**Exercise: Adding a role-playing dimension**

**Introduction**

At this stage, you’ve reviewed the basics of role-playing dimensions and the application of the **USERELATIONSHIP** function in **DAX** when handling the inactive relationship within a data model in **Power BI**.

This exercise asks you to apply your knowledge of these concepts by configuring a role-playing dimension between two tables in Power BI.

By completing this exercise, you will demonstrate your ability to:

* Configure a table as a role-playing dimension by creating multiple relationships between the tables.
* Create measures by writing **DAX** expressions using **USERELATIONSHIP** and **CALCULATE** functions.

**Scenario**

Adventure Works needs your help to analyze its sales data based on the shipping dates for a specific month. However, its data model does not have a separate **Shipping date** table. So, you'll need to configure the **Date** dimension and create a measure for the total sales for August.

The company provides you with an Excel file called *AdventureWorksData.xlsx*. The file consolidates all required data into a table containing all relevant fields related to the company’s sales data.

You must load this dataset into Power BI and add a role-playing dimension so that Adventure Works can generate the required insights.

[AdventureWorksData](https://d3c33hcgiwev3.cloudfront.net/o87LBIKqSLiM_8u1yV_7tA_2f3d771465d44a36b821f09f6cee39e1_AdventureWorksData.xlsx?Expires=1711152000&Signature=RiWJKL1b4w81VAPFzjpyzEvkPm6aB2D562z6rUlWYxNE6FIwX0tp23oLw8trb0aScNXlcUaVpZdbk9C0TdGiuwvwFlIsygLhufNufI8RLik0Ji37Igwq-P8M3HfBpVBxNaQhsIJdKnF7Ofd8oD79MYhGkl95TZE-qVQUk6Xi7~A_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

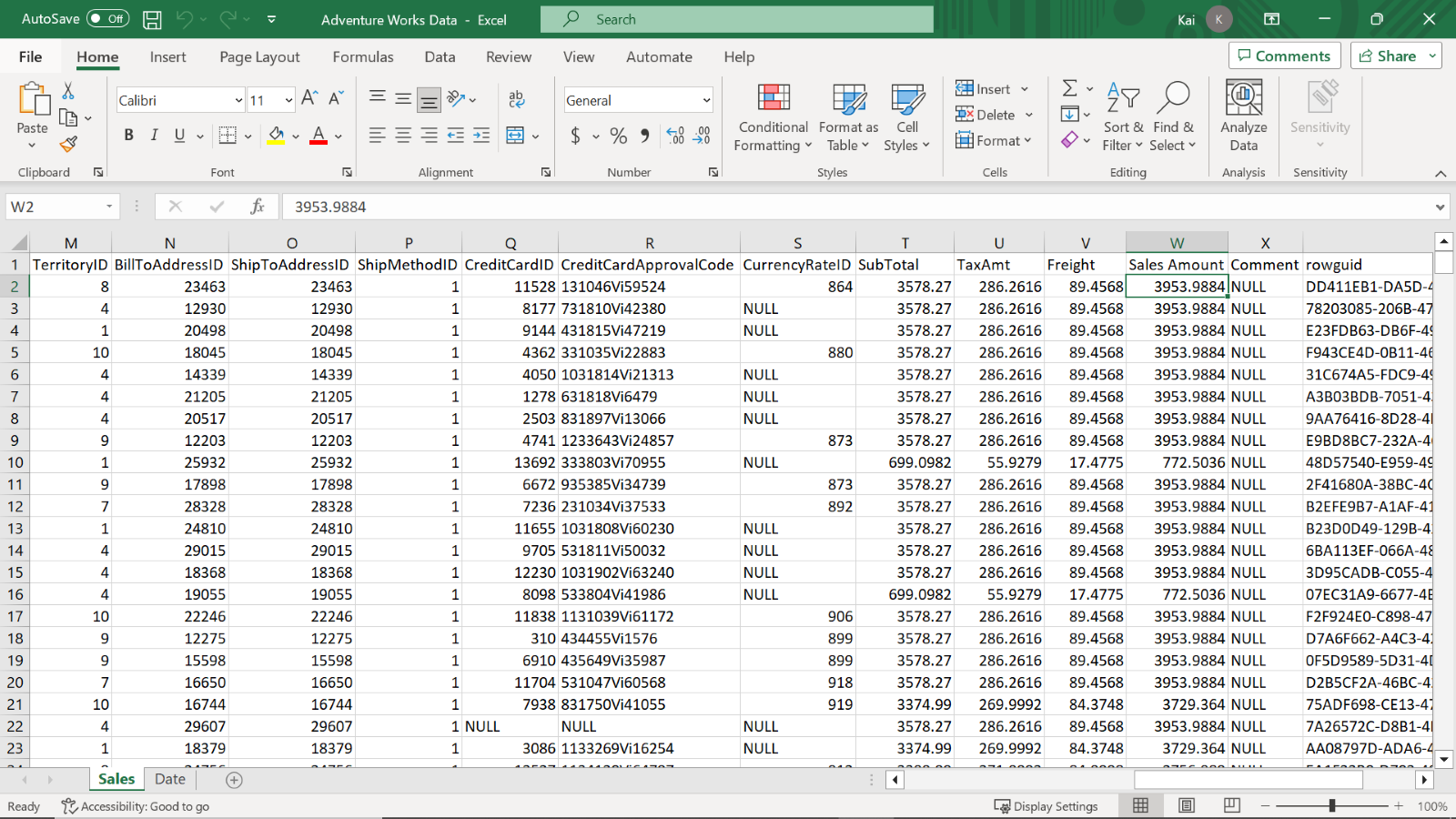
[XLSX File](https://d3c33hcgiwev3.cloudfront.net/o87LBIKqSLiM_8u1yV_7tA_2f3d771465d44a36b821f09f6cee39e1_AdventureWorksData.xlsx?Expires=1711152000&Signature=RiWJKL1b4w81VAPFzjpyzEvkPm6aB2D562z6rUlWYxNE6FIwX0tp23oLw8trb0aScNXlcUaVpZdbk9C0TdGiuwvwFlIsygLhufNufI8RLik0Ji37Igwq-P8M3HfBpVBxNaQhsIJdKnF7Ofd8oD79MYhGkl95TZE-qVQUk6Xi7~A_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

**Instructions**

Create a new Power BI project called *Exercise –Adding a role-playing dimension*. Follow the steps below to complete the exercise.

