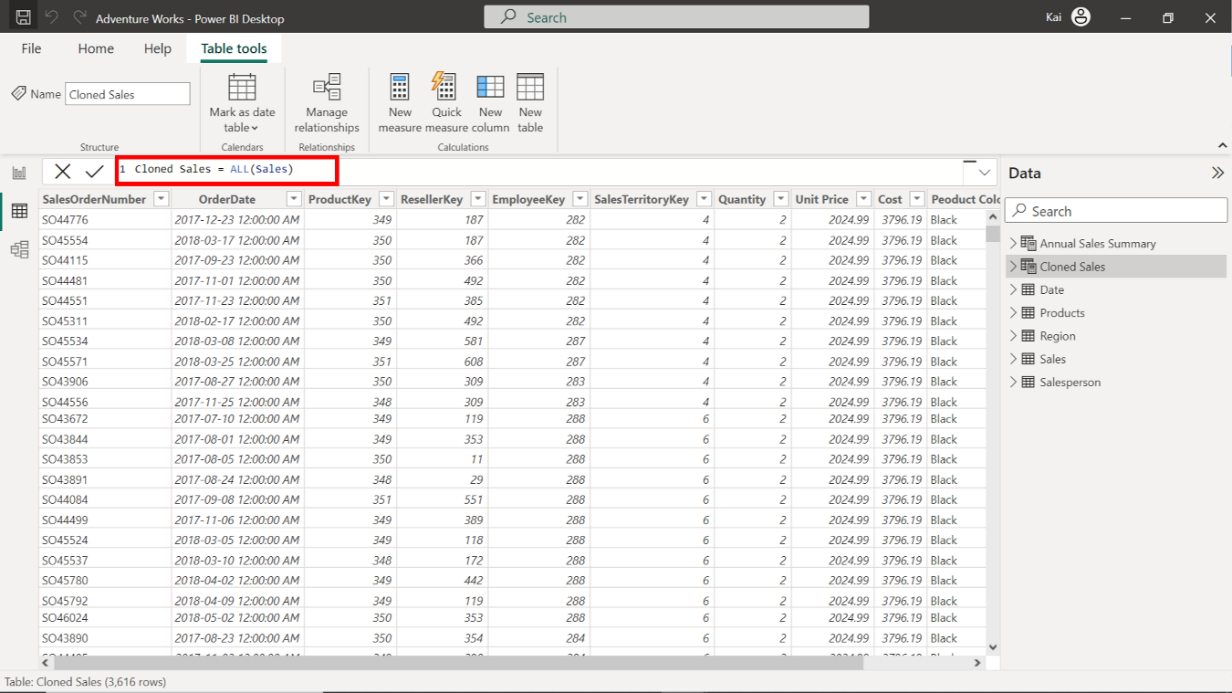
**Cloned and calculated tables**

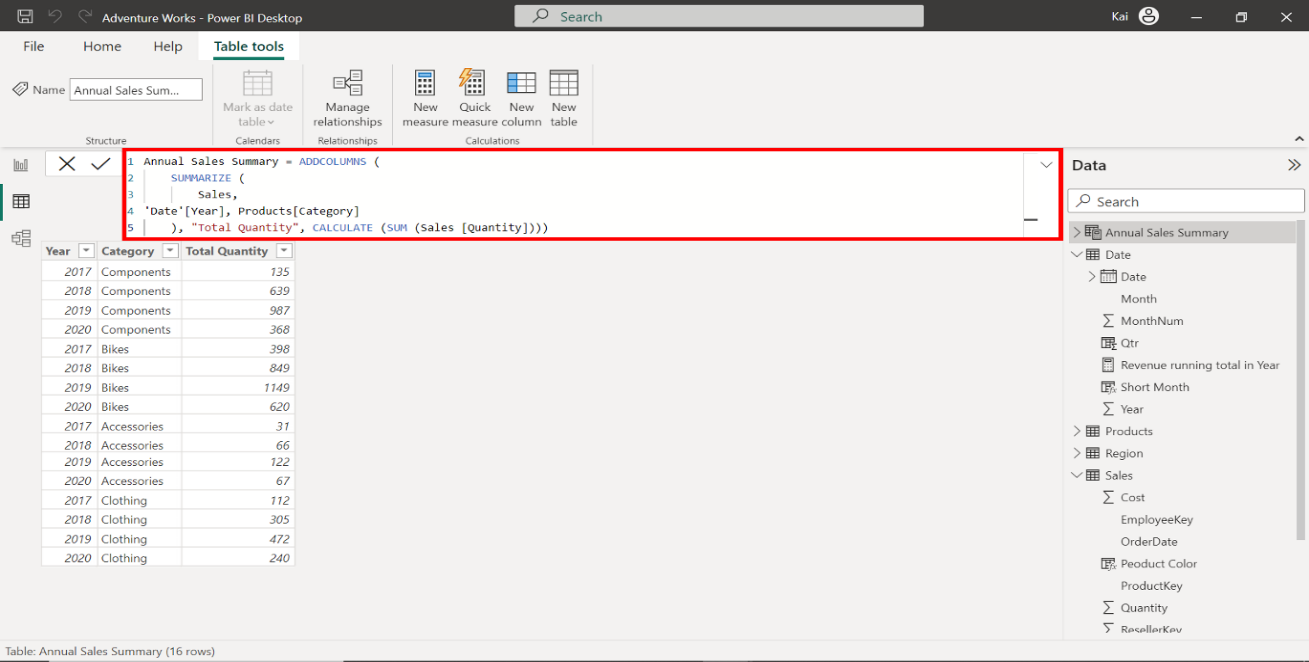
**Cloned tables**

Cloning a table means duplicating the original table. In other words, creating an exact copy. The cloned table inherits all the original table's columns, data, and relationships. A cloned table is useful if you want to perform manipulations or analysis on data while preserving the original records.



