

Design a Data Model in Power BI

Lab story

In this lab, you'll 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'll also create hierarchies and create quick measures.

In this lab you learn how to:

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

This lab should take approximately 45 minutes.

Get started

In this task, you'll set up the environment for the lab.

1. Open Power BI Desktop.



Tip: By default, the Getting Started dialog box opens in front of Power BI Desktop. You can choose to sign-in, and then close the pop-up.

2. To open the starter Power BI Desktop file, select the **File > Open Report > Browse Reports**.
3. Navigate to the **D:\PL300\Labs\03-configure-data-model-in-power-bi-desktop\Starter** folder and select the **Sales Analysis** file.
4. Close any informational windows that may open.
5. Go to **File > Save As** and save the file to the **D:\PL300\MySolution** folder.

Create model relationships

In this task, you'll create model relationships. The file was configured to not identify relationships between tables in the previous labs. This isn't the default setting, but is recommended to prevent extra work creating the correct relationships for your model.

*Important: 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.*

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



2. If you don't 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.

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

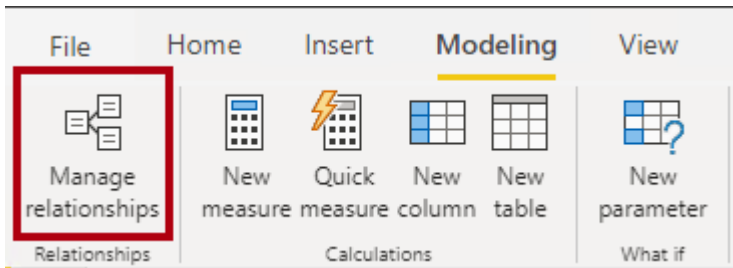


- To view all table fields, in the **Data** pane, right-click an empty area, and then select **Expand All**.
- To create a table visual, in the **Data** pane, from inside the **Product** table, check the **Category** field.
- To add another column to the table, in the **Data** pane, check the **Sales | Sales** field.
- 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.

*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 isn't filtered. You'll now add a relationship to propagate filters between the tables.*

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

- On the **Modeling** ribbon tab, from inside the **Relationships** group, select **Manage Relationships**.



- In the **Manage Relationships** window, notice that no relationships are yet defined.
- To create a relationship, select **New**.
- In the **Create Relationship** window, in the first dropdown list, select the **Product** table.

Create relationship

Select tables and columns that are related.

Product

- 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

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

The columns were selected because they share the same name and data type. You may need to find matching columns with different names in real data.

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.*

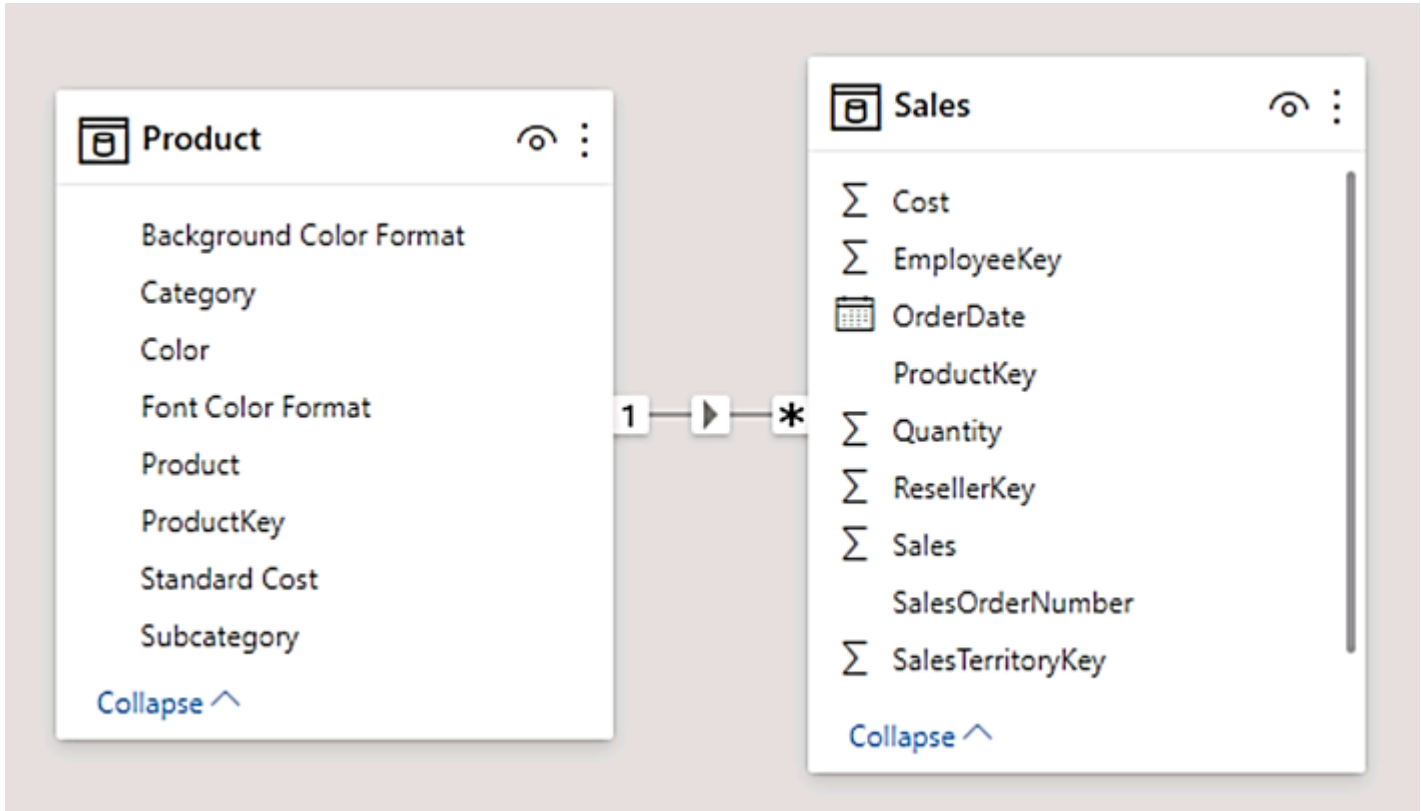
16. Notice that the **Mark This Relationship Active** is checked.