**Step 1: Download and connect to the Adventure Works Dataset.**

1. Download and save the Excel workbook *AdventureWorksData.xlsx*. The workbook contains two tables of data: **Sales** and **Date.**



1. Load the data from the Excel sheet into **Power BI**. Ensure you load all tables in the workbook.

**Tip:** You can import data using the **Get Data** drop-down menu.

**Step 2: Review the data model and establish relationships.**

1. Ensure an appropriate relationship between the **Fact** table and the **Date** dimension table is established. Once the data has been loaded, **Power BI** will attempt to establish the relationship between the tables. If the relationship is missing, create a manual relationship between the **Sales** and **Date** table based on the **Order date**. This must be an active relationship.
2. Create another relationship between the **Sales** and **Date** table based on the **Shipping date** column from the **Sales** table. This must be an inactive relationship as the **Date** table is the role-playing dimension in the data model.

**Tip:** You can view and configure model relationships in **Model view** of Power BI desktop. You can also create and edit relationships in **Manage Relationship** of Power BI desktop.

**Step 3: Create a measure by writing a DAX expression.**

1. Once you configure the **Date** table as a role-playing dimension and establish the relationship in the data model, create a new measure called **August Sales by Shipping date**. You must filter the total sales by month and apply the **USERELATIONSHIP** function to override the active relationship between the **Sales** and **Date** tables.
2. Format the measure as currency with 2 decimal places.

**Tip:** You can create this measure using the **CALCULATE** and **USERELATIONSHIPDAX** functions in the formula bar of **Power BI** desktop interface.

**Step 4: Save the Power BI project.**

Save your Power BI project to your local computer.

**Tip:** Make sure you select an appropriate project name and folder path.

**Conclusion**

This exercise provides you with hands-on experience of configuring role-playing dimensions and handling inactive relationships within a data model in Power BI. You can assist Adventure Works with analyzing its data from various unique perspectives without creating redundant data tables.

# **Exemplar: Adding a role-playing dimension**

**Overview**

In the exercise *Adding a role-playing dimension,* you were asked to configure the **Date** table as a role-playing dimension for the **order date** and **shipping date**. Your task was also to create a measure to calculate the total sales for August based on the **shipping date** by using the **USERELATIONSHIP** function in **DAX**.

Your tasks in this exercise were to:

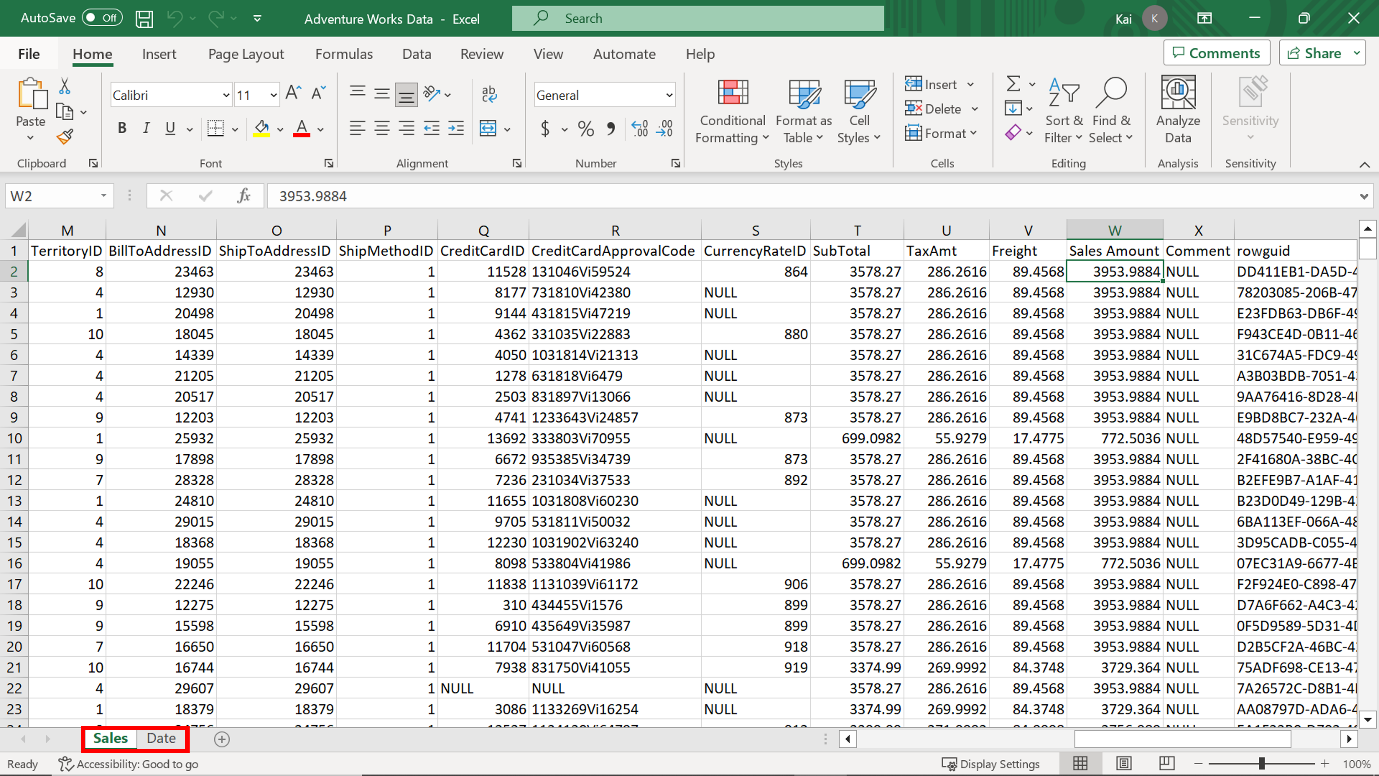
* Download and connect to the required dataset.
* Create an active and inactive relationship between the **Sales** and the **Date** table.
* Create a measure within your data model by overriding the default relationship.