**Calculated tables**

Calculated tables are not simple duplicates. Instead, they are created based on calculations, transformations, or aggregations performed on existing data. For instance, you can create a calculated table showing each bicycle model's total sales.



**Real-World applications of calculated tables**

There are many real-world applications of calculated tables. Here are some examples of how you can make use of calculated tables.

**Profitability Analysis**

By creating a calculated table for profitability analysis, you can draw insights into gross margins, net profits, and profit margins. This helps organizations identify their most profitable products, categories, services, and customer segments.

**Customer Segmentation**

Understanding customer behavior is crucial for marketing efforts. You can create a calculated table with DAX to facilitate customer segmentation based on transaction history. This helps businesses to tailor their marketing strategies for each customer segment.

**Time Intelligence Analysis**

A date table is one of the most common tables that can be created using various methods in Power BI. One method of creating a date table is using the DAX function **CALENDER**. For example, a company can record several date columns in their dataset, like order date or shipping date. The calculated table can then analyze the data using time intelligence built-in DAX functions like **TOTALYTD**, **year-to-date**, **month-to-date**, and so on.

**Budgeting and Forecasting**

By defining a calculated table in DAX for budget allocation and integrating it with historical data, you can perform variance analysis and make data-driven forecasts for future periods.

**Best practices for creating calculated tables with DAX**

Here are a few best practices for creating calculated tables to ensure they’re optimized, readable, and consistent.

**Optimize DAX Formulas**

Complex calculations slow down model performance, so optimize the formula according to your needs.

**Use Variables**

Using variables is an excellent way to enhance formula readability. Variables are recommended wherever you need to write a complex expression. By defining variables, you can avoid repeating the same expression.

Take the following expression, which aims to calculate the sales growth of Adventure Works as an annual percentage:

Sales YoY Growth % =

DIVIDE ((

    [Sales] - CALCULATE ( [Sales],

    PARALLELPERIOD ( 'Date'[Date], -12, MONTH ) )),

    CALCULATE ( [Sales],

    PARALLELPERIOD ( 'Date'[Date], -12, MONTH ) ))

The above formula uses the **PARALLELPERIOD** function to compute the year-over-year growth rate of the company.

This formula repeats the expression **same period last year**. It can be made more efficient by introducing a variable called **SalesPriorYear** as follows:

Sales YoY Growth % = VAR SalesPriorYear =

CALCULATE ( [Sales],

PARALLELPERIOD ( 'Date'[Date], -12, MONTH ) )

RETURN

DIVIDE ( ( [Sales] - SalesPriorYear ), SalesPriorYear )

Variables are useful to improve code readability, better performance, and easier debugging.

**Format DAX syntax**

Formatting DAX formulas and expressions is crucial for maintaining consistency and readability. When working in a team, format your syntax to enhance comprehension and simplify troubleshooting. For example, consider the following syntax:

Total Revenue = CALCULATE(SUM(Sales[Revenue]),

FILTER

(Sales, Sales[OrderDate]=2018

&&

Sales[Product Color]="Blue"))

The above DAX expression calculates the total sales of blue-colored products for 2018.

The syntax is complex. It contains many arguments and is hard to comprehend. On formatting the syntax, it looks like this:

Total Revenue =

CALCULATE

( SUM ( Sales[Revenue] ),

FILTER ( Sales, Sales[OrderDate] = 2018

&&

Sales[Product Color] = "Blue" ))

It became instantly readable, easy to follow along with each function, and the argument is broken down into a new line. Using line breaks and tabs makes the syntax comprehensible and is especially important when working in a team.

**Test and Validate**

Always test and validate your calculated tables to ensure they produce the desired results.

**Conclusion**

Powered by DAX in Power BI, calculated tables enhance data analysis by providing a flexible and powerful tool for creating custom tables with calculated values.

They offer benefits like time intelligence functions, profitability assessment, and customer segmentation. Use these benefits to generate real-world insights into the hidden values of the datasets.

By mastering DAX, you can unleash the full potential of data and gain a competitive advantage in today’s data-centric business landscape.

**Exemplar: Adding a calculated table and column**

**Overview**

In the exercise *Adding a calculated table and column*, you were asked to create new calculated tables and columns using DAX within your data model to address specific analytical and visualization concerns.

Your tasks in this exercise were to:

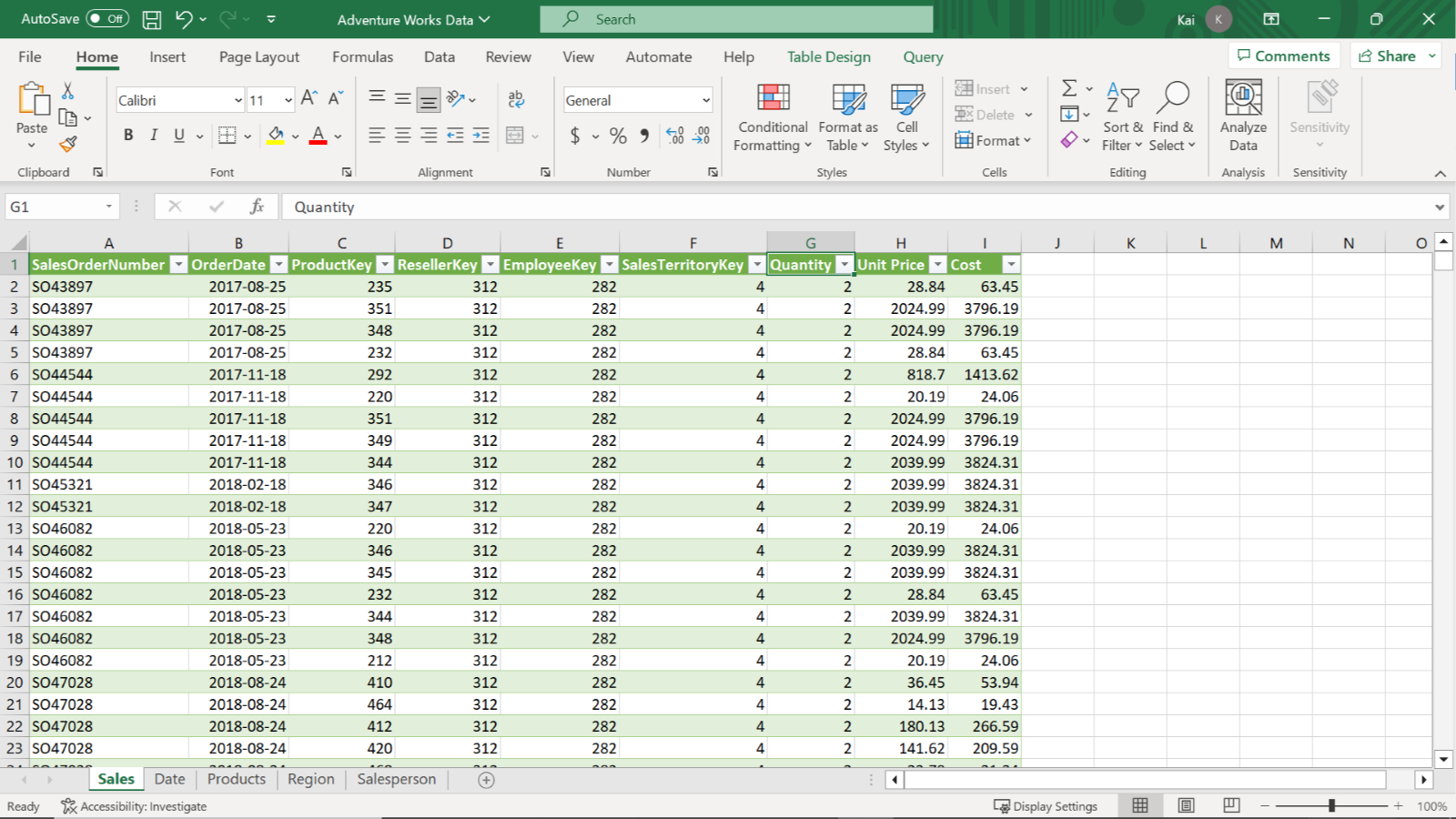
* Create a calculated table from the existing dataset within your data model.
* Add calculated columns to a specific table within the dataset.
* Ensure data standardization and consistency.

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.

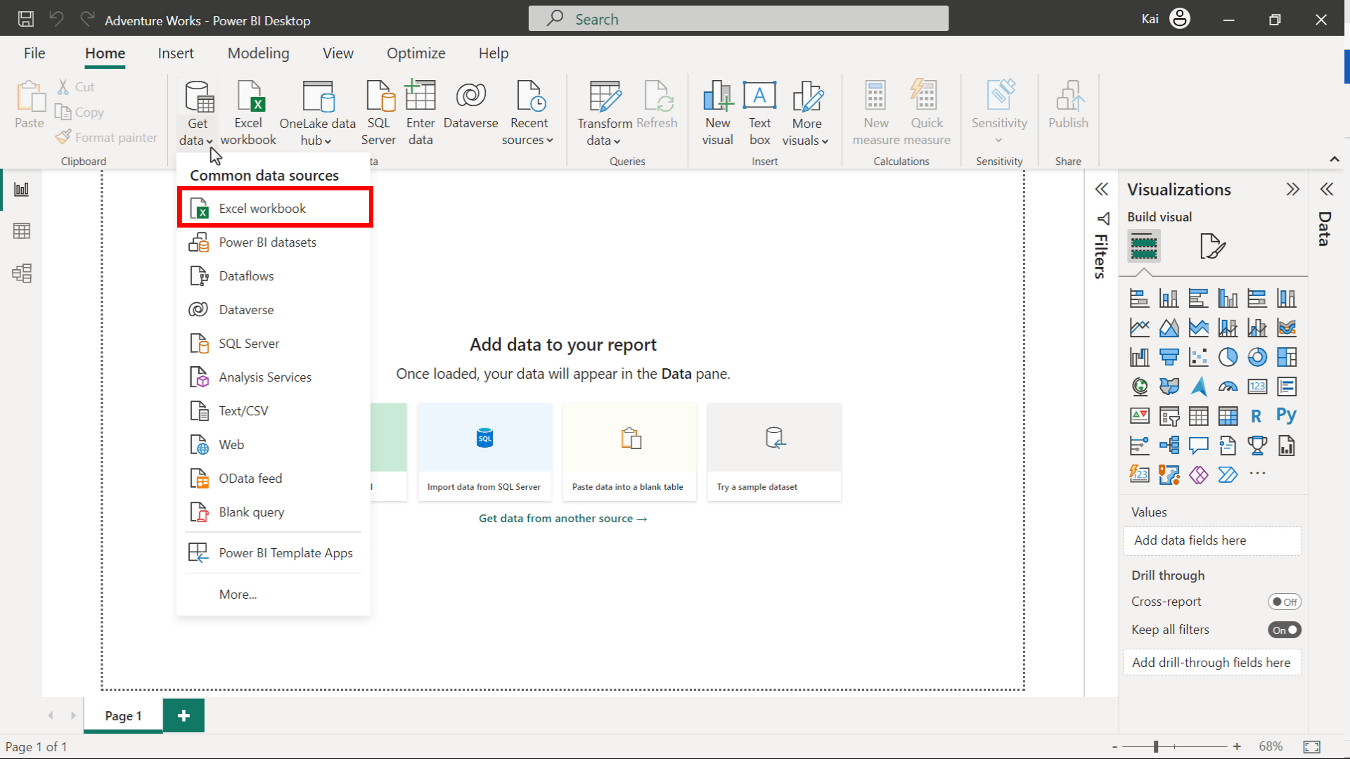
You can also review the [*Introduction to calculated tables*](https://www.coursera.org/learn/data-modeling-in-power-bi/lecture/OLf8s/introduction-to-calculated-tables) and [*Creating calculated columns*](https://www.coursera.org/learn/data-modeling-in-power-bi/lecture/Jkzfj/creating-calculated-columns) videos for guidance on using DAX in Power BI.

