Microsoft Power Platform

Power BI

BETA - DAX in a Day

Lab 03

Add Calculated table and Columns

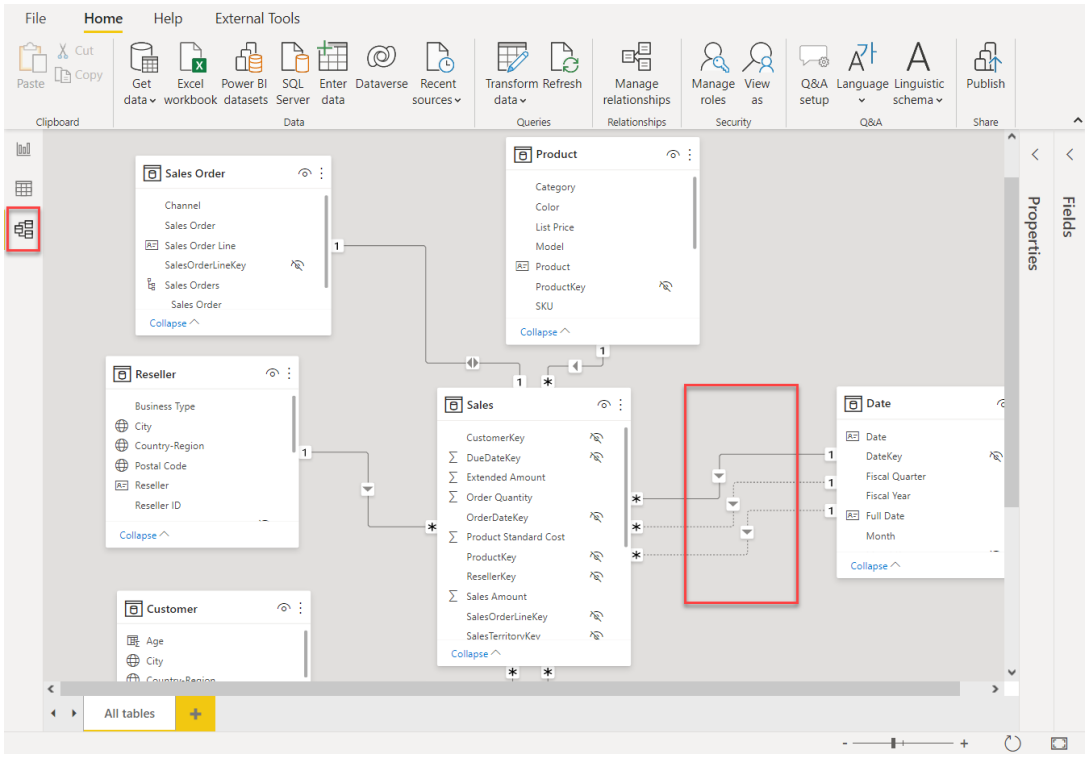
Overview

The estimated time to complete this lab is:

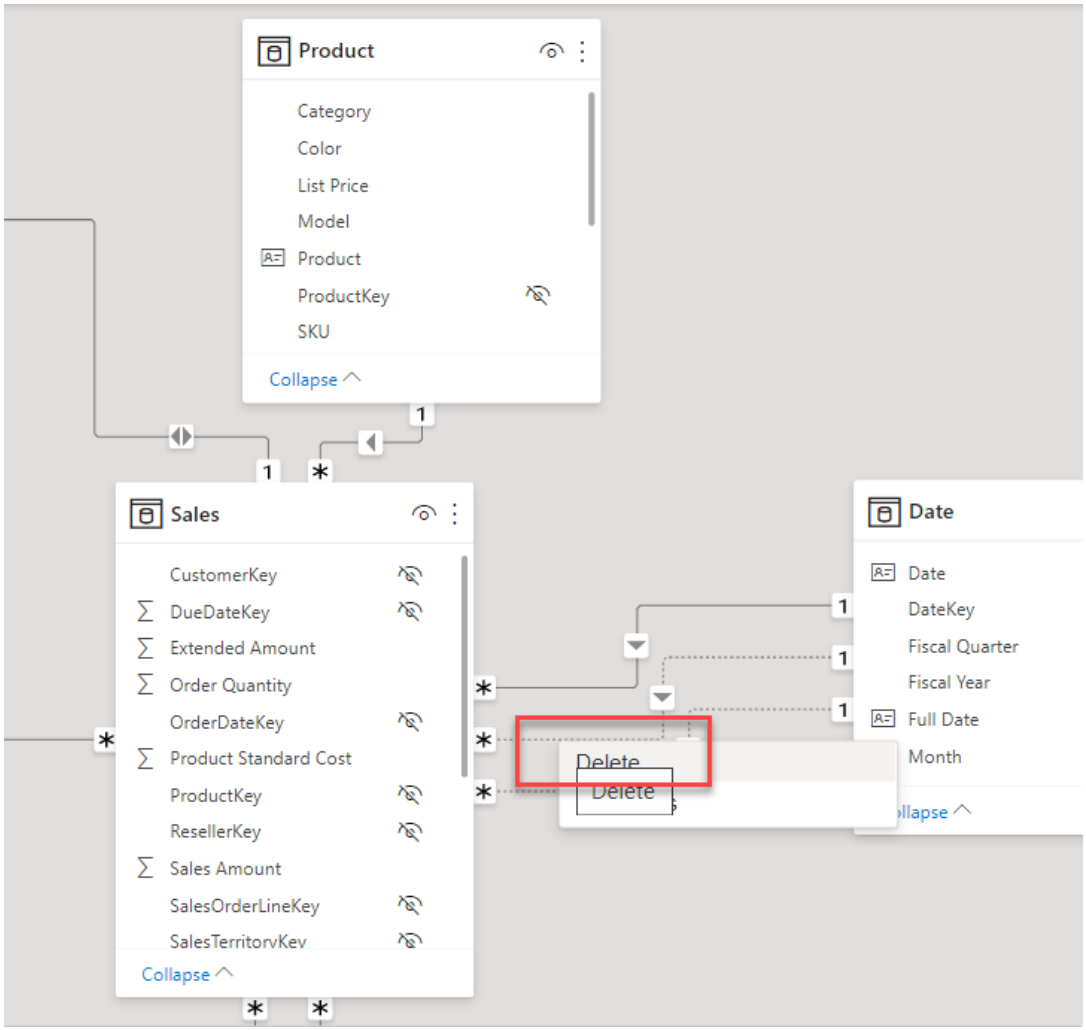
Exercise 1 – Duplicate Table

The next exercise shows how to duplicate table.

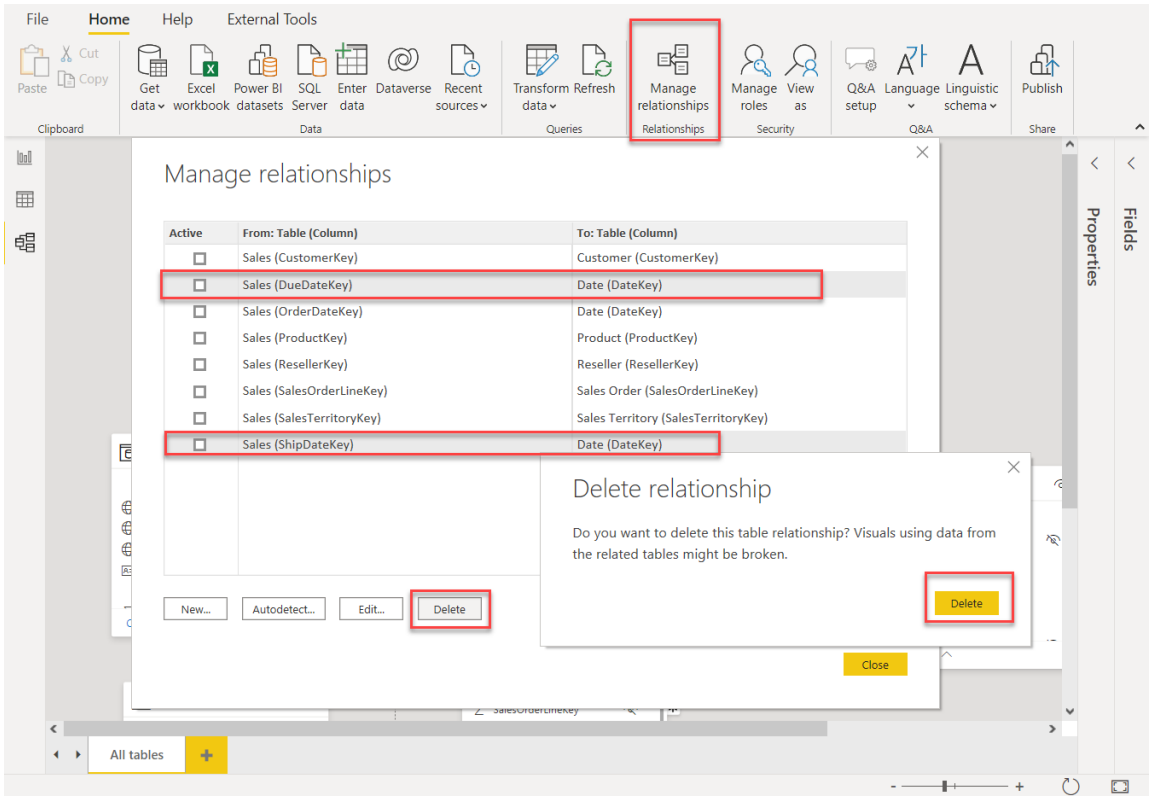
1. Open the **Adventure Works DW 2020 M02.pbix** Power BI Desktop file.
2. Switch to Model diagram view. Mouse over the 3 relationships between sales and Date, notice the doted line between Sales and Date (Inactive relationship).