This reading provides you with a step-by-step guide for completing these tasks. It also includes screenshots that you can compare against your work.

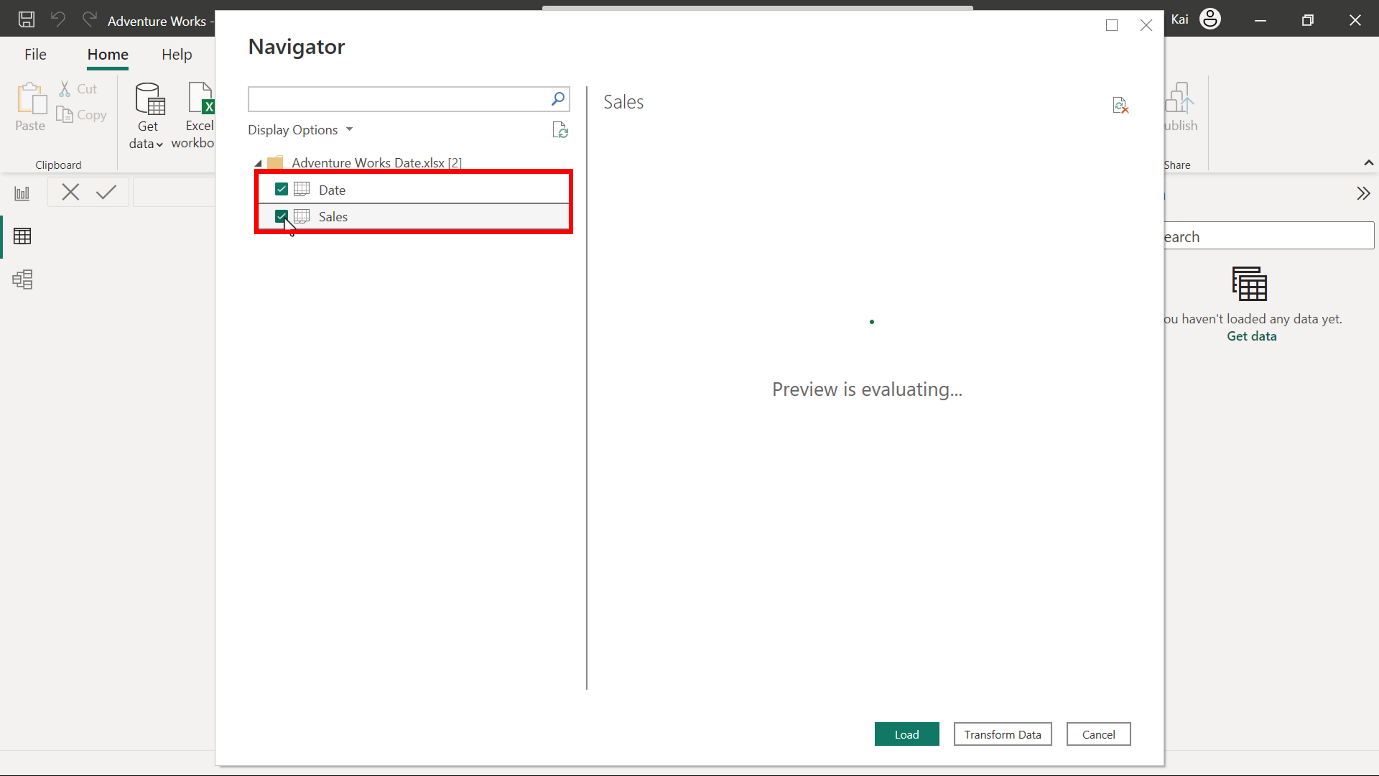
You can also review [*Creating quick measures*](https://www.coursera.org/learn/data-modeling-in-power-bi/lecture/xZZBs/creating-quick-measures) and [*Creating custom measures with DAX*](https://www.coursera.org/learn/data-modeling-in-power-bi/lecture/m25v9/creating-custom-measures-with-dax).

**Step 1: Download and connect to the Adventure Works dataset.**

1. Download and save the workbook **Adventure Works Date.xlsx**. The workbook contains two tables of data: **Sales** and **Date**.