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

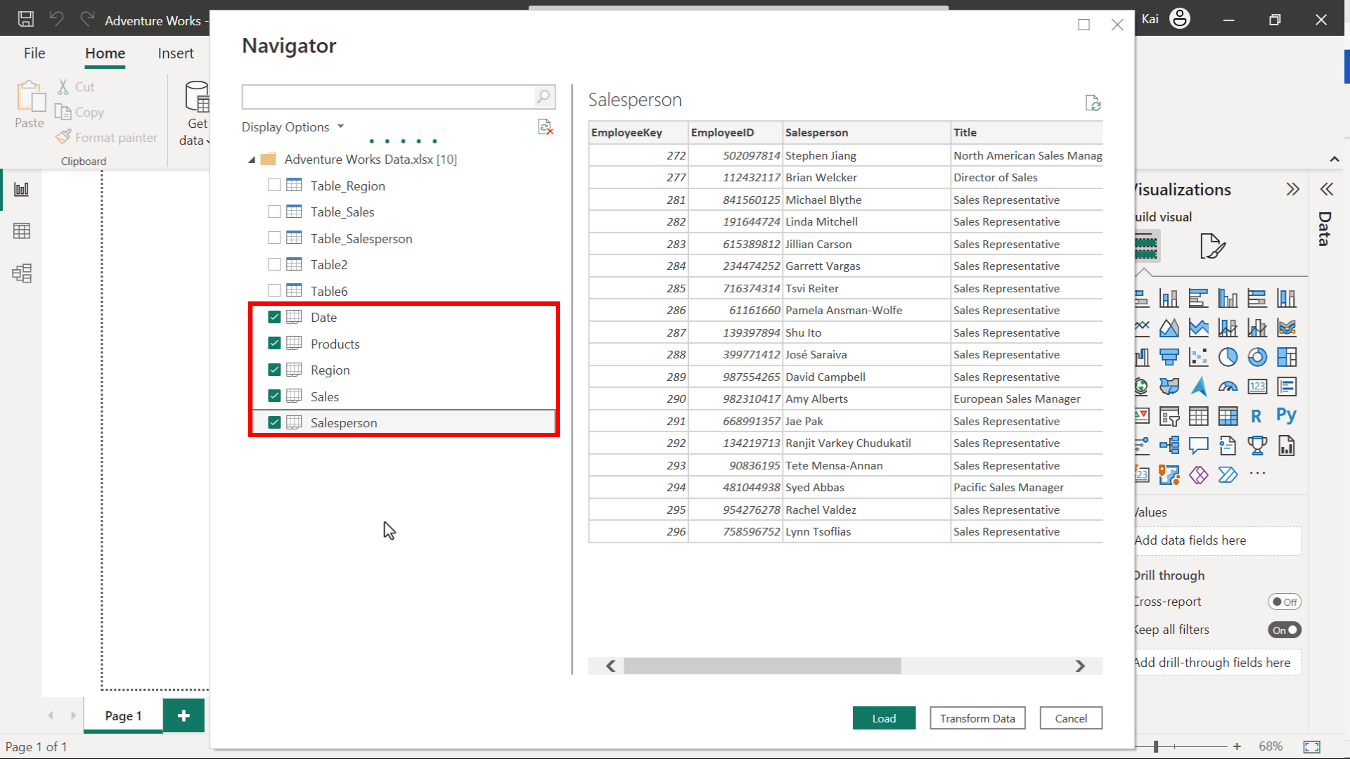
1. Download the workbook from the exercise page on the Coursera platform.



1. Launch Power BI desktop. To create a new project, select the **File** menu, then select **New**. Import the Adventure Works dataset that you have downloaded. In the **Home** tab, select the **Get Data** drop-down menu. Then select an appropriate data source. For the current exercise, select **Excel Workbook** and navigate to the Adventure Works dataset folder.