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. Select **OK**, notice in the **Manage Relationships** window that the new relationship is listed, and then select **Close**.

18. Notice there’s now a connector between the two tables (it doesn’t matter if the tables are positioned next to each other).

- a. You can interpret the cardinality that is represented by the **1** and **(*)** indicators.
- b. Filter direction is represented by the arrow head.
- c. A solid line represents an active relationship; a dashed line represents an inactive relationship.
- d. 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.

19. To create a new relationship using a different technique, from the **Reseller** table, drag the **ResellerKey** column onto the **ResellerKey** column of the **Sales** table.

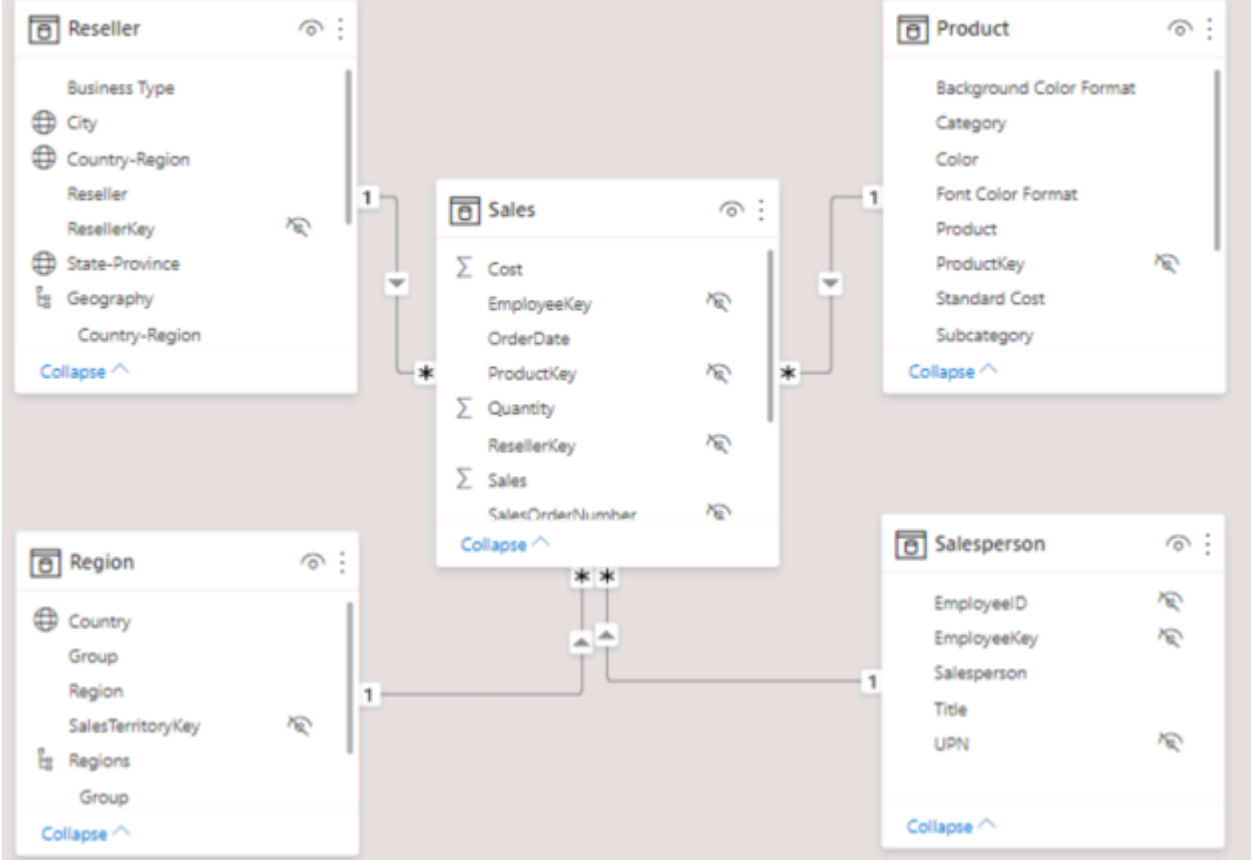
Tip: Sometime a column doesn’t want to be dragged. If this situation arises, select a different column, and then select the column you intend to drag again, and then try again. Ensure that you see the new relationship added to the diagram.

![Drag ResellerKey from Reseller table onto Sales ResellerKey](/PL-300-Microsoft-Power-BI-Data-Analyst/Instructions/Labs/Linked_image_Files/03-configure-data-model-in-power-bi-desktop_image22.png)

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

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

21. 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.