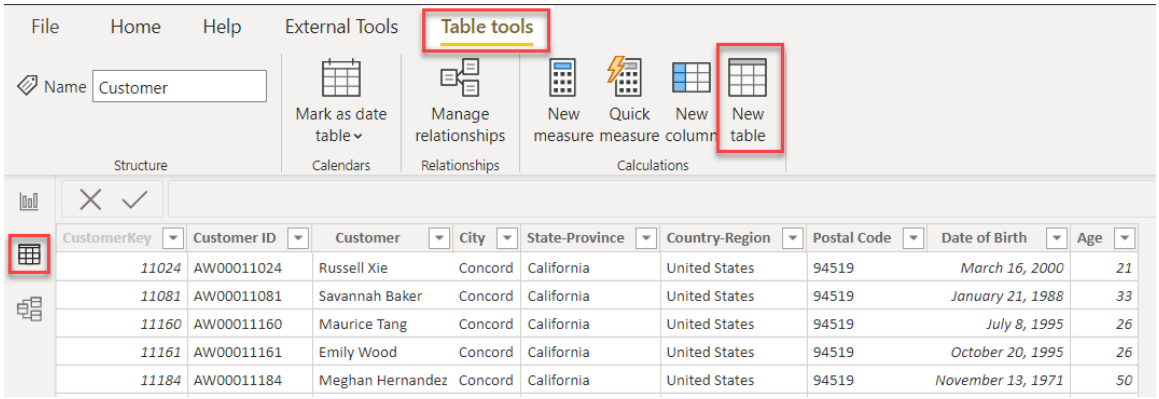
3. Notice that since the active relationship is between [OrderDateKey] and [DateKey], any slicing from the Date would filter Sales table by Order Date.
4. To delete inactive relationship, right click on the first dotted line and select delete.



5. Repeat the above step one more time to delete the second dotted line.
6. Alternatively, you can select Manage relationships from the Home -> Relationships options. Using Ctrl key, select both Sales(DueDateKey) - > Date (DateKey) and Sales (ShipDateKey) - > Date (DateKey) relationship, then select Delete.



