktop, on the taskbar, click the Microsoft Power BI Desktop shortcut.



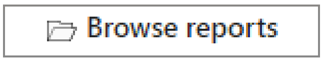
2. To close the getting started window, at the top-left of the window, click **X**.

3. To open the starter Power BI Desktop file, click the **File** ribbon tab to open the backstage view.

4. Select **Open Report**.



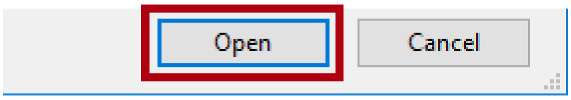
5. Click **Browse Reports**.



6. In the **Open** window, navigate to the **D:\PL300\Labs\03-configure-data-model-in-power-bi-desktop\Starter** folder.

7. Select the **Sales Analysis** file.

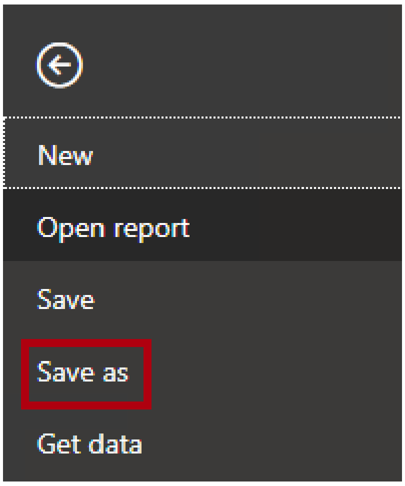
8. Click **Open**.



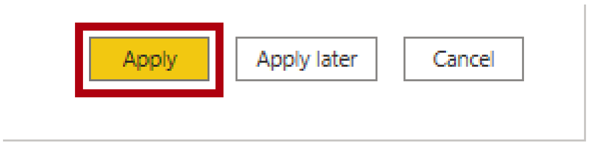
9. Close any informational windows that may open.

10. To create a copy of the file, click the **File** ribbon tab to open the backstage view.

11. Select **Save As**.

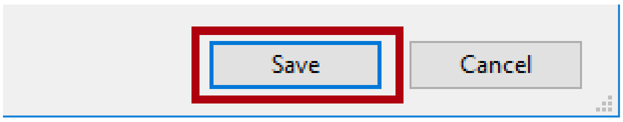


12. If prompted to apply changes, click **Apply**.



13. In the **Save As** window, navigate to the **D:\PL300\MySolution** folder.

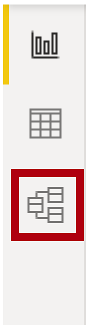
14. Click **Save**.



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.

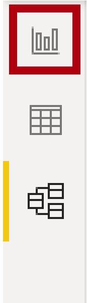


2. If you do not see all seven tables, scroll horizontally to the right, and then drag and arrange the tables more closely together so they can all be seen at the same time.

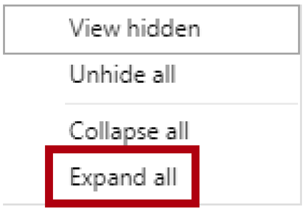
   Tip: You can also use the zoom control located at the bottom of the window.

   *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.*

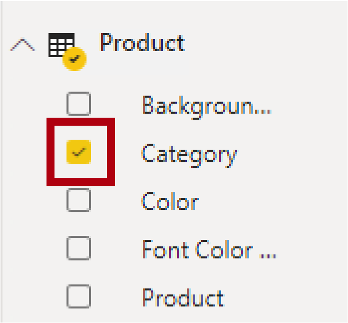
3. To return to Report view, at the left, click the **Report** view icon.



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



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



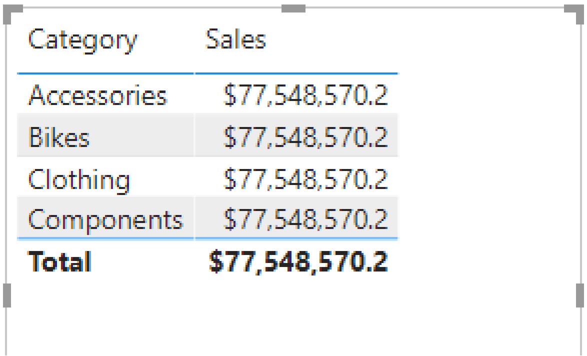
Product

☐ 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.

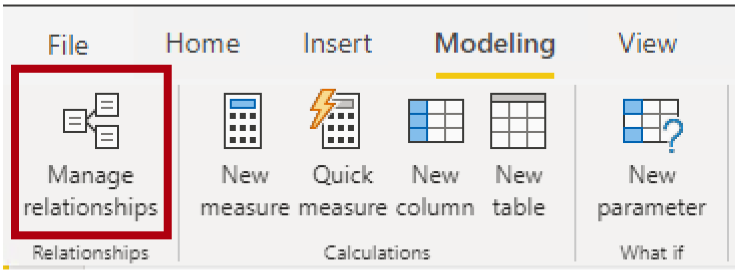
7. Notice that the table visual lists four product categories, and that the sales value is the same for each, and the same for the total.



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**

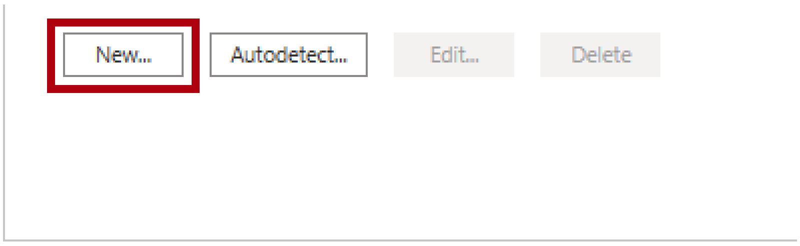
*The issue is that the table is based on fields from different tables. The expectation is that each product category displays the sales for that category. However, because there isn't a model relationship between these tables, the **Sales** table is not filtered. You'll now add a relationship to propagate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

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

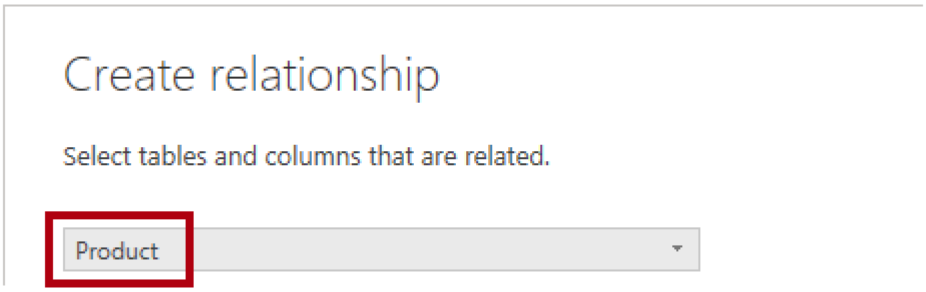


9. In the **Manage Relationships** window, notice that no relationships are yet defined.

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

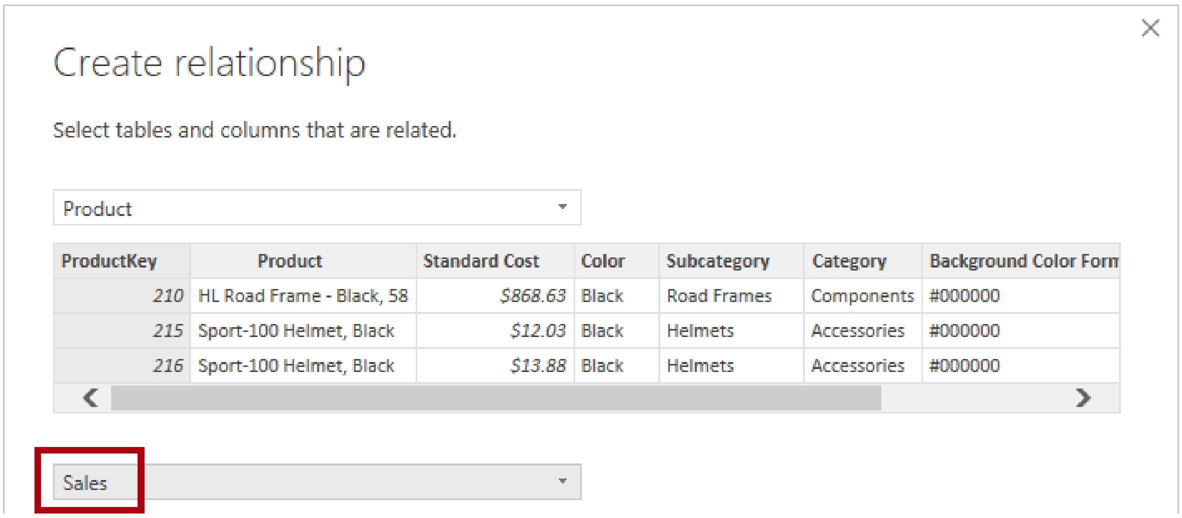


11. In the **Create Relationship** window, in the first dropdown list, select the **Product** table.


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.