[Lab story](#)

[Get started](#)

[Create model relationships](#)

[Configure Tables](#)

[Configure the Product table](#)

[Configure the Region table](#)

[Configure the Reseller table](#)

[Configure the Sales table](#)

[Bulk update properties](#)

[Review the Model Interface](#)

[Review the model interface](#)

[Create Quick Measures](#)

[Create quick measures](#)

[Create a many-to-many relationship](#)

[Relate the Targets table](#)

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

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

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

23. Save the Power BI Desktop file.

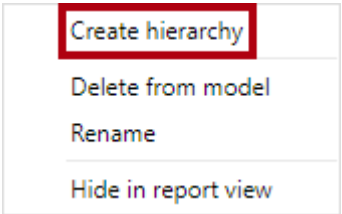
Configure Tables

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

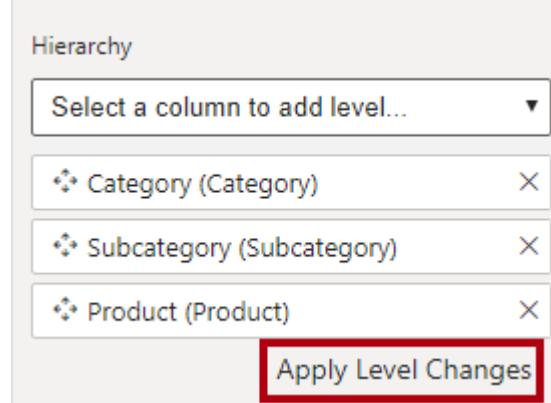
Configure the Product table

In this task, you'll configure the **Product** table.

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

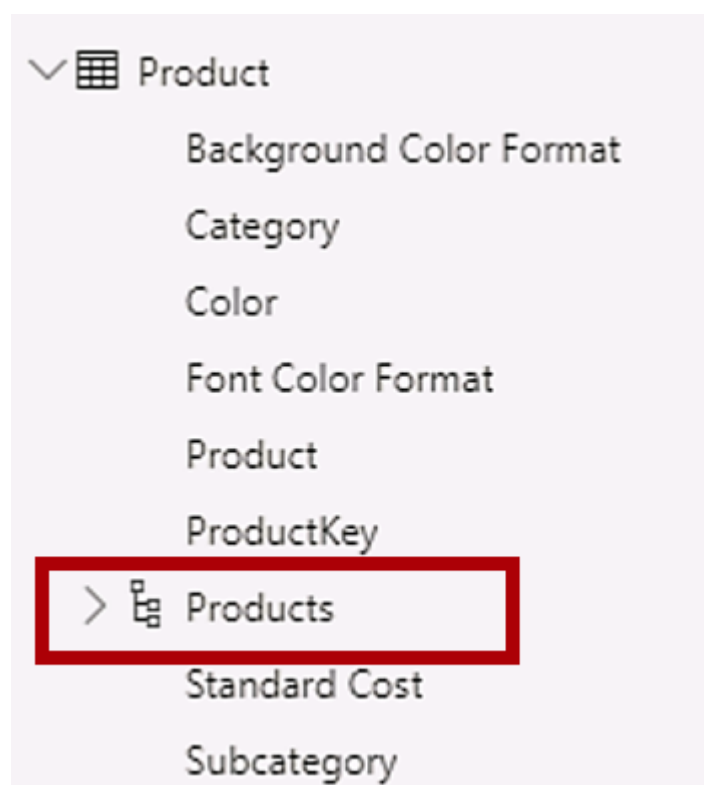


3. In the **Properties** pane (to the left of the **Data** pane), in the **Name** box, replace the text with **Products**.
4. To add the second level to the hierarchy, in the **Properties** pane, in the **Hierarchy** dropdown list, select **Subcategory** (you might need to scroll down inside the pane).
5. To add the third level to the hierarchy, in the **Hierarchy** dropdown list, select **Product**.
6. To complete the hierarchy design, select **Apply Level Changes**.

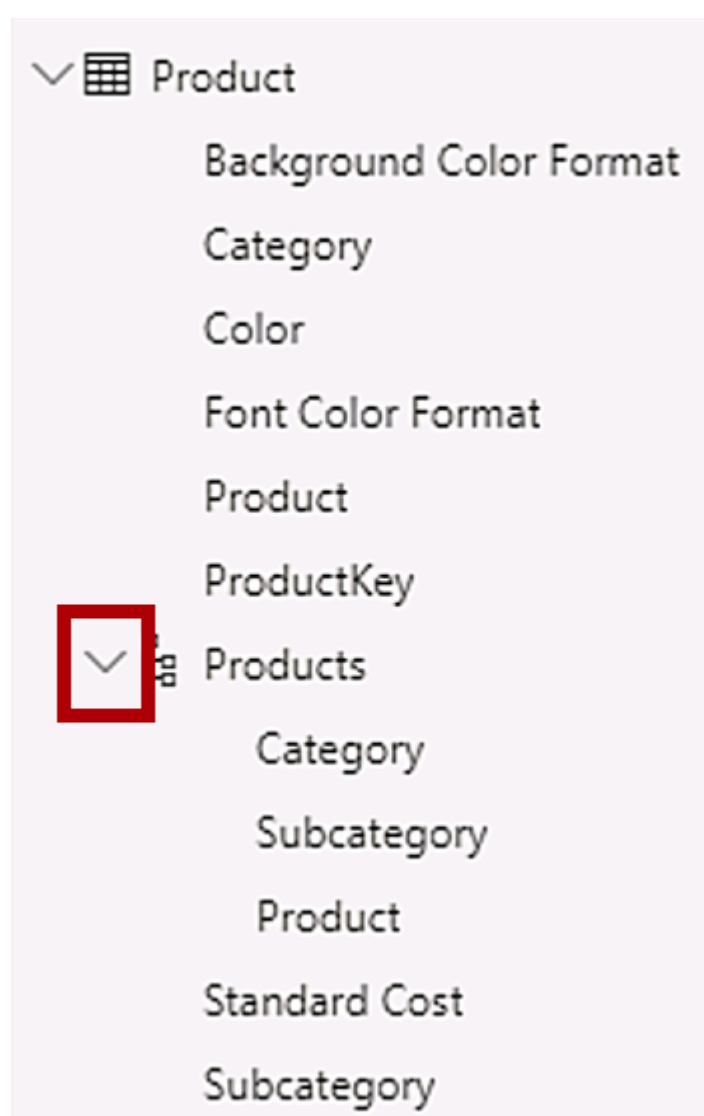


*Tip: Don't forget to select **Apply Level Changes**—it's a common mistake to overlook this step.*