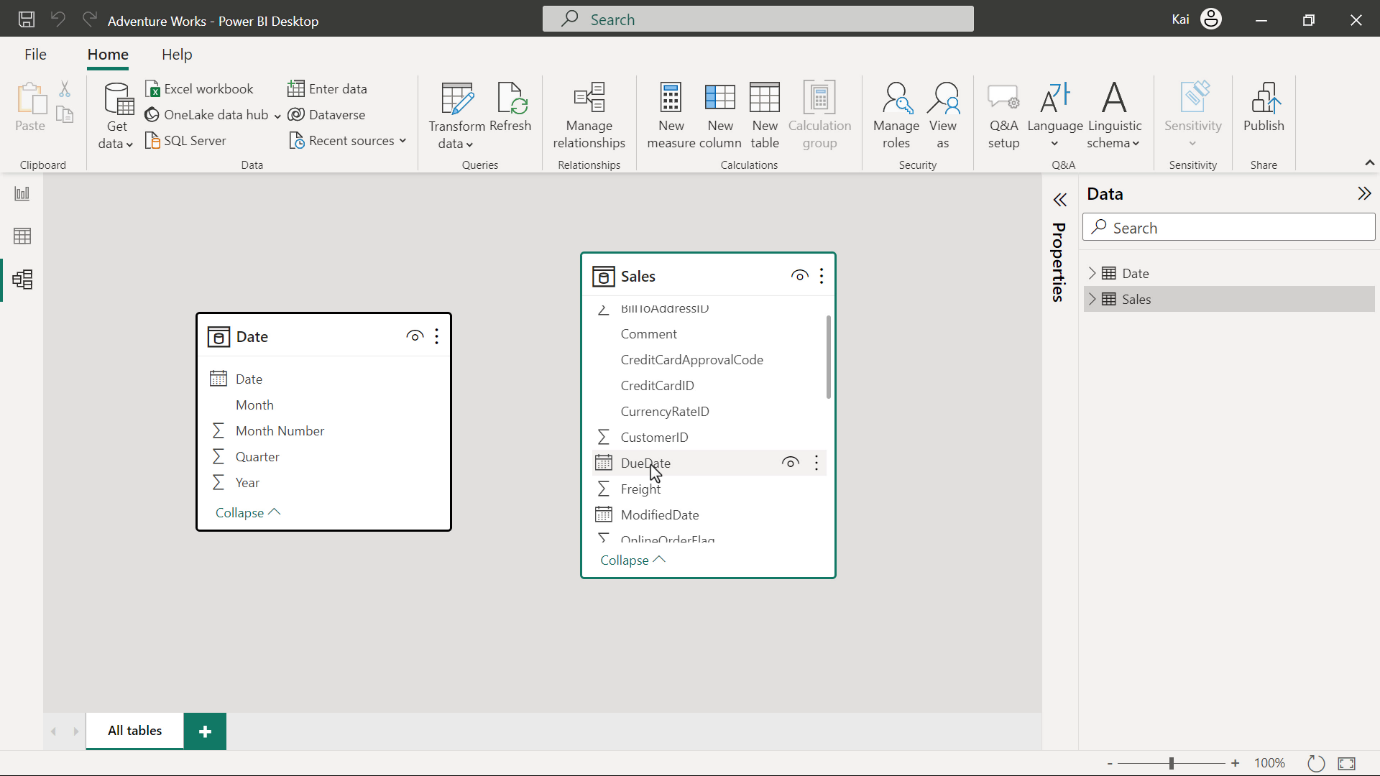


2. Load the data to **Power BI,** ensuring you load both data tables available in the workbook to the data model, that is the **Sales** table and the **Date** table.



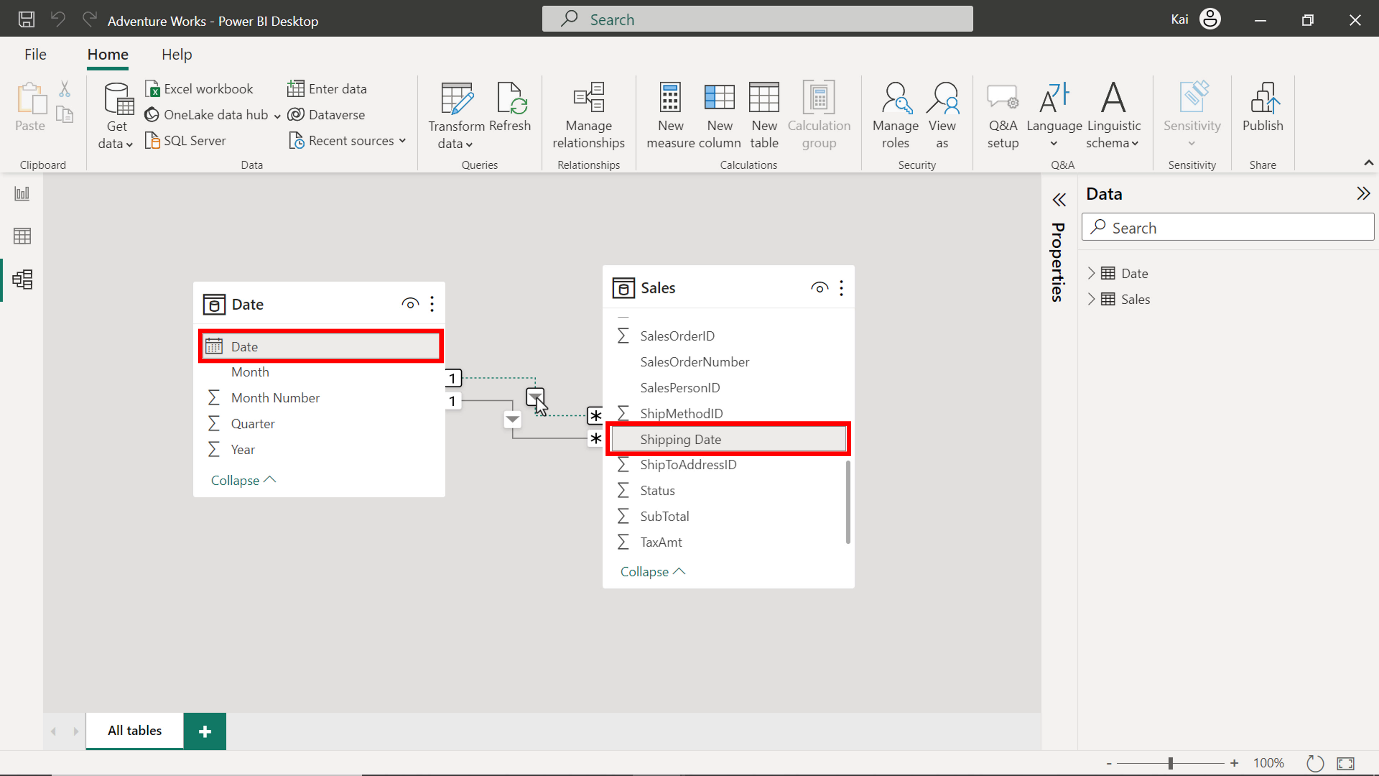
**Step 2: Review the data model and establish relationships.**