Product

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

Sales

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.*

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.*

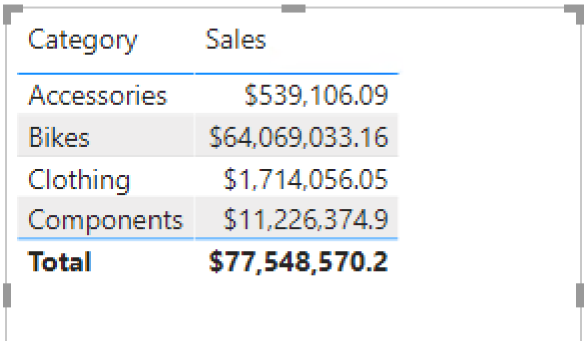
16. Notice that the **Mark This Relationship Active** is checked.

    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.

17. Click **OK**.

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

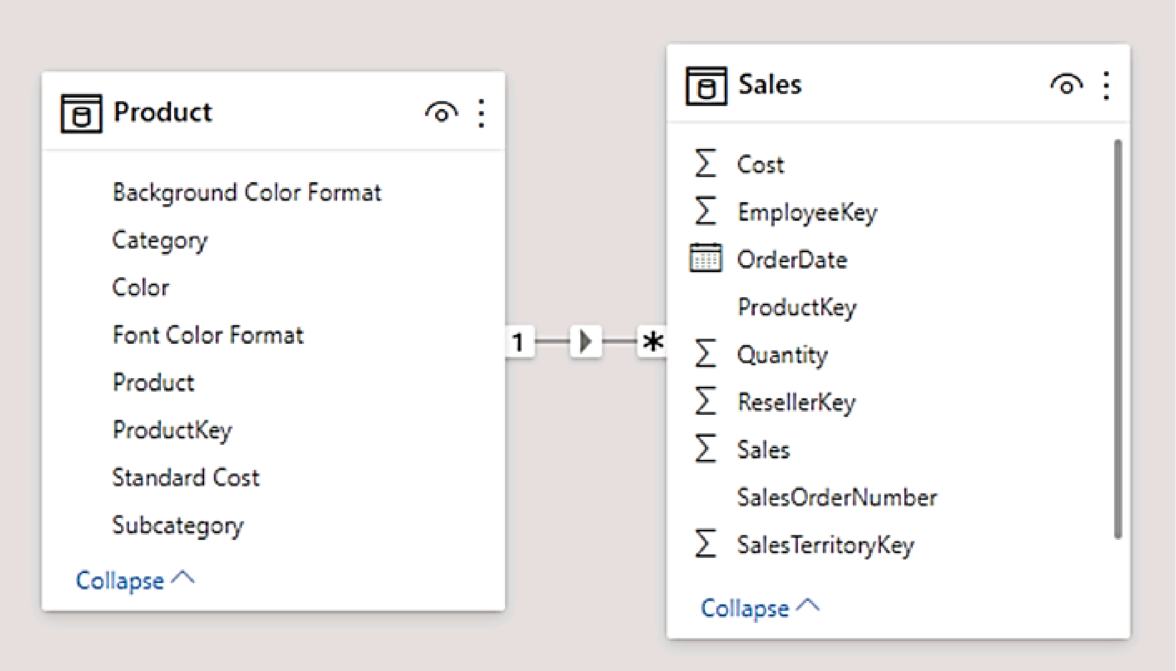
19. In the report, notice that the table visual updated to display different values for each product category.



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).



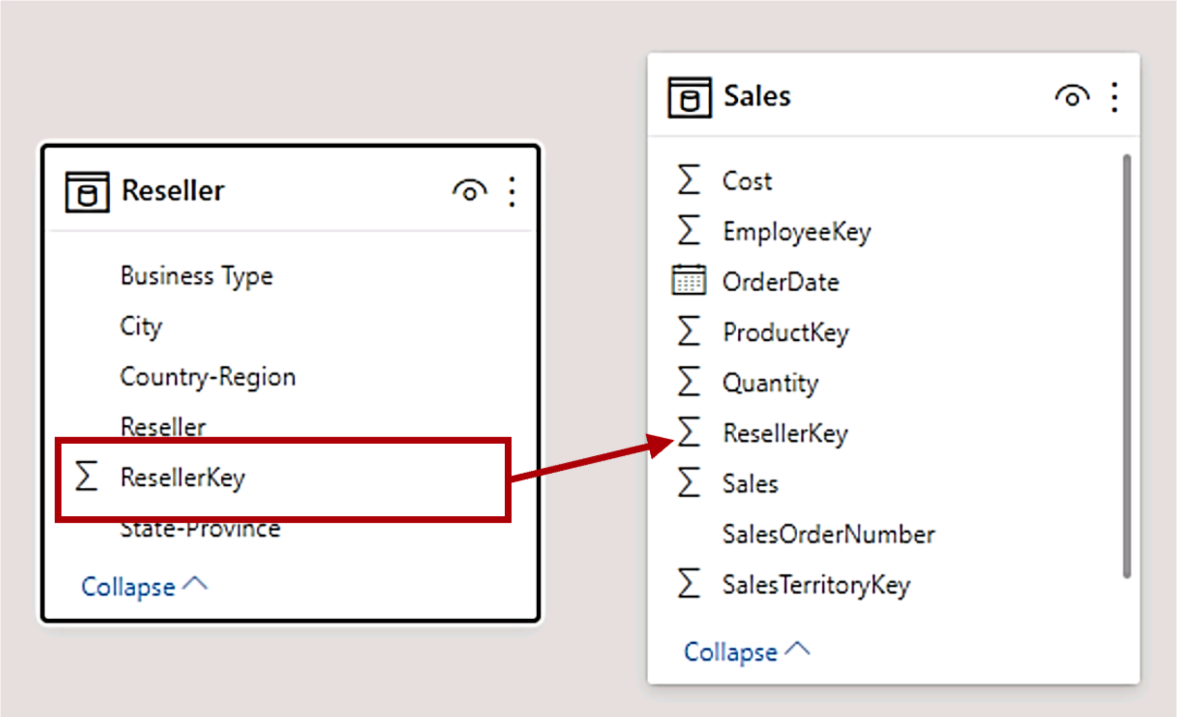
21. In the diagram, notice that you can interpret the cardinality which is represented by the **1** and ***** indicators.

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.

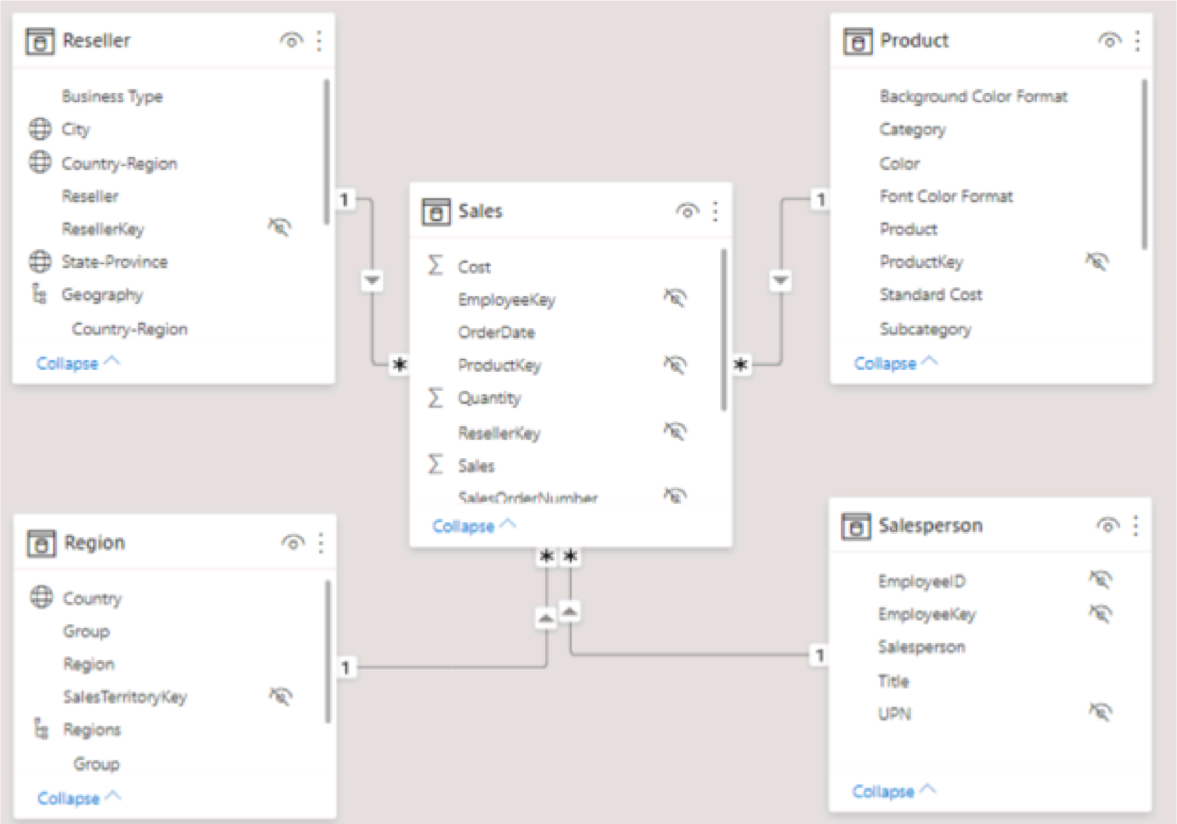


Tip: Sometime a column doesn't want to be dragged. If this situation arises, select a different column, and then select the column you intend to drag again, and then try again. Ensure that you see the new relationship added to the diagram.