1. In the **Data** pane, notice the **Products** hierarchy.



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

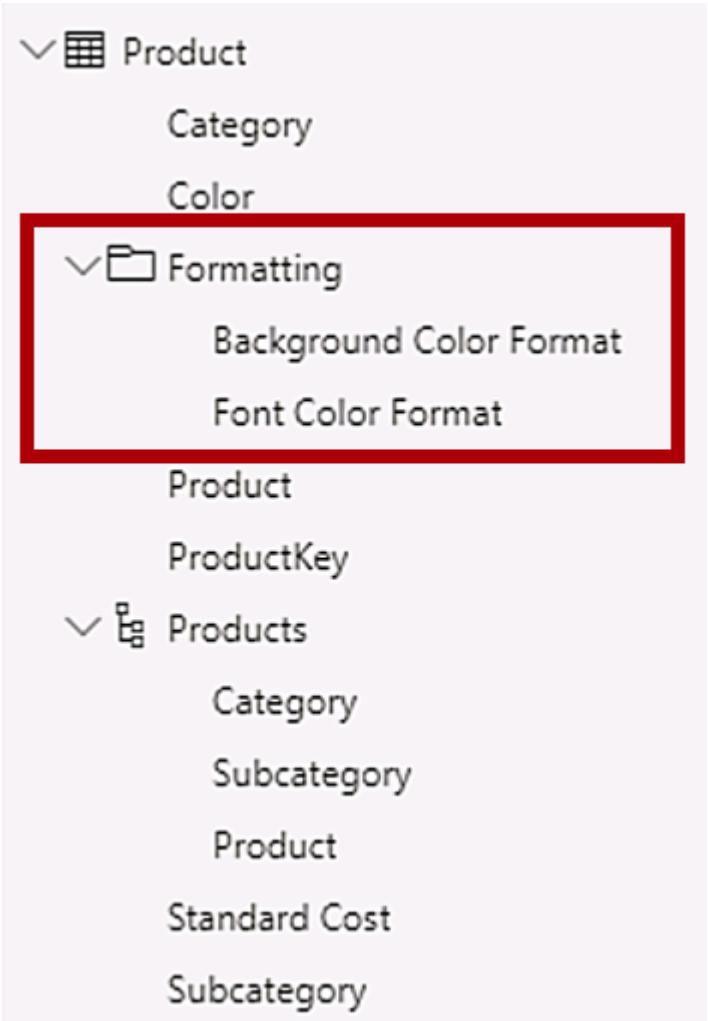


3. To organize columns into a display folder, in the **Data** pane, first select the **Background Color Format** column.
4. While pressing the **Ctrl** key, select the **Font Color Format** column.
5. In the **Properties** pane, in the **Display Folder** box, enter **Formatting**.



6. In the **Data** pane, notice that the two columns are now inside a folder.

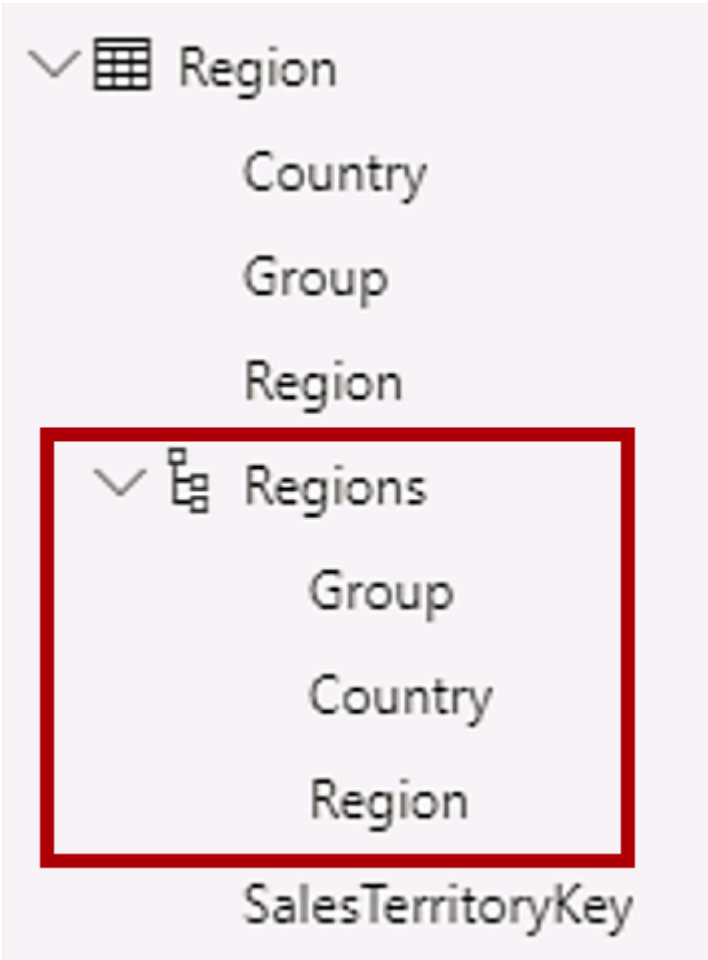
Display folders are a great way to declutter tables—especially for tables that comprise many fields. They're logical presentation only.