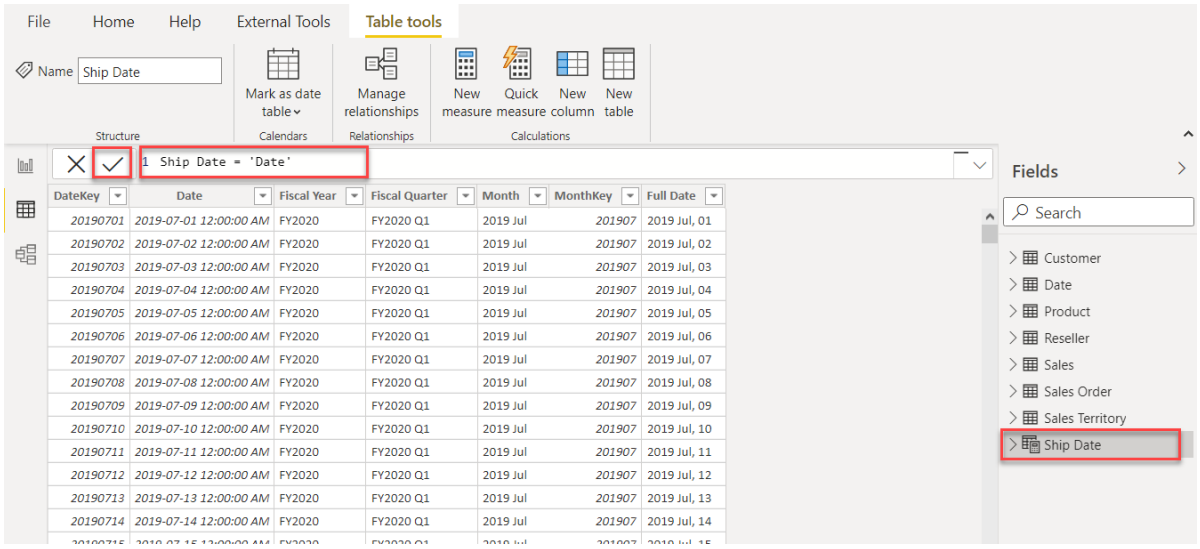
7. Switch the view to Data. From “Table tools” menu select “new table”.



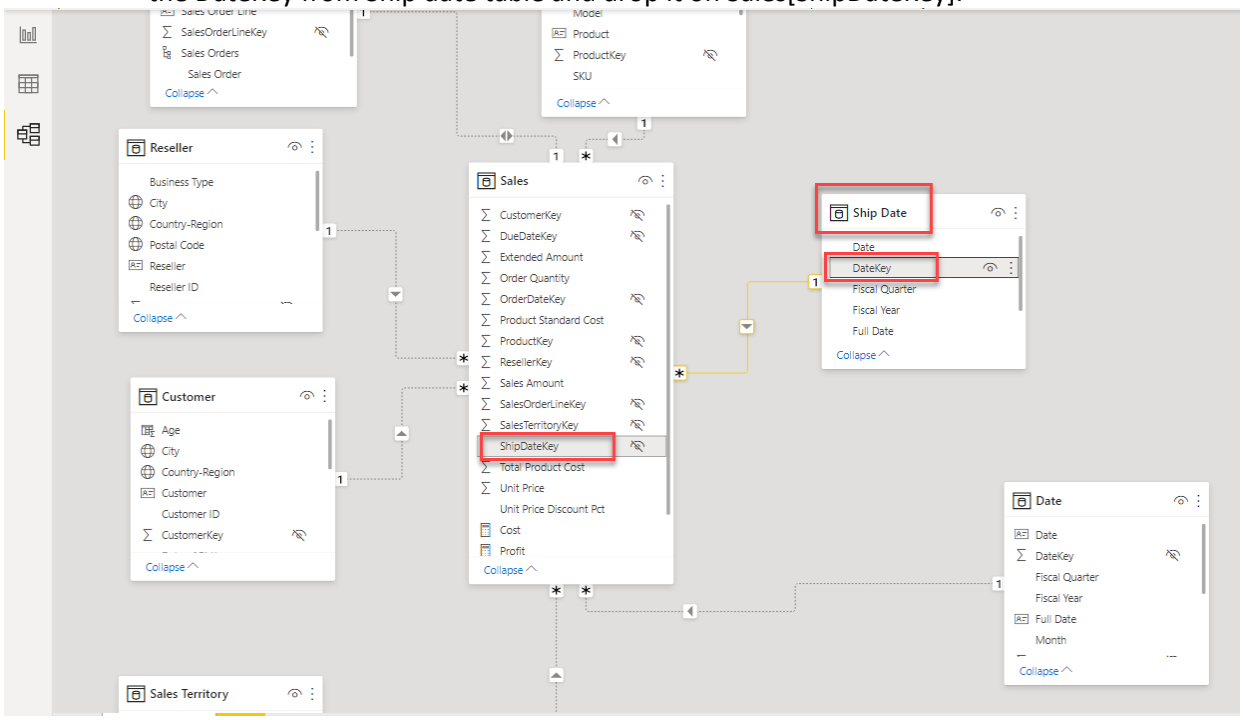
CustomerKey	Customer ID	Customer	City	State-Province	Country-Region	Postal Code	Date of Birth	Age
11024	AW00011024	Russell Xie	Concord	California	United States	94519	March 16, 2000	21
11081	AW00011081	Savannah Baker	Concord	California	United States	94519	January 21, 1988	33
11160	AW00011160	Maurice Tang	Concord	California	United States	94519	July 8, 1995	26
11161	AW00011161	Emily Wood	Concord	California	United States	94519	October 20, 1995	26
11184	AW00011184	Meghan Hernandez	Concord	California	United States	94519	November 13, 1971	50