24. Use the new technique to create the following two model relationships:

   - **Region | SalesTerritoryKey** to **Sales | SalesTerritoryKey**
   - **Salesperson | EmployeeKey** to **Sales | EmployeeKey**

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

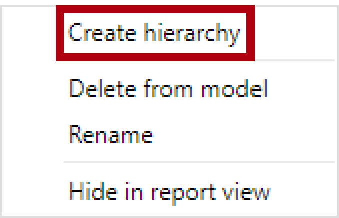
In this exercise you will configure each table by creating hierarchies, and hiding, formatting, and categorizing columns.

Task 1: Configure the Product table

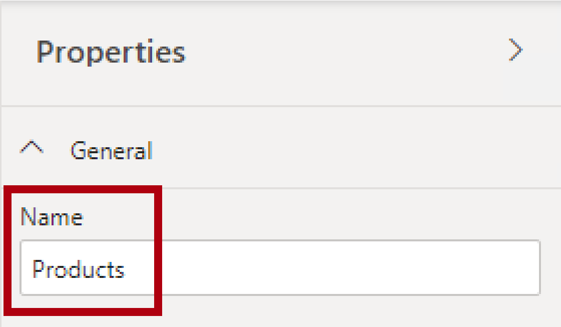
In this task you will configure the **Product** table.

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**.

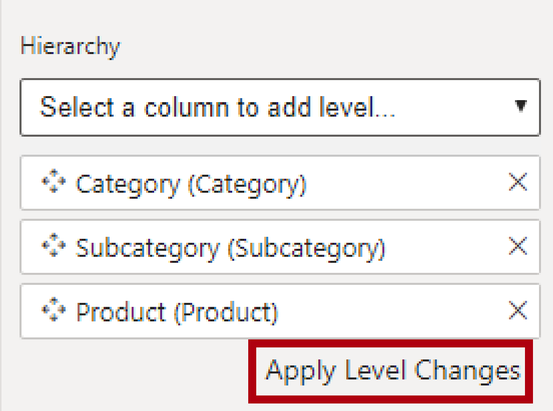


Name
Products

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

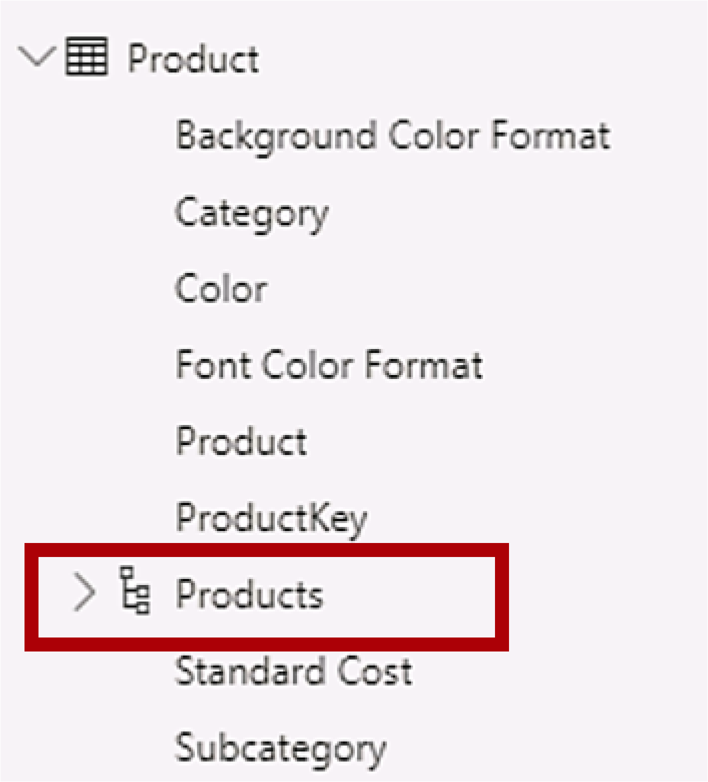
Category (Category)

Subcategory (Subcategory)

Product (Product)

*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

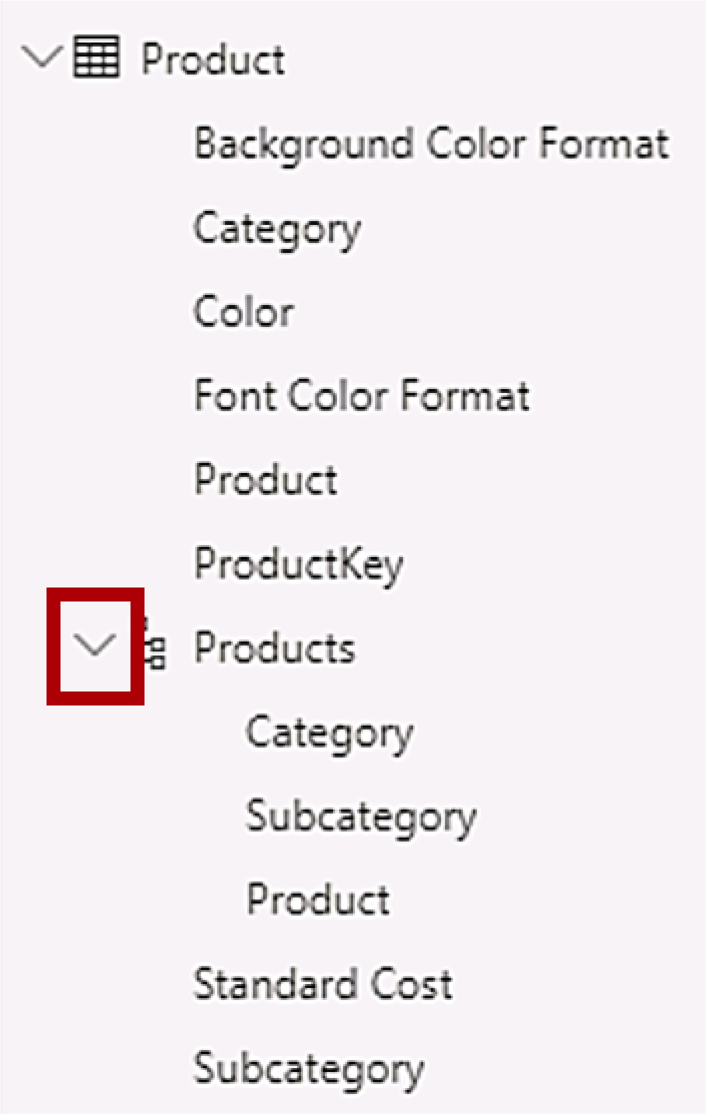
ProductKey

Products

Standard Cost

Subcategory

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



Product

Background Color Format

Category

Color

Font Color Format

Product

ProductKey

Products

Category

Subcategory

Product

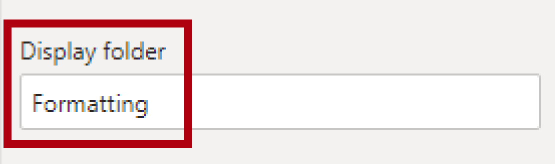
Standard Cost

Subcategory

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.