1. Make sure there is an appropriate relationship between the **Fact** table and the **Date dimension** table.



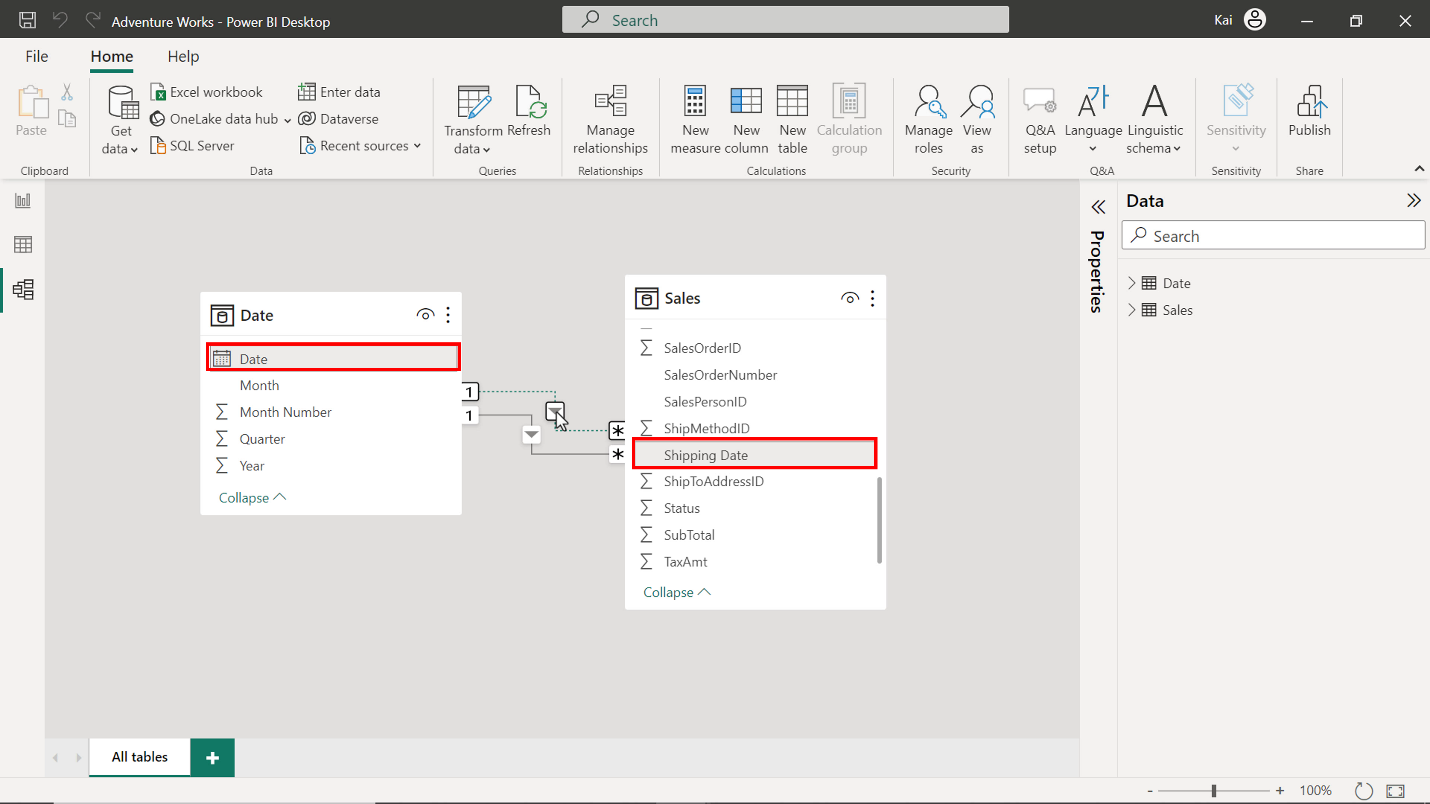
After you Load the data, **Power BI** attempts to establish the relationship between the tables. If the relationship is missing, create a manual relationship between the sales and date table based on the order date. That is the active relationship.

You can drag and drop the **order date** field from the **Date** table to the **order date** field in the **Sales** table. Alternatively, navigate to **Manage relationship** option from the **model view** of **Power BI** desktop. This opens the **Manage relationship** dialog box. Select **New** to create a new relationship.



1. Create another relationship between the **Sales** and **Date** table based on the **Shipping date** column from the **Sales** table. This will be an inactive relationship as the **Date** table is the role-playing dimension in the data model.

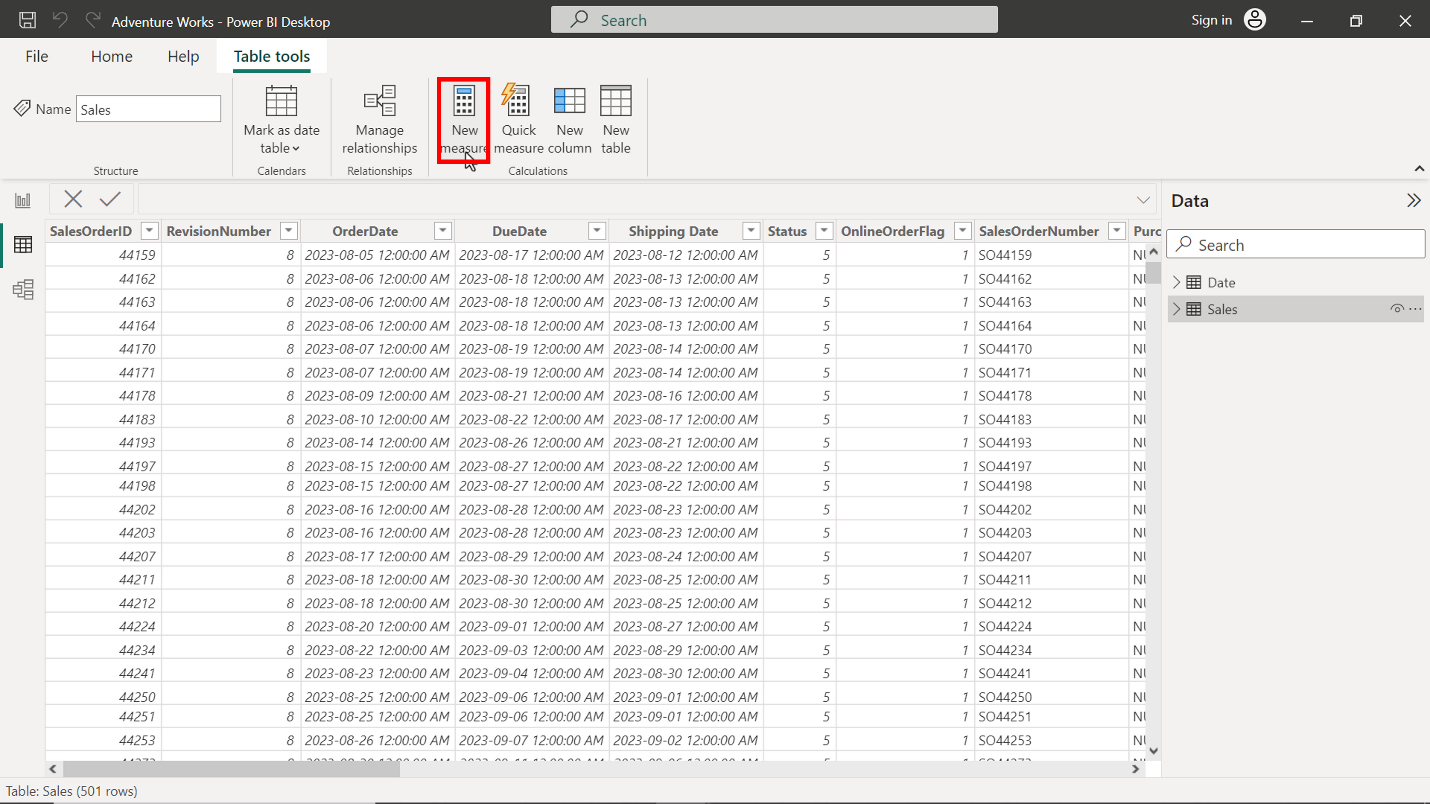
You can repeat the drag-and-drop process. Select the **Shipping date** column from the **Sales** table, then drag and drop it to the **Date** column of the **Date** table. The dashed line between the **Date** and the **Sales** table represents this relationship.