Configure the Region table

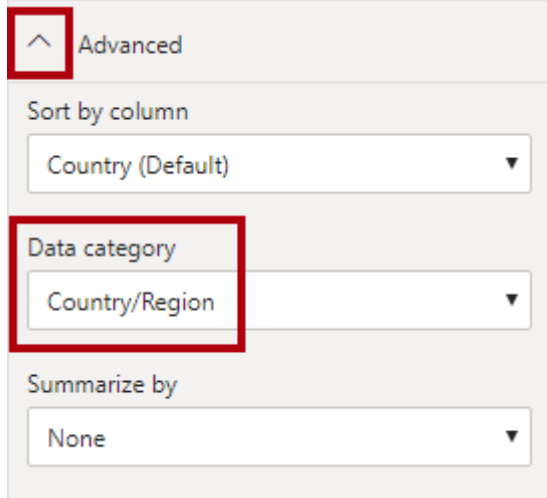
In this task, you'll configure the **Region** table.

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**.

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.



Configure the Reseller table

In this task, you'll 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.

Configure the Sales table

In this task, you'll 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.*

*Descriptions can be applied to tables, columns, hierarchies, or measures. In the **Data** 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**.
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**.

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

Bulk update properties

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

1. In the **Data** 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**.

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** lab.

1. Multi-select the following three columns:

- Product | Standard Cost
- Sales | Cost
- Sales | Sales

2. In the **Properties** pane, from inside the **Formatting** section, set the **Decimal Places** property to **0** (zero).

Review the Model Interface

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

Review the model interface

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

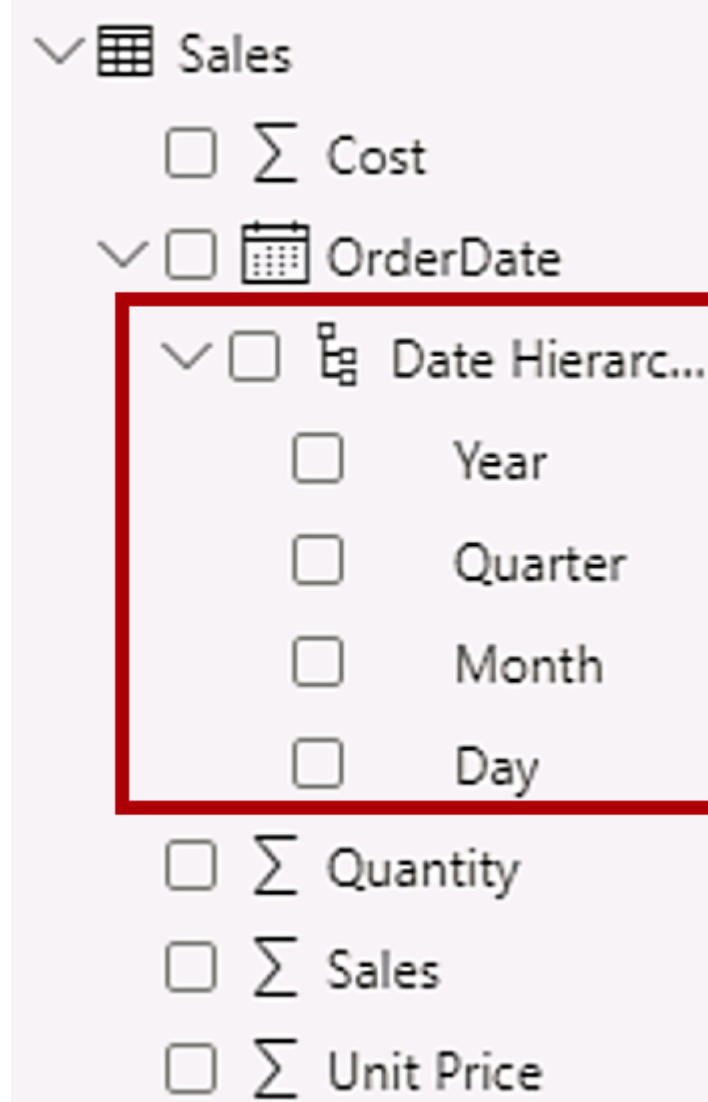
1. Switch to Report view.

2. In the **Data** 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 isn't 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

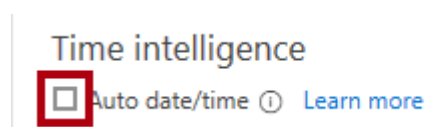
3. Expand the **Sales | OrderDate** field, and then notice that it reveals a date hierarchy.

*The **Targets | TargetMonth** field delivers a similar hierarchy. These hierarchies weren't created by you. They were created automatically. There's 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.*

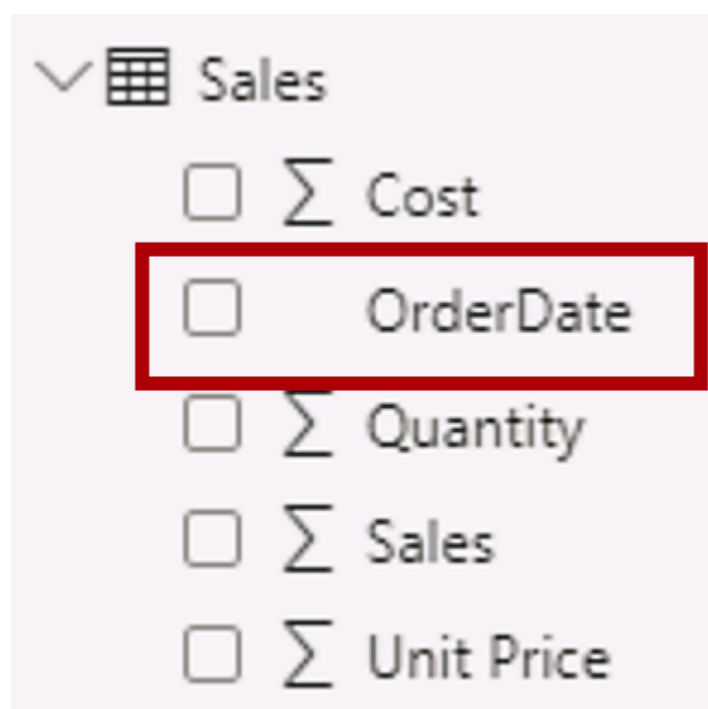


You'll now turn off this automatic behavior. In the **Create DAX Calculations in Power BI Desktop** lab, you'll use DAX to create a date table, and configure it to define the Adventure Works' calendar.