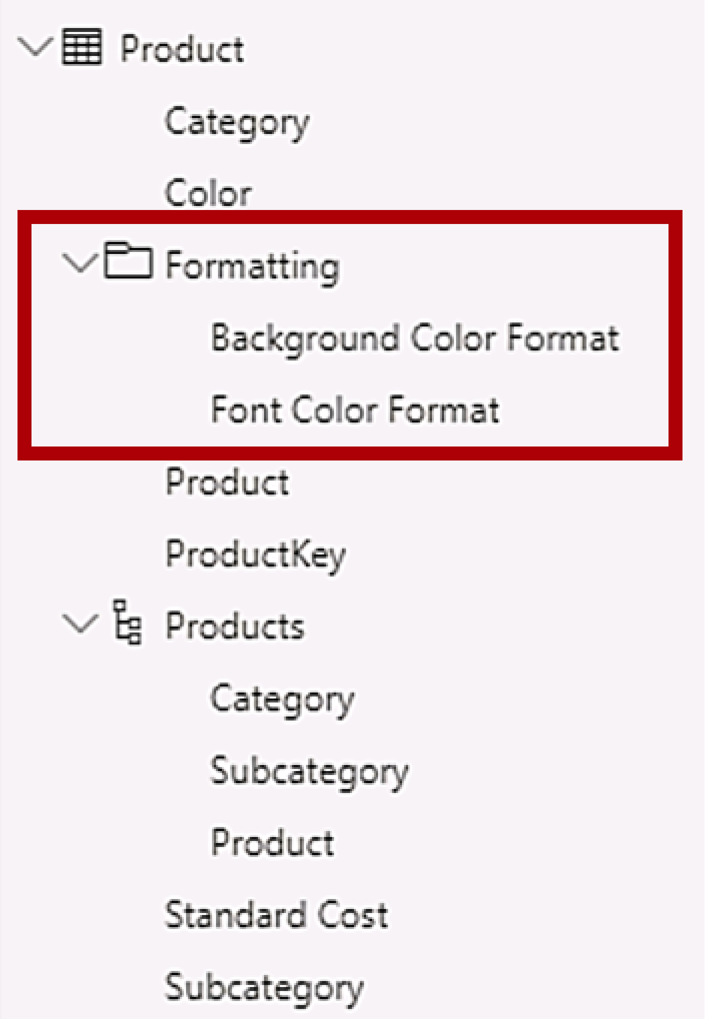
11. In the **Properties** pane, in the **Display Folder** box, enter **Formatting**.



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

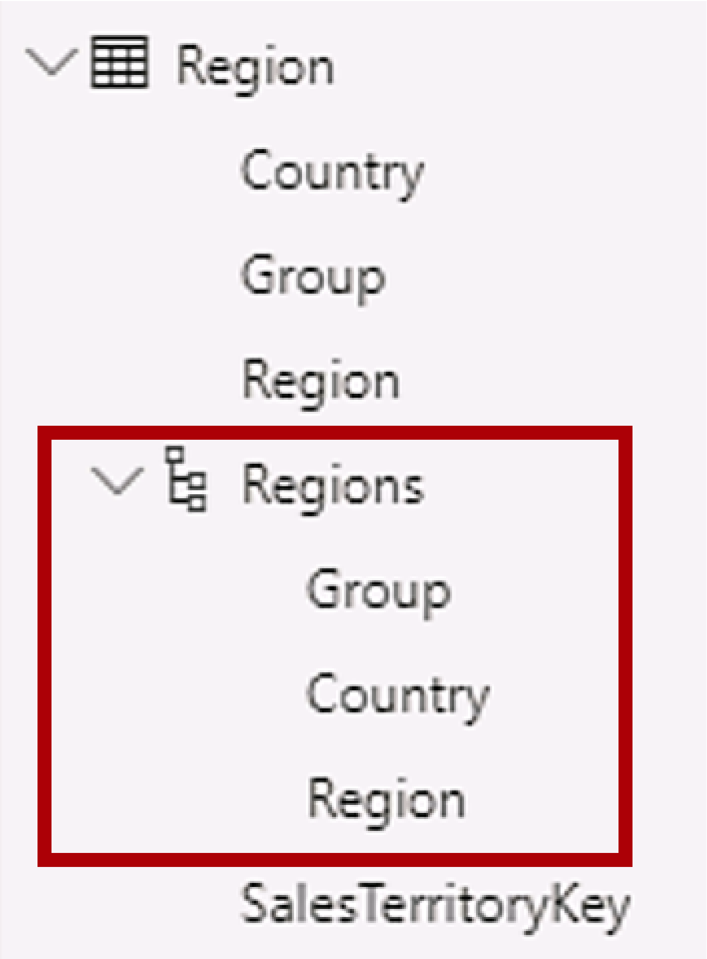
Display folders are a great way to declutter tables—especially for tables that comprise many fields.

Task 2: Configure the Region table

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

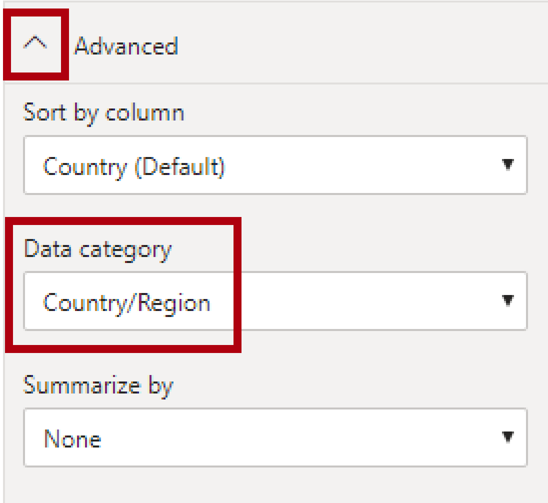
1. In the **Region** table, create a hierarchy named **Regions**, with the following three levels:

   - Group
   - Country
   - Region



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.