**Step 3: Create Measure by writing DAX expression.**

1. Once you configure both active and inactive relationships in the data model, create a new measure called **August Sales by Shipping date**.

Go the **Data view.** Under the **Datepane**, select the **Sales table** and then the **New measure** option from the **calculations group**. This action expands the DAX formula bar. Add the DAX expression to compute the measure **August sales by shipping date**.



1

2

3

4

5

6

7

August Sales by Shipping date =

CALCULATE

(

    SUM ( Sales[Sales Amount] ),

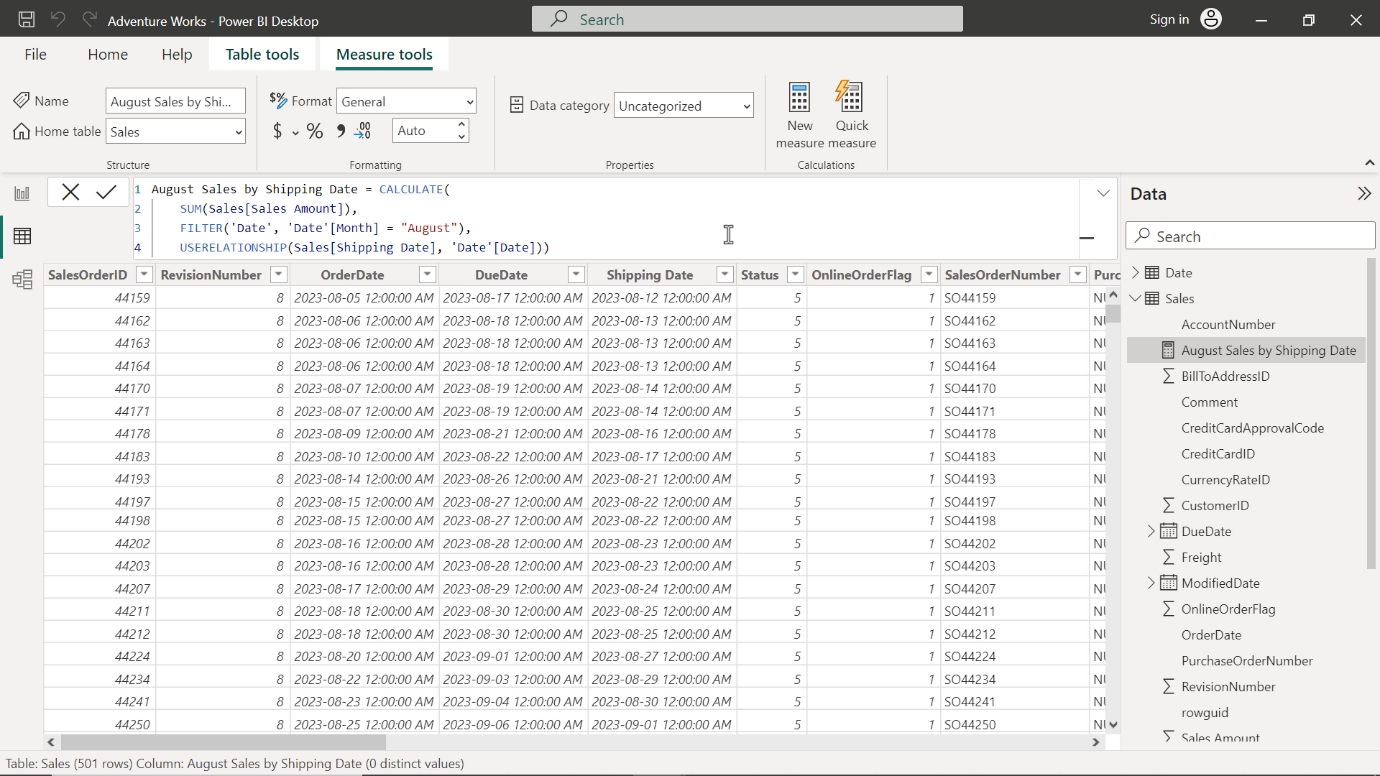
    FILTER ( 'Date', 'Date'[Month] = "August" ),

    USERELATIONSHIP ( Sales[Shipping Date], 'Date'[Date] )

)