1. To turn off auto/date time, Navigate to **File > Options and Settings > Options > Current File** group, and select **Data Load**.
 - a. In the **Time Intelligence** section, uncheck **Auto Date/Time**.



2. In the **Data** pane, notice that the date hierarchies are no longer available.



Create Quick Measures

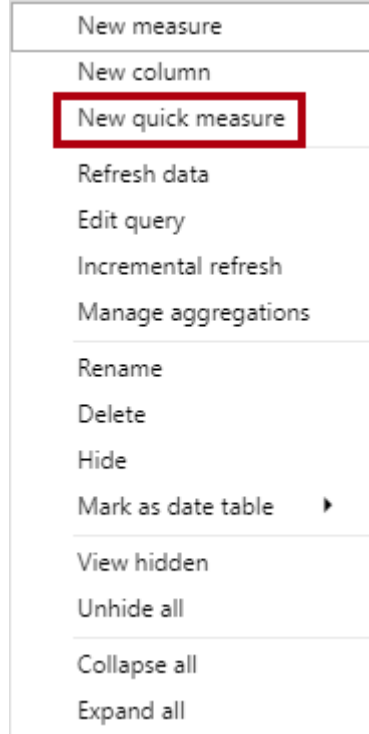
In this exercise, you'll create two quick measures.

Create quick measures

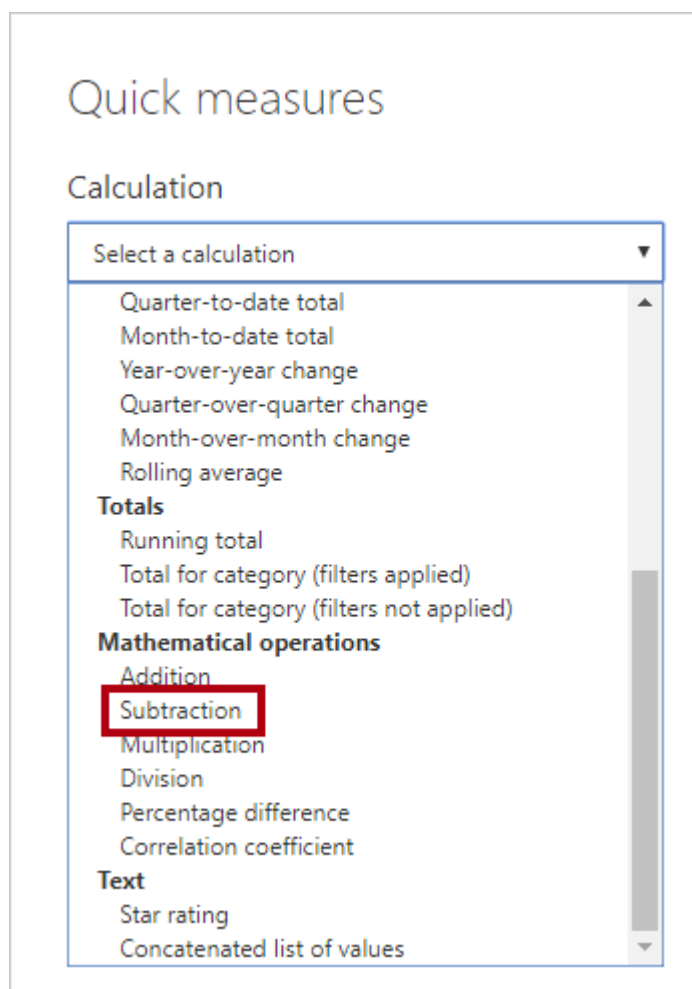
In this task, you'll create two quick measures to calculate profit and profit margin.

*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** lab.*

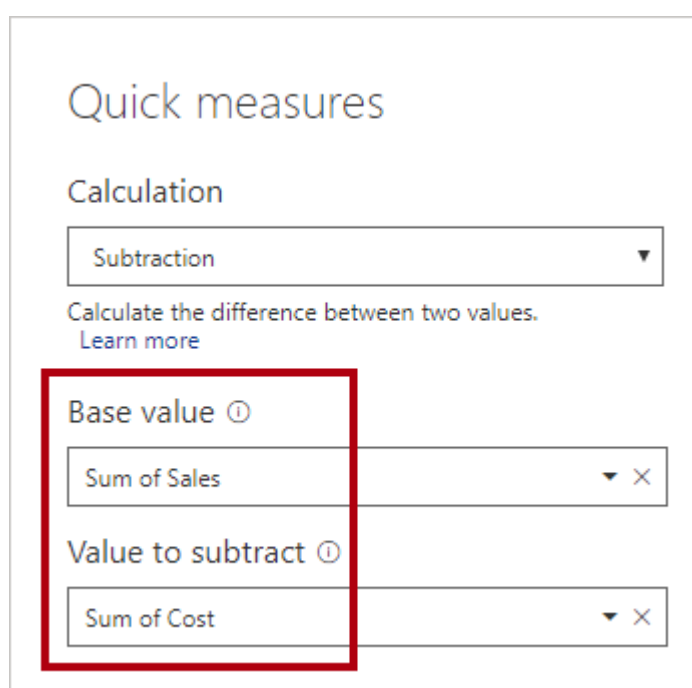
1. In the **Data** pane, right-click the **Sales** table, and then select **New Quick Measure**.