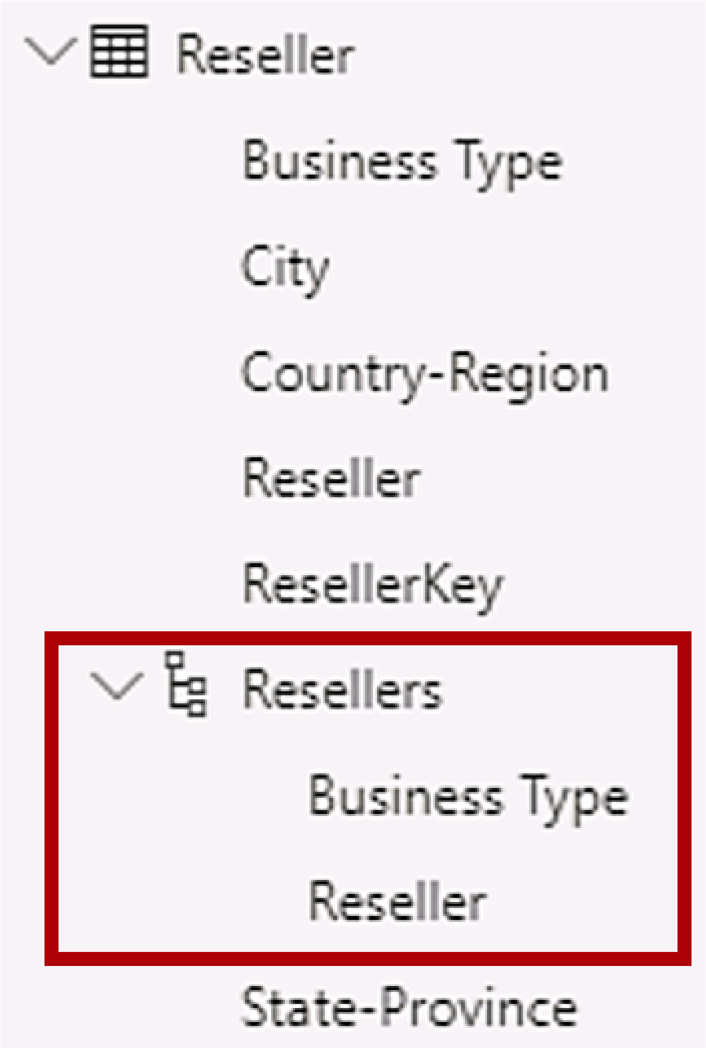
Task 3: Configure the Reseller table

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



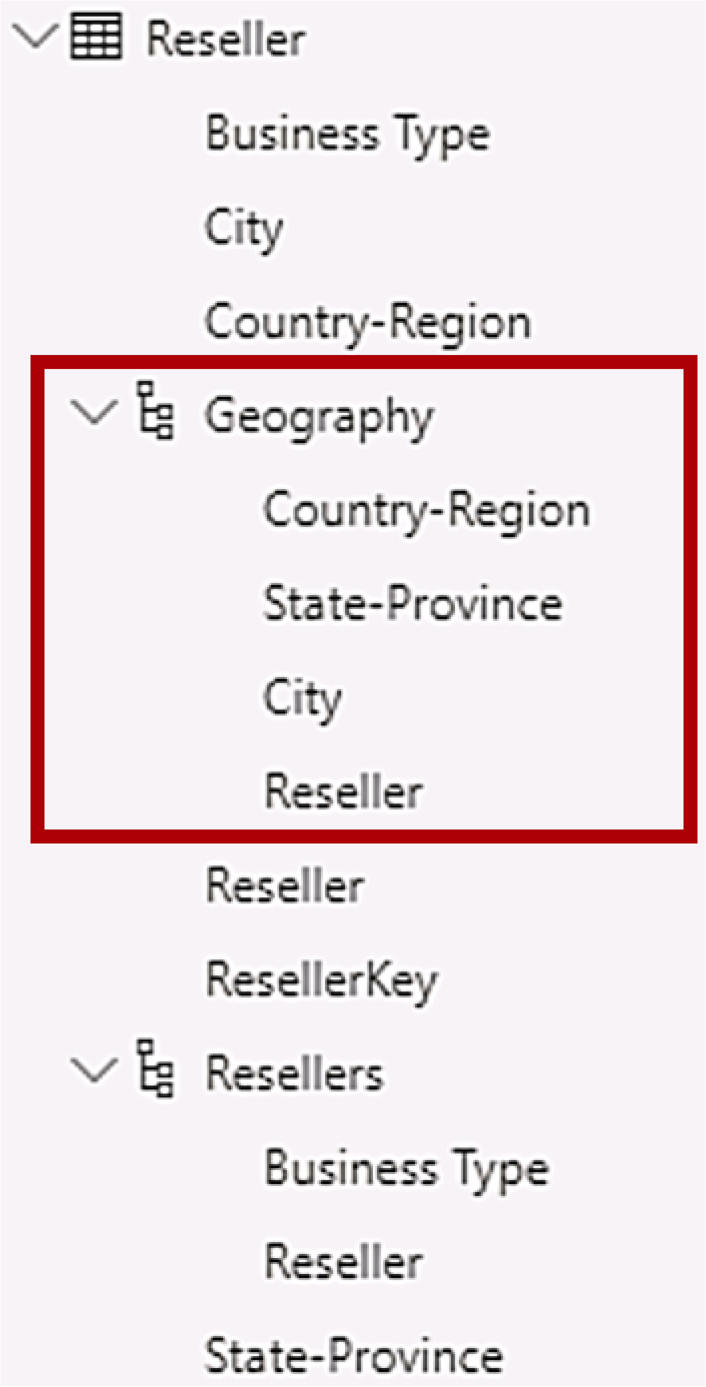
2. Create a second hierarchy named **Geography**, with the following four levels:

    - Country-Region

    - State-Province

    - City

- Reseller



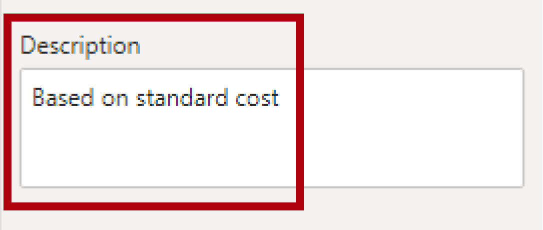
3. Set the **Data Category** for the **Country-Region**, **State-Province**, and **City** columns (not the hierarchy level) to **Country/Region**, **State or Province**, and **City**, respectively.

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**



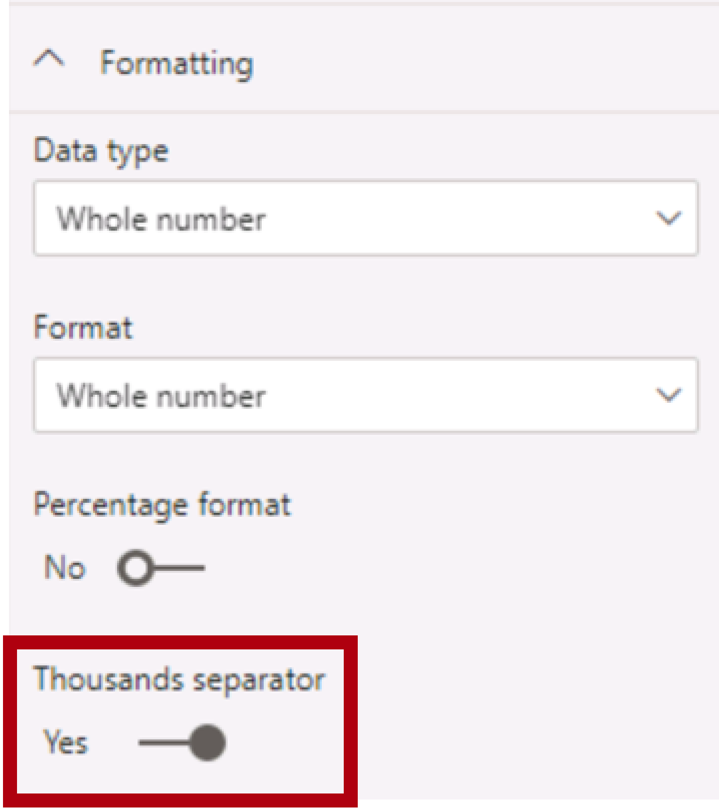
Description

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 **Yes**.



Formatting

Data type: Whole number

Format: Whole number

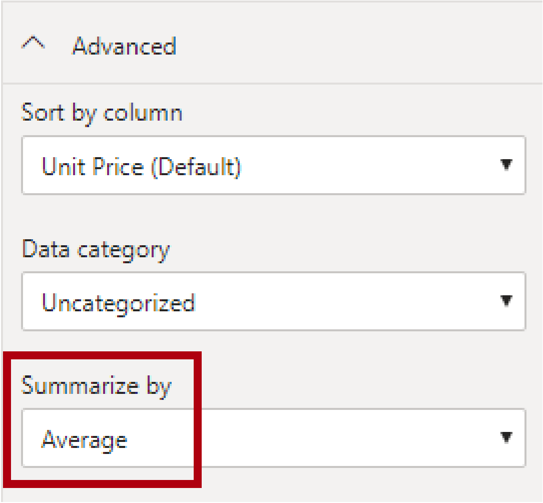
Percentage format: No

Thousands separator: Yes

5. Select the **Unit Price** column.

6. In the **Properties** pane, from inside the **Formatting** section, set 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**.



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.

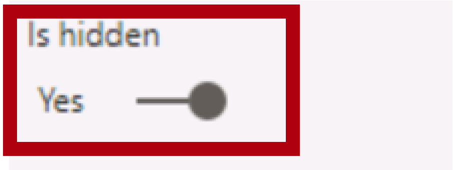
1. In the **Fields** pane, select the **Product | ProductKey** column.

2. While pressing the **Ctrl** key, select the following 13 columns (spanning multiple tables):

   - Region | SalesTerritoryKey
   - Reseller | ResellerKey
   - Sales | EmployeeKey
   - Sales | ProductKey
   - 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 **Yes**.

   Is hidden

   Yes

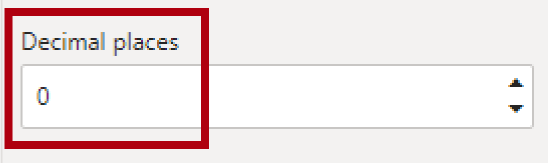
   The columns were hidden because they're either used by relationships or will be used in row-level security configuration or calculation logic.

   *You'll 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.

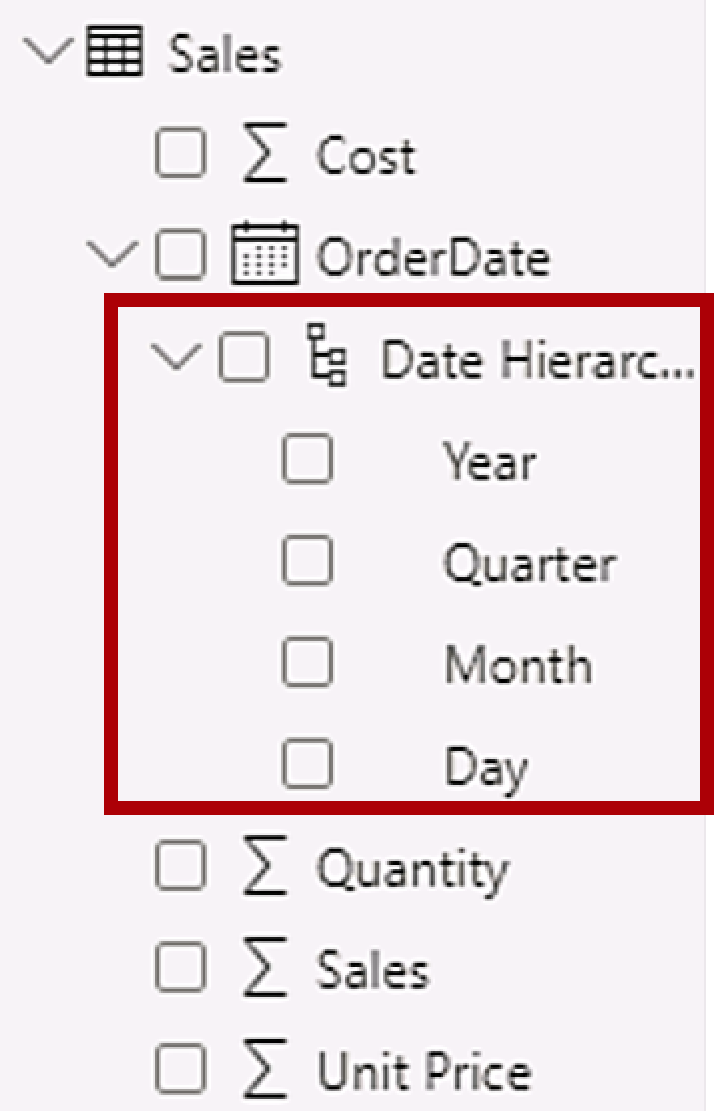
1. Switch to Report view.