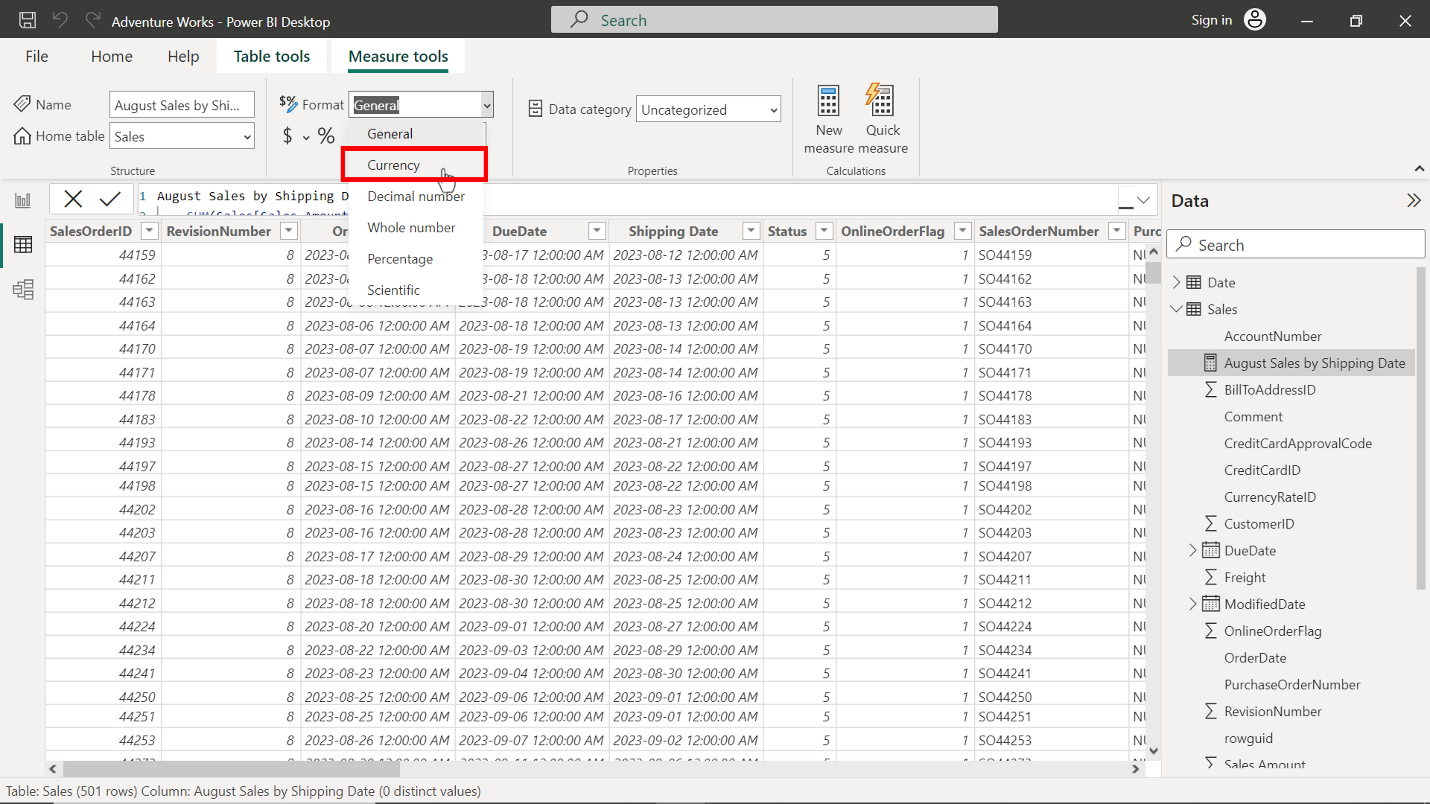
* The expression calculates the total sales for August based on the shipping date.
* **SUM** calculates the total **Sales** column from the **Sales** table
* **FILTER** filters the values for August from the **Month** column of the **Date** table
* **USERELATIONSHIP** overrides the table relationship to consider the **shipping** date instead of the **order date,** which is the default relationship.

Once you execute the code, a new measure appears in the data pane under the **Sales** table.