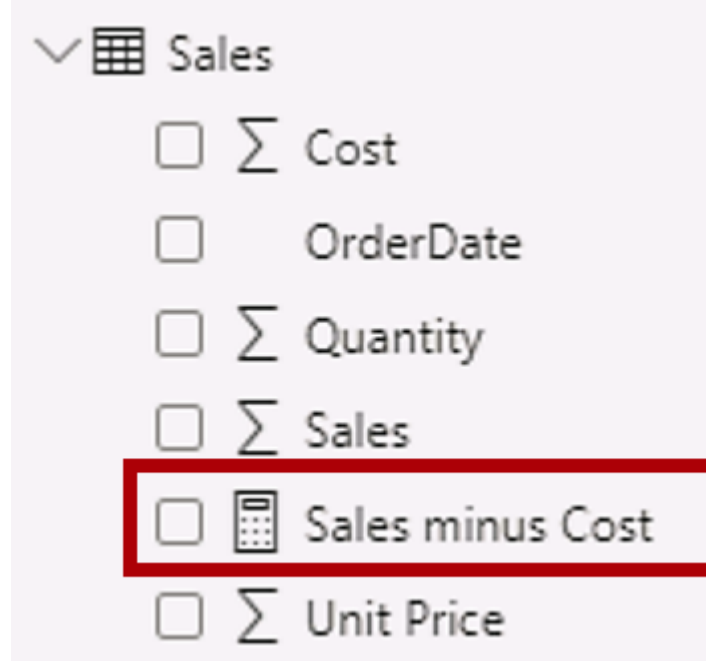
2. In the **Quick Measures** window, in the **Calculation** dropdown list, from inside the **Mathematical Operations** group, select **Subtraction**.



3. In the **Data** 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.



6. In the **Data** pane, inside the **Sales** table, notice that new measure.
 - a. Measures use the calculator icon.



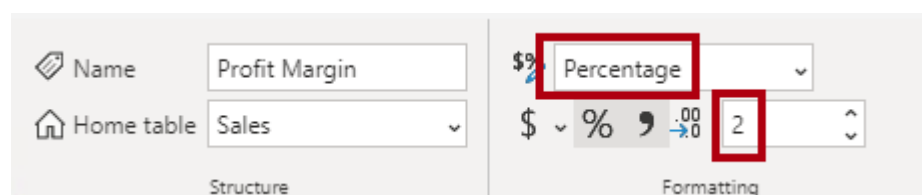
7. To rename the measure, right-click it, select **Rename**, then rename to **Profit**.

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

8. In the **Sales** table, add a second quick measure, based on the following requirements:

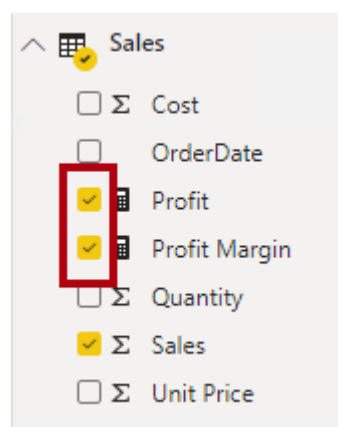
- Use the **Division** mathematical operation
- Set the **Numerator** to the **Sales | Profit** field
- Set the **Denominator** to **Sales | Sales** field
- Rename the measure as **Profit Margin**

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



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

11. In the **Data** pane, check the two measures.



12. Select 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	

13. Verify that the measures produce reasonable results that are correctly formatted.

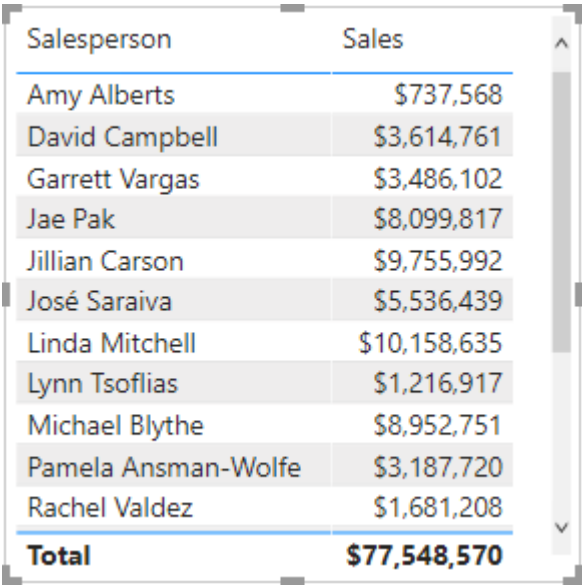
Category	Sales	Profit	Profit Margin
Accessories	\$539,106	\$188,081	34.89 %
Bikes	\$64,069,033	(\$435,679)	-0.68 %
Clothing	\$1,714,056	\$245,857	14.34 %
Components	\$11,226,375	\$1,001,235	8.92 %
Total	\$77,548,570	\$999,495	1.29 %

Create a many-to-many relationship

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

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.

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

A screenshot of a Power BI table visual. The table has two columns: 'Salesperson' and 'Sales'. It lists 13 salespeople and a total row. The sales values are in US dollars. The total sales amount is \$77,548,570.

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.