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.

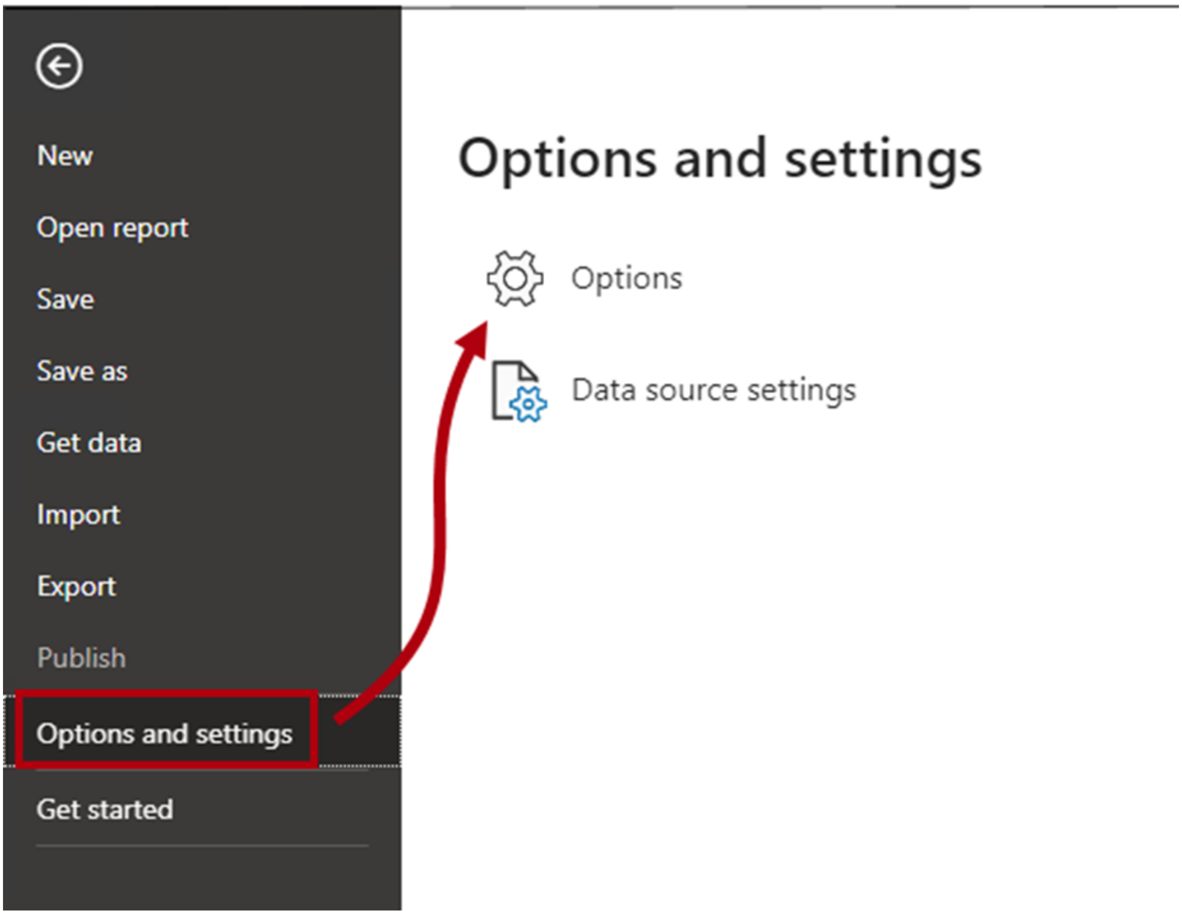


*The **Targets | TargetMonth** field delivers a similar hierarchy. These hierarchies were not created by you. They were created automatically. There is a problem, however. The Adventure Works financial year commences on July 1 of each year. But, in these automatically created date hierarchies, the date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

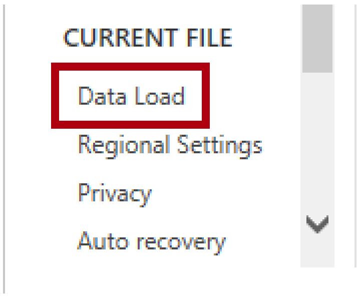
*You'll now turn this automatic behavior off. In the **Create DAX Calculations in Power BI Desktop, Part 1** lab, you'll use DAX to create a date table, and configure it define the Adventure Works' calendar.*

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

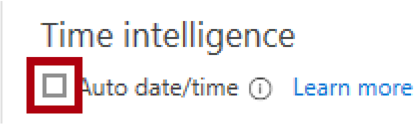
5. At the left, select **Options and Settings**, and then select **Options**.



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

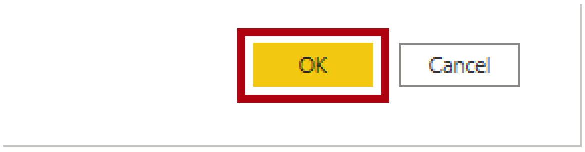


7. In the **Time Intelligence** section, uncheck **Auto Date/Time**.

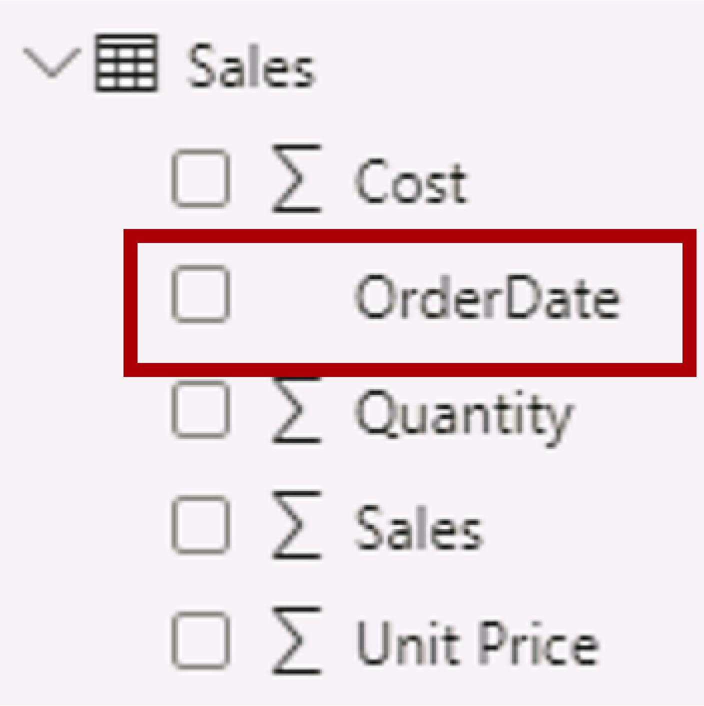


☐ Auto date/time

8. Click **OK**.



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



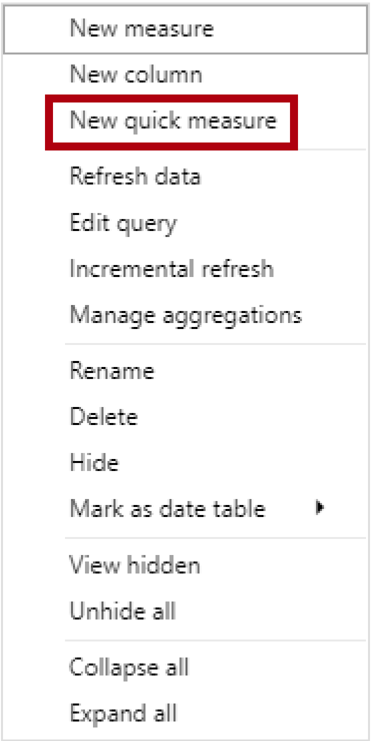
Exercise 4: Create Quick Measures

In this exercise you will create two quick measures.