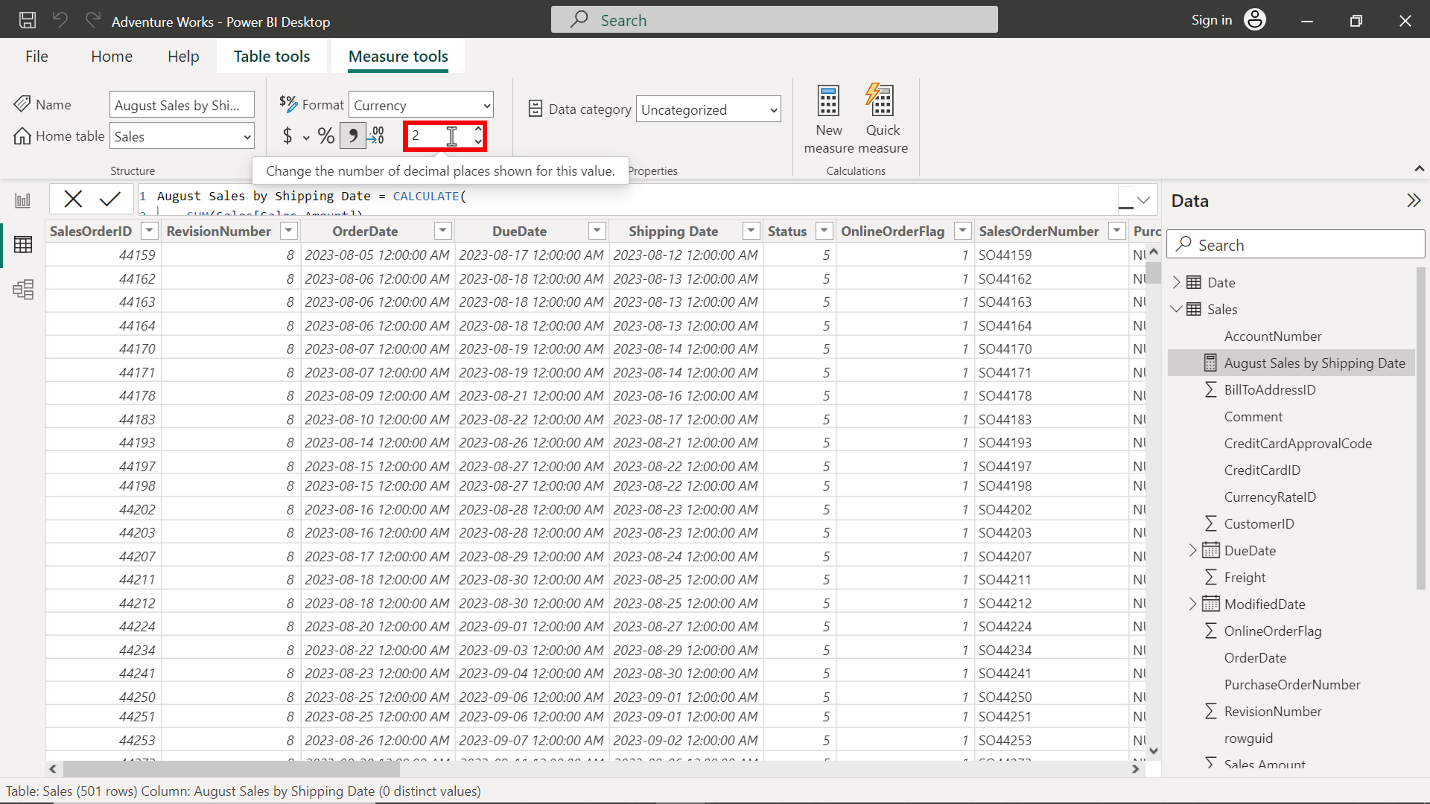


1. Format the measure as currency with 2 decimal places.

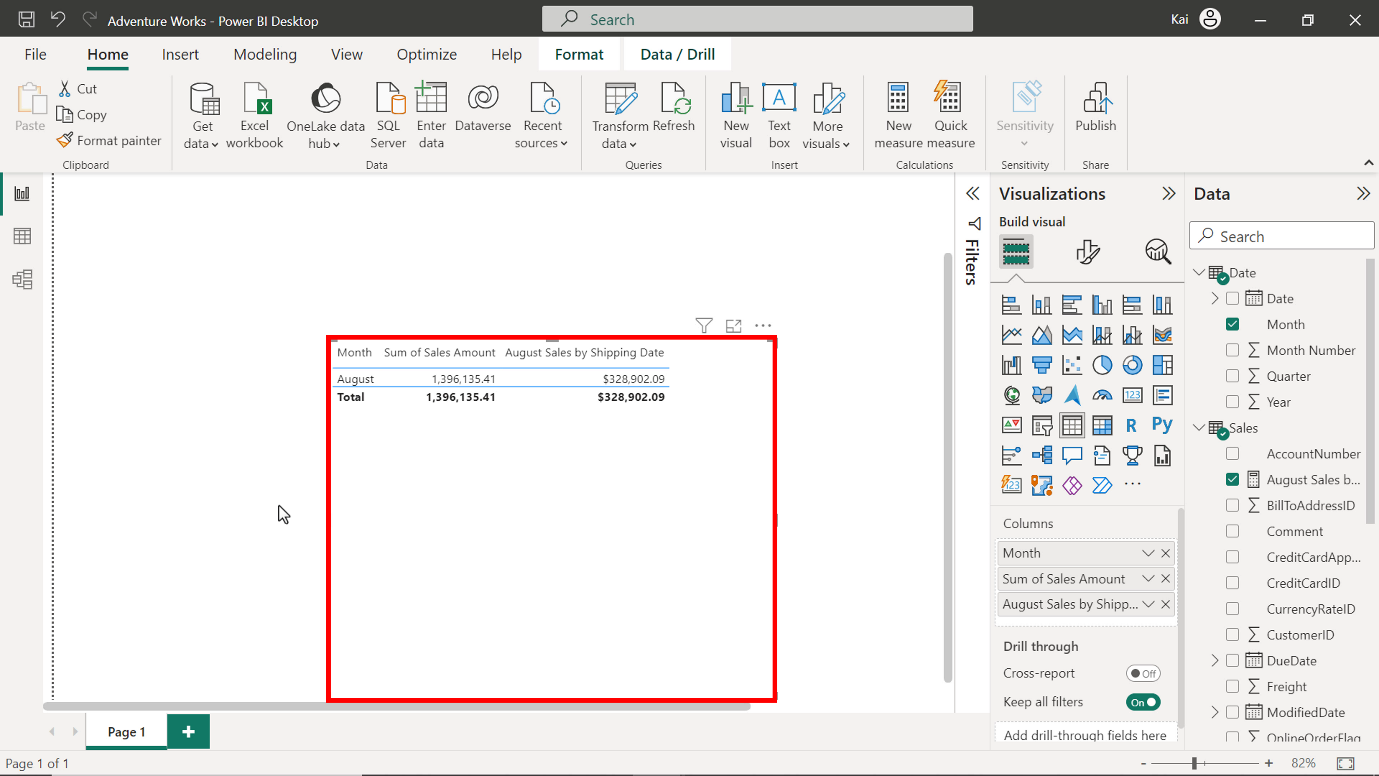
To format the measure, select the newly created measure from the data pane. Navigate to the formatting group in the **Measure tools** tab of Power BI. Select **Currency** from the **Format** drop-down menu.



Enter **2** in the decimal places (which is auto by default). This action formats the measure as **Currency** within **2** decimal places and is good for visualization.



You can view the results of the measure in the following diagram:



**Step 4: Save the Power BI project.**

* To save the project, open the **File** menu, select **Save As,** and provide an appropriate name for the project along with a path to the folder on your computer.

**Conclusion**

With these steps, you have successfully created one quick measure using the **Power BI** desktop interface and another by writing a DAX query. You can now analyze Adventure Works data based on the analytical and business requirements.

Remember that when using DAX formulas, always ensure they are correctly formatted and that the column names match the actual column names in your data.