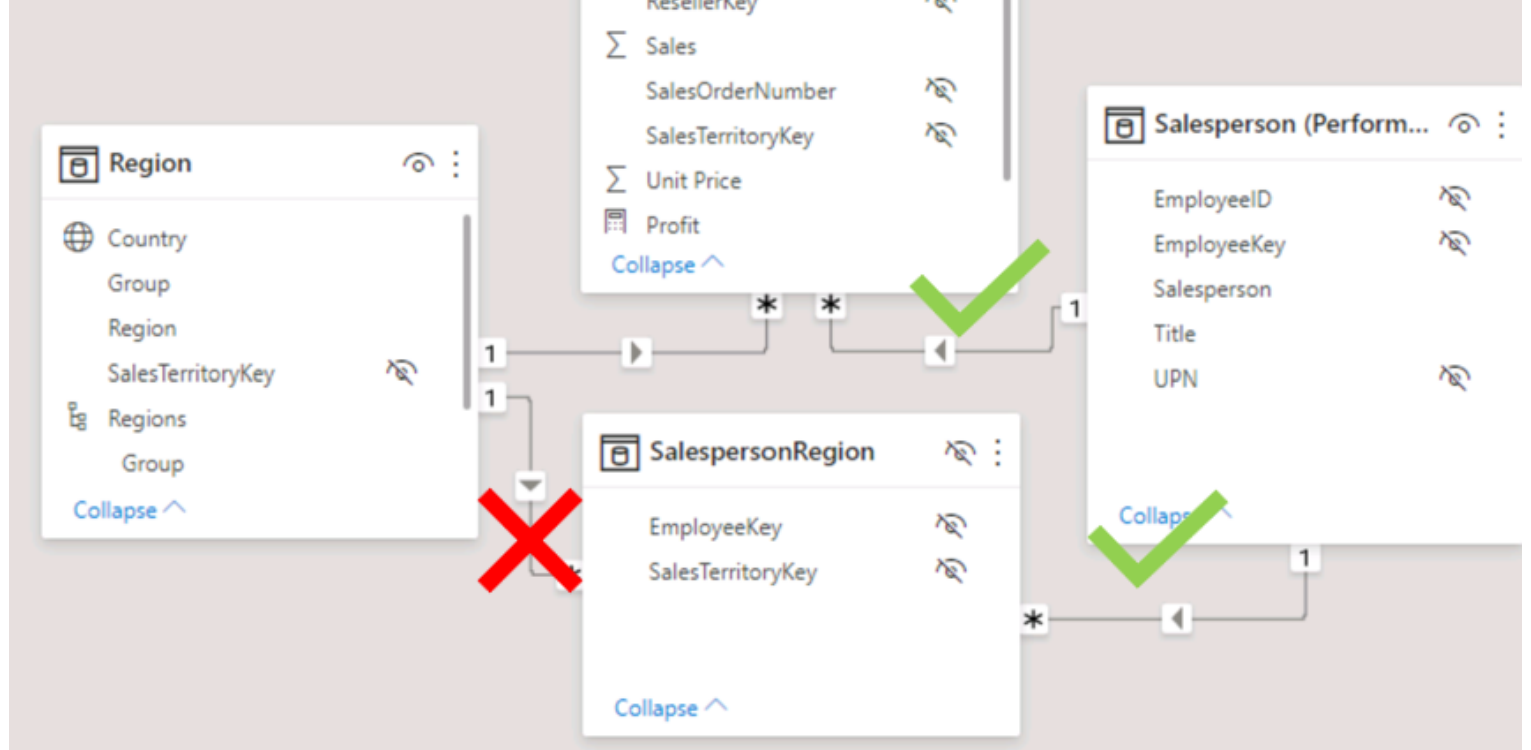
From a performance management perspective, a salesperson's sales (based on their assigned territories) need to be analyzed and compared with sales targets. You'll create relationships to support this analysis in the next exercise.

2. Notice that **Michael Blythe** has sold almost \$9 million.
3. Switch to Model view, then drag the **SalespersonRegion** table to position it between the **Region** and **Salesperson** tables.
4. 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.

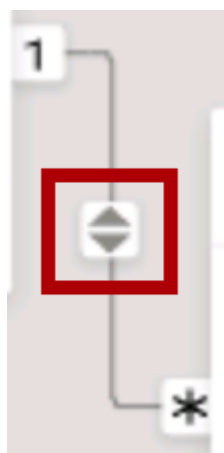
5. Switch to Report view, and then notice that the visual hasn't updated—the sales result for Michael Blythe hasn't changed.
6. Switch back to Model view, and then follow the relationship filter directions (arrowhead) from the **Salesperson** table.

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



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

10. Notice that the relationship has a double arrowhead now.



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

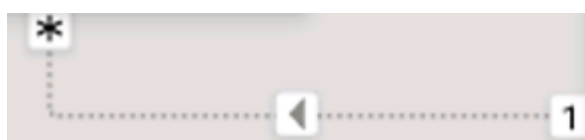
*The issue now relates to the fact that there are two possible filter propagation paths between the **Salesperson** and **Sales** tables. This ambiguity is internally resolved, based on a “least number of tables” assessment. To be clear, you shouldn’t design models with this type of ambiguity—the issue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

12. Switch to Model view to force filter propagation via the bridging table. Edit (double-click) the relationship between the **Salesperson** and **Sales** tables.

13. In the **Edit Relationship** window, uncheck the **Make This Relationship Active** checkbox, and select **OK**.

The filter propagation will now follow the only active path.

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



15. Switch to Report view, and then notice that the sales for Michael Blythe are 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

16. 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 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** lab.*

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

18. 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.

Relate the Targets table

In this task, you'll 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 aren't additive, and so the total shouldn't 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 Advanced DAX Calculations in Power BI Desktop** lab that will return BLANK when more than one salesperson is filtered.*

Finish up

In this task, you'll complete the lab.

Save the Power BI Desktop file, and select **Apply Later** if prompted to apply queries.