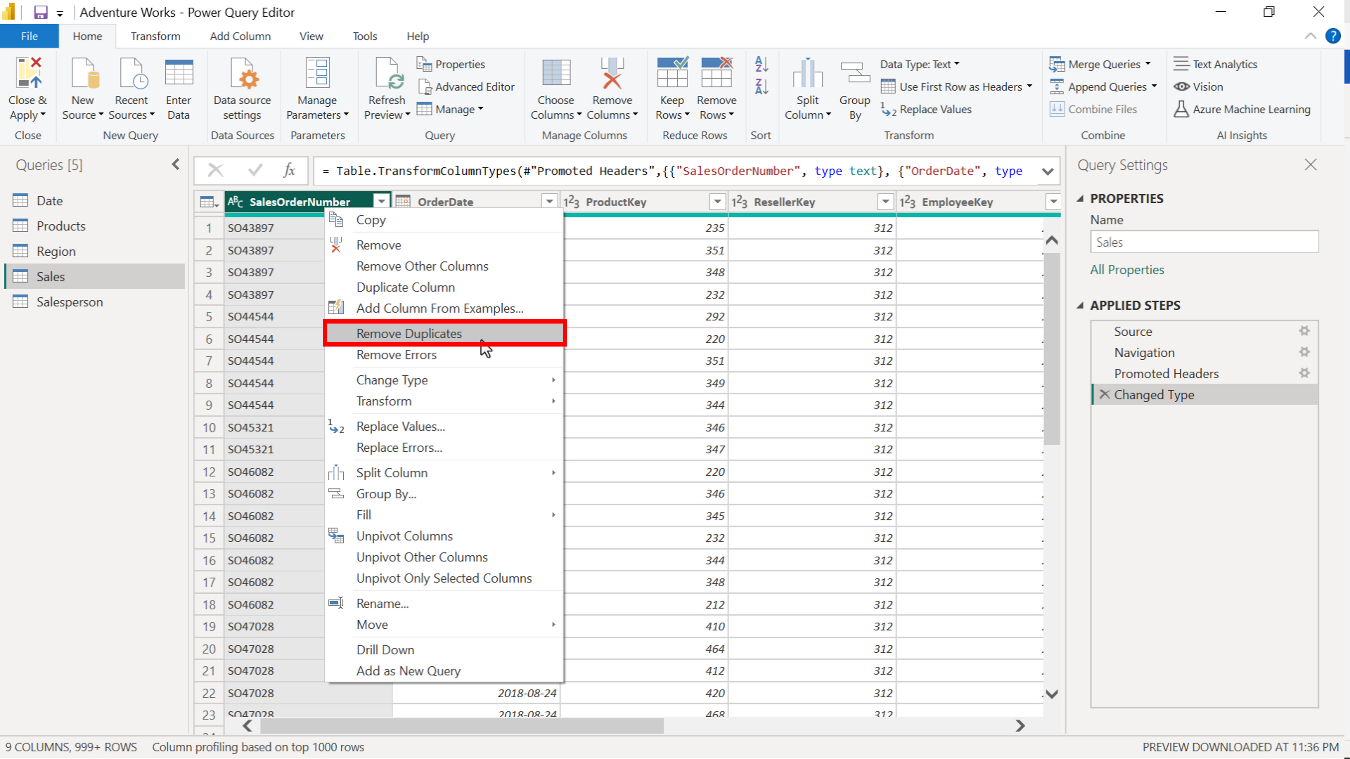


3. Once you select and load the data, Power BI opens a **Navigator** dialog box that lists all the tables available to load in the Excel file, along with the data preview on the right side of the Navigator. Select the **Sales**, **Product**, **Region**, **Date,** and **Salesperson** tables, then select **Load**.