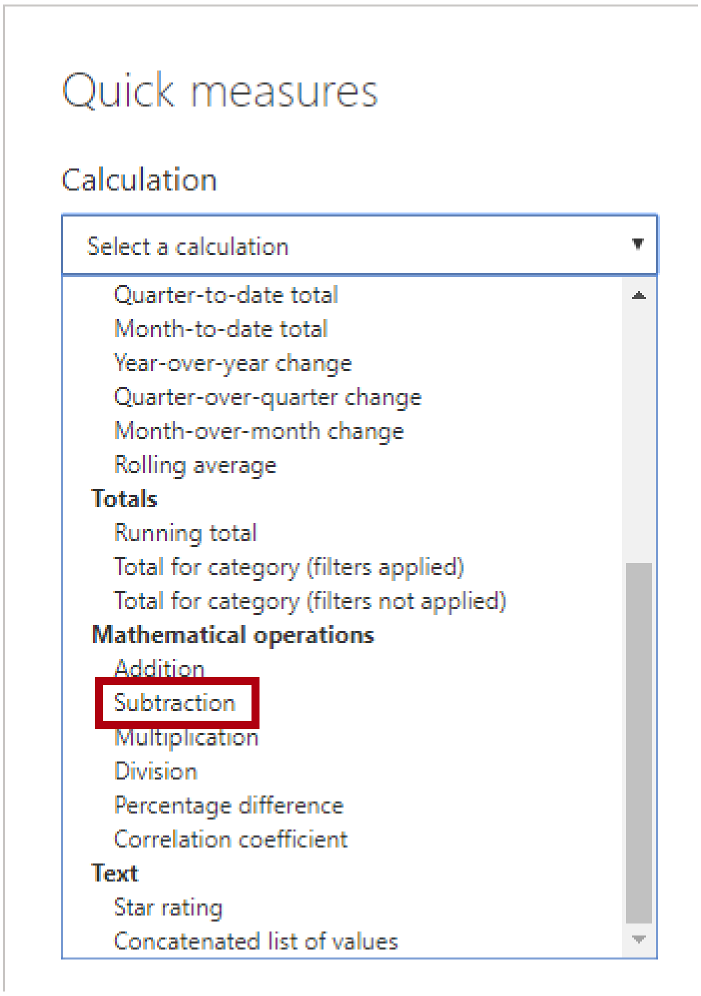
Task 1: Create quick measures

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**.



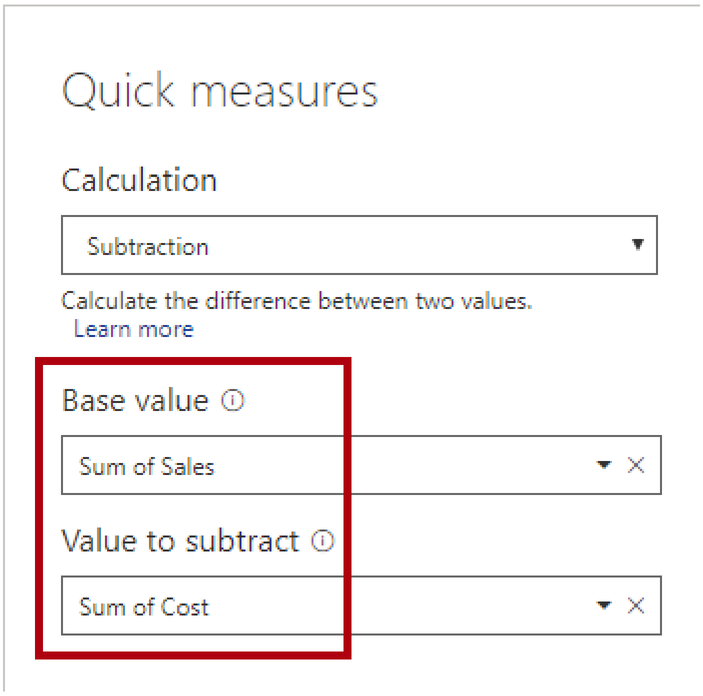
2. In the **Quick Measures** window, in the **Calculation** dropdown list, from inside the **Mathematical Operations** group, select **Subtraction**.



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.



Calculation

Subtraction

Calculate the difference between two values.
Learn more

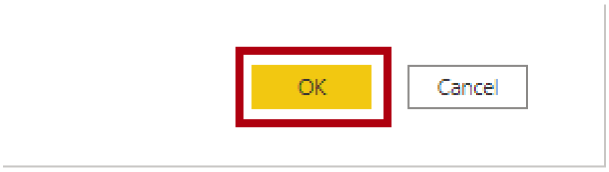
Base value

Sum of Sales

Value to subtract

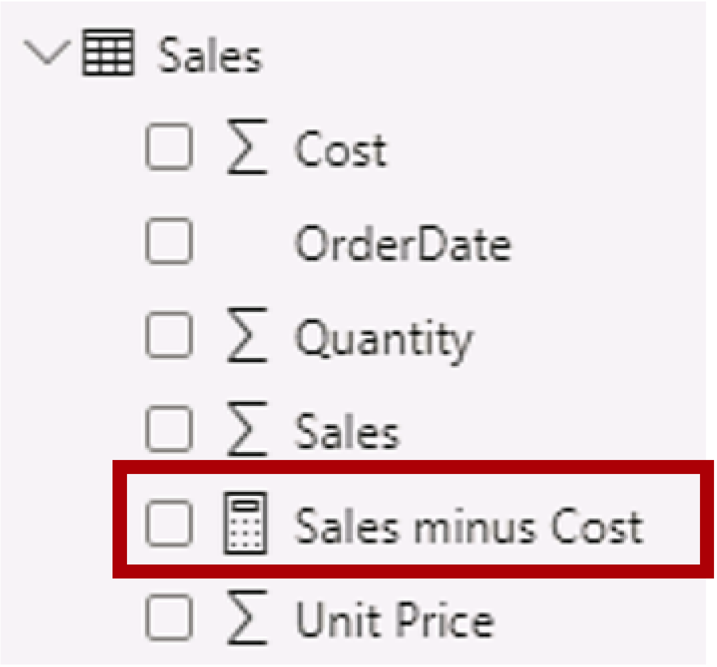
Sum of Cost

6. Click **OK**.



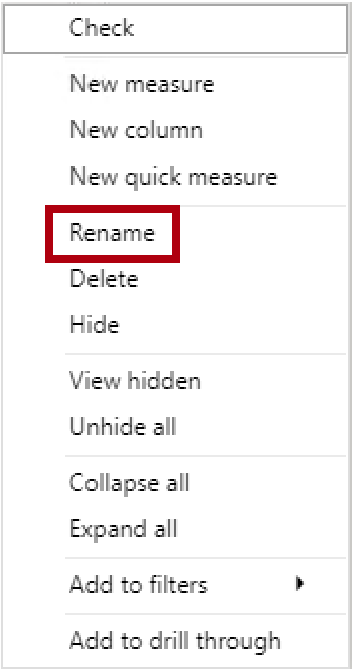
*A quick measure creates the calculation formula for you. They're easy and fast to create for simple and common calculations. You'll create measures without using this tool in the **Create DAX Calculations in Power BI Desktop, Part 1** lab.*

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**.

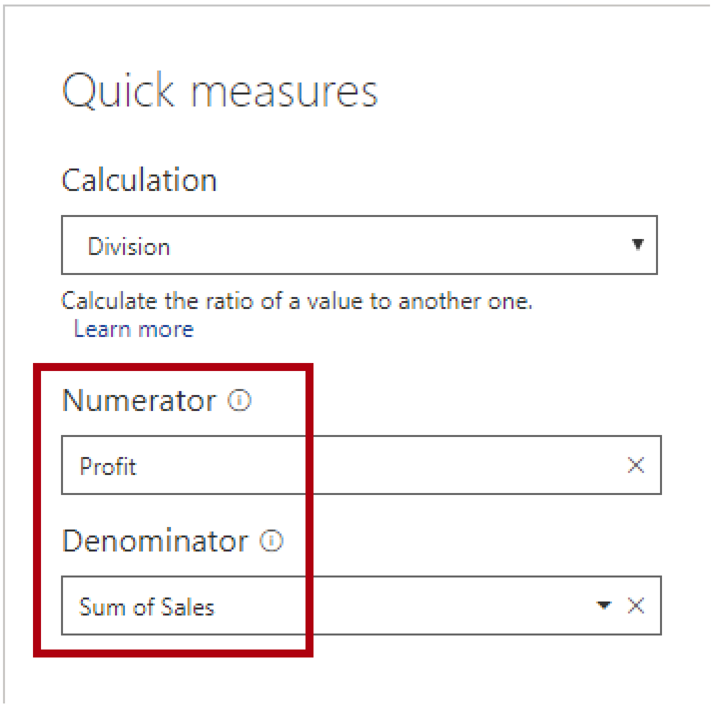


*Tip: To rename a field, you can also double-click it, or select it and press **F2**.*