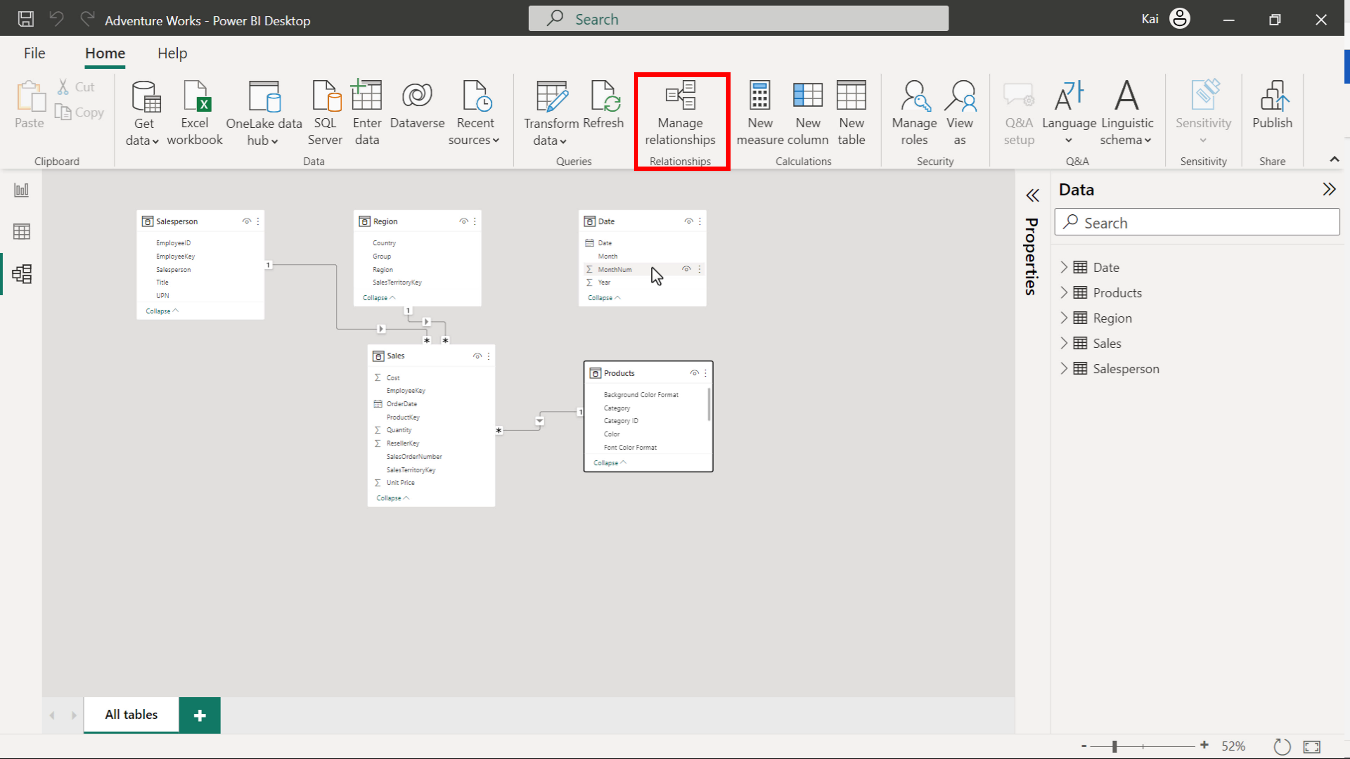


**Step 2: Remove all duplicate values and set the relationships between the tables.**

1. To eliminate all duplicate data, access the **Power Query editor**, right-click on the **SalesOrderNumber** columns, and select **Remove duplicates** from the drop-down menu.



1. To configure the model relationships, access the **Model view** of Power BI desktop and select **Manage relationships**. From here, you can edit cardinality and cross-filter direction between the tables.



**Step 3: Create a calculated table.**

1. Access the **Model view** in the calculations group to create a new table. Select **New table**. Copy and paste the following DAX code into the formula bar:

Yearly Sales by color =

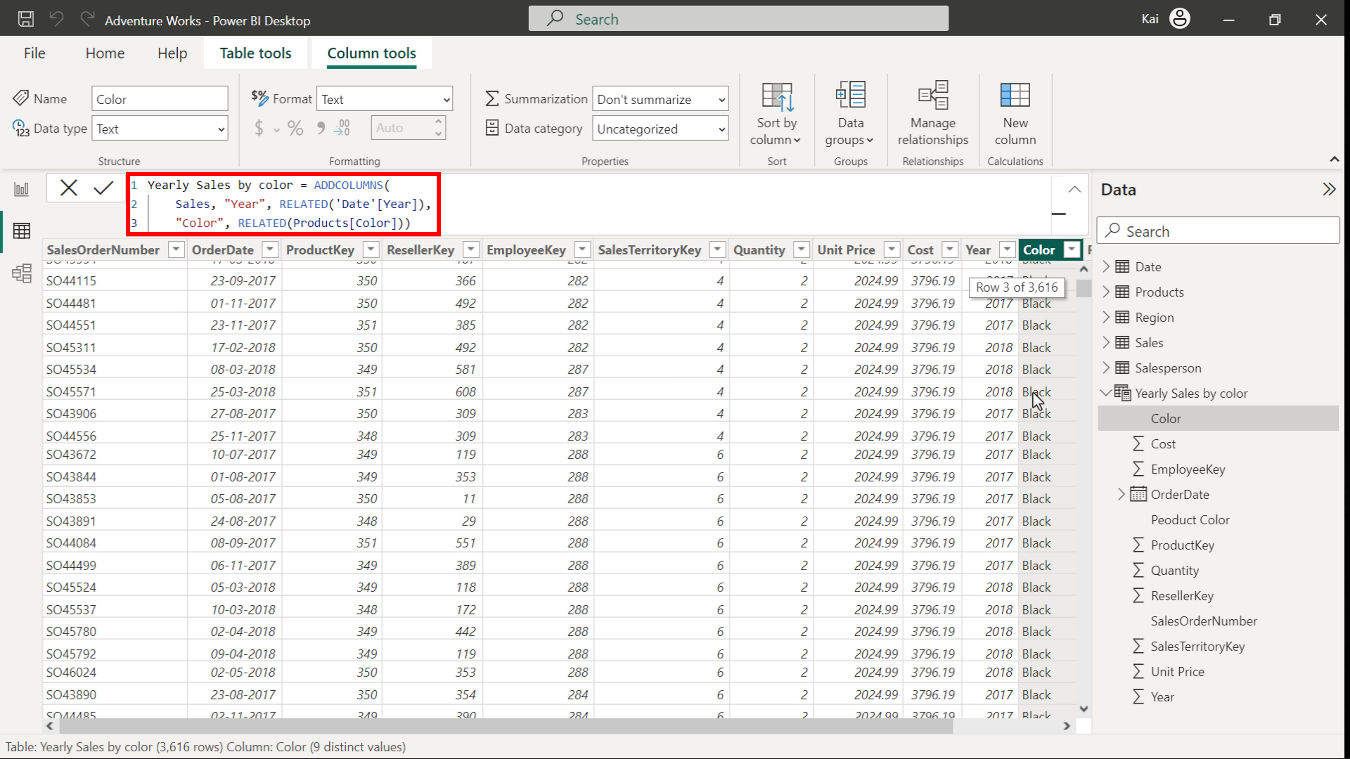
ADDCOLUMNS (

Sales,

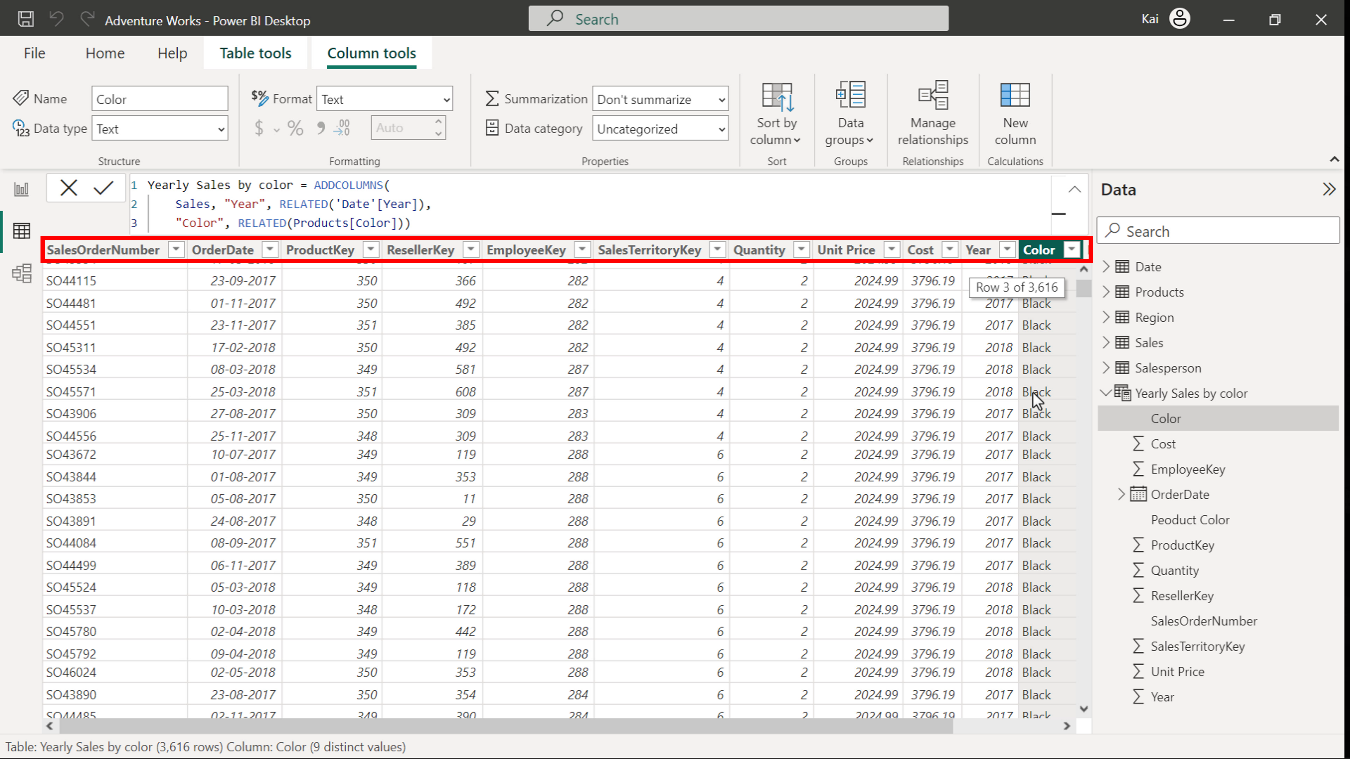
"Year", RELATED ( 'Date'[Year]),

"Color", RELATED ( Products[Color]))

* **ADDCOLUMNS**: Adds calculated columns to the given table or table expression. In this instance, the **Sales** table is the main table to which you need to add two more columns, one from the **Date** table and one from the **Product** table.
* **Year** and **Color** in double quotes are the names of the new columns to be added in the new calculated table.
* **RELATED**: Returns a related value from another table. In this case, **Product color values** from the **Product table** and **Year** information from the **Date** table.