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

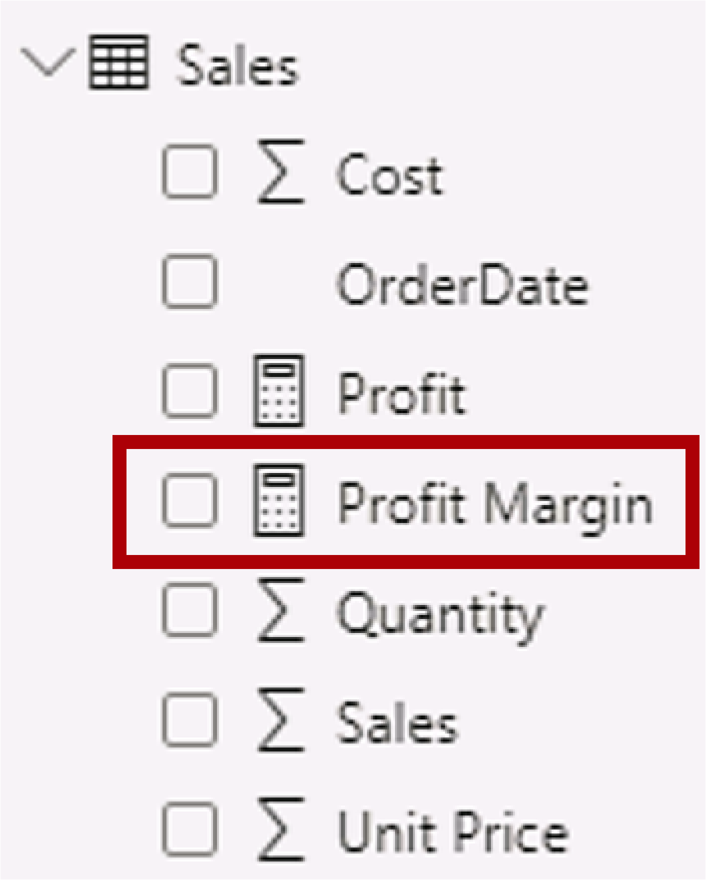
Calculate the ratio of a value to another one.
Learn more

Numerator

Profit

Denominator

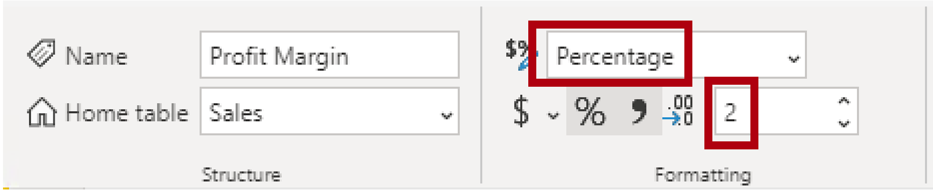
Sum of Sales



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

Structure

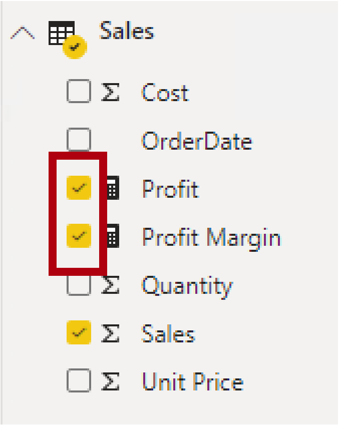
Percentage

$ % , 2

Formatting

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

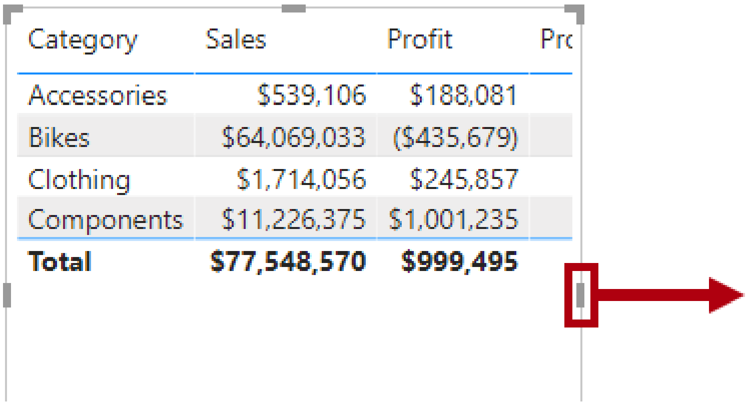
13. In the **Fields** pane, check the two measures.



Sales

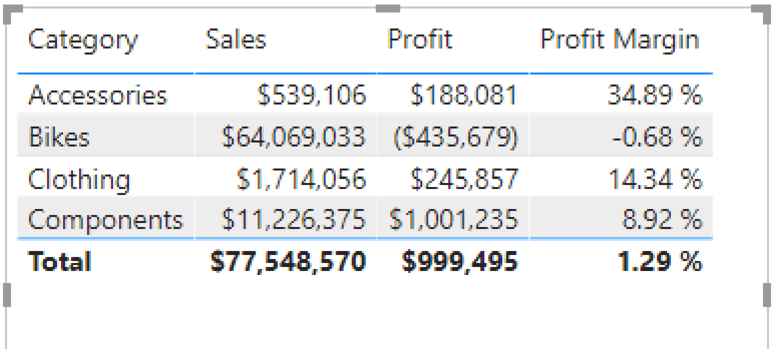
- □ Σ Cost
- □ OrderDate
- ☑ Profit
- ☑ Profit Margin
- □ Σ Quantity
- ☑ Σ Sales
- □ Σ Unit Price

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



Category	Sales	Profit	Pro
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 %**

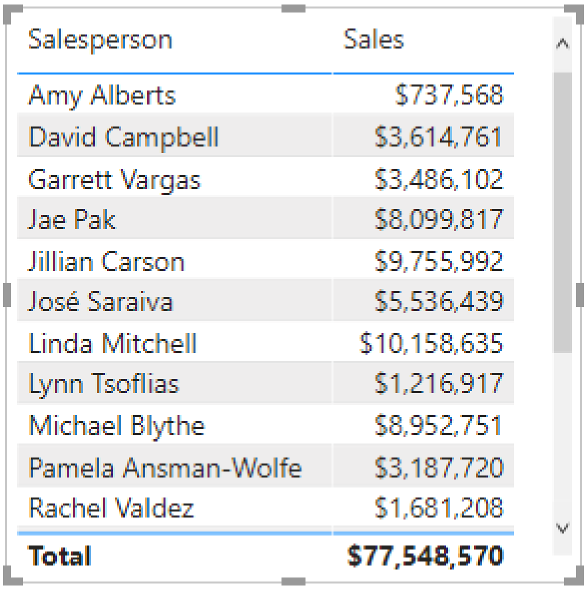
Task 2: Create a many-to-many relationship

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.*



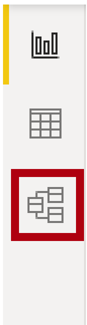
Salesperson	Sales
Amy Alberts	$737,568
David Campbell	$3,614,761
Garrett Vargas	$3,486,102
Jae Pak	$8,099,817
Jillian Carson	$9,755,992
José Saraiva	$5,536,439
Linda Mitchell	$10,158,635
Lynn Tsoflias	$1,216,917
Michael Blythe	$8,952,751
Pamela Ansman-Wolfe	$3,187,720
Rachel Valdez	$1,681,208
Total	**$77,548,570**

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**

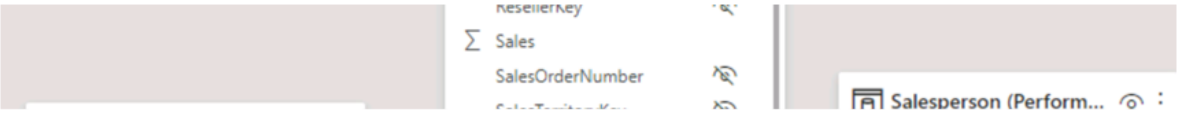
   - **Region | SalesTerritoryKey** to **SalespersonRegion | SalesTerritoryKey**

   *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.

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.

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.

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 reactivate the relationship when you introduce a calculated table that will allow analyzing sales made in the sales region(s) assigned to the salesperson (for performance analysis) 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.

> *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 be disabled by formatting the visual or removed by using calculation logic. You'll follow the second approach by creating 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.