1. Note that the resulting table has 11 columns.



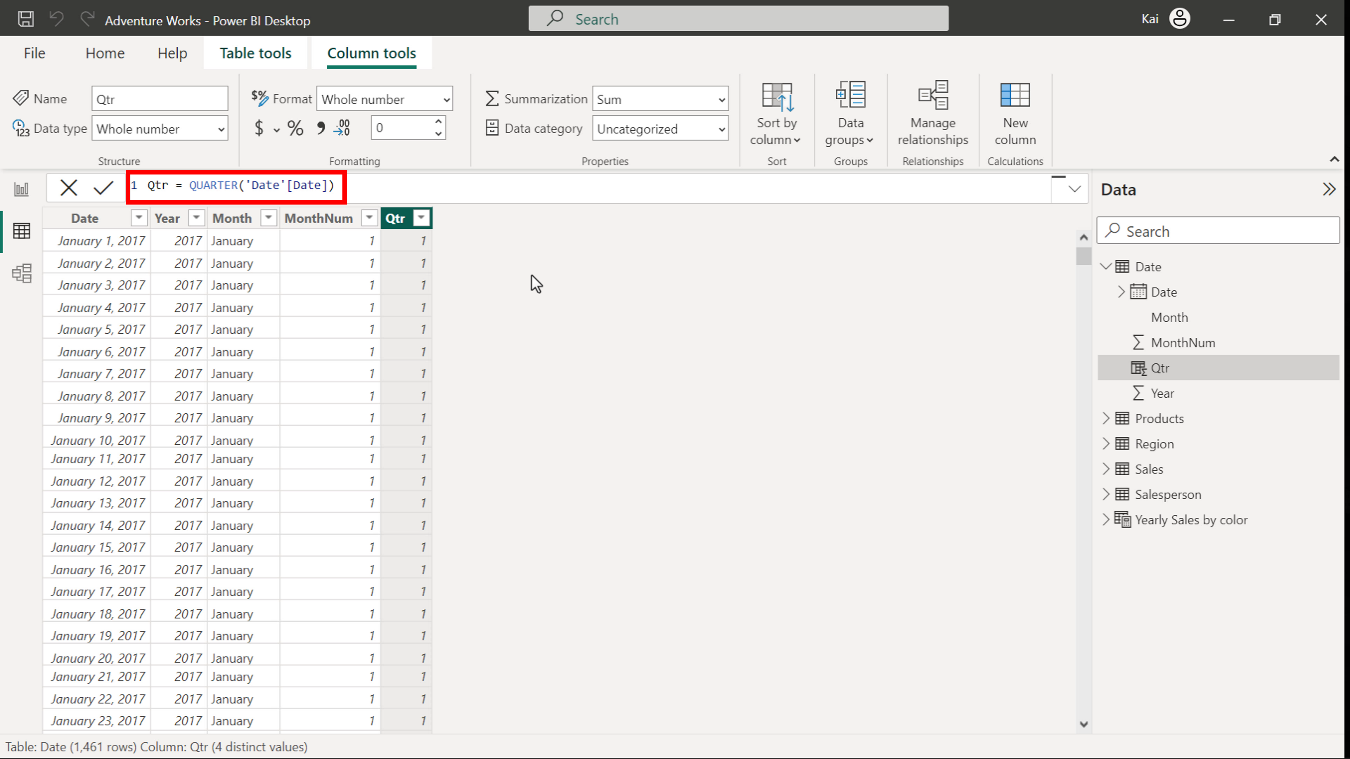
**Step 4: Create calculated columns.**

1. To create a new column, select the **Date** table from the **Data pane** on the right side of Power BI interface. Access **Model view** in the **Calculations group** and select **New column**. Copy and paste the following DAX code into the formula bar:

1

Qtr = QUARTER('Date'[Date])

* **QUARTER**: Returns each quarter as a number from the **Date** column.
* **Date** in single quotes is the table, and **Date** in square brackets is the column within the table.

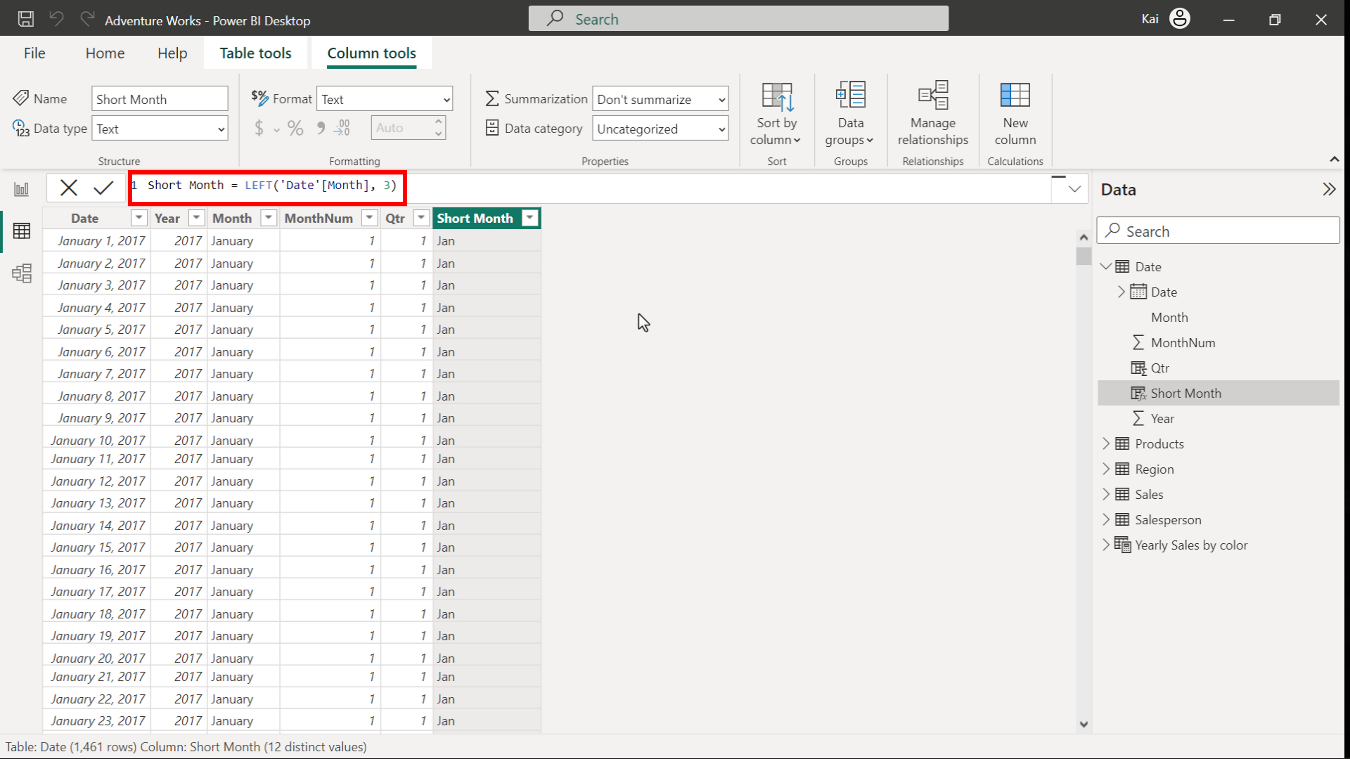


1. Select the **Date** table from the **Data pane** on the right side of Power BI interface. Access **Model view** in the **Calculations group** and select **New column**. Copy and paste the following DAX code into the formula bar:

1

Month =LEFT ( 'Date'[Month], 3 )

* **LEFT**: Returns the specified number of characters from the start of a text string.
* **Date** in single quotes is the table to be referenced, and **Month** in square brackets is the column name. The number **3** specifies the number of characters in the short month column.



1. To create a new column, select the **Product** table from the **Data pane** on the right side of Power BI interface. Access **Model view** in the **calculations group**. Thenselect **New column** to expand the formula bar. Copy and paste the following DAX code into the formula bar:

Product Color = RELATED ( Products[Color] )

* **RELATED** here is the same as referencing a column from another table.