8. Notice a formula bar appears, change the formula as follows and select Ok or hit enter key from the keyboard.

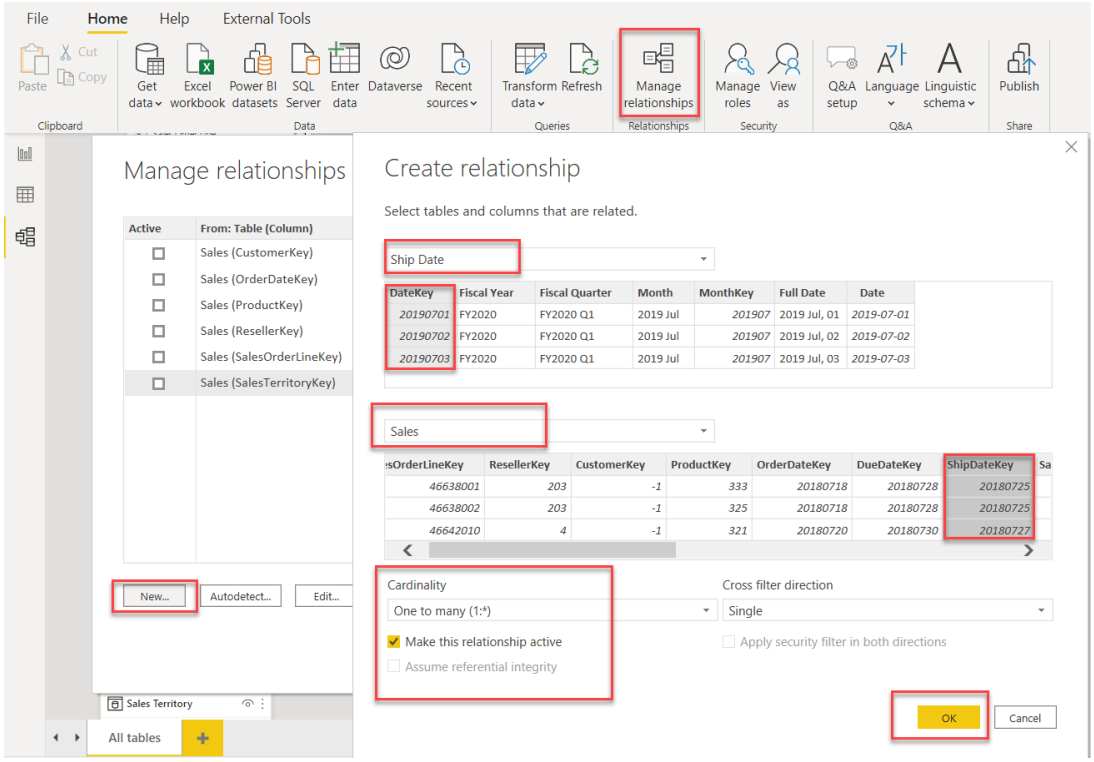
```
Ship Date = 'Date'
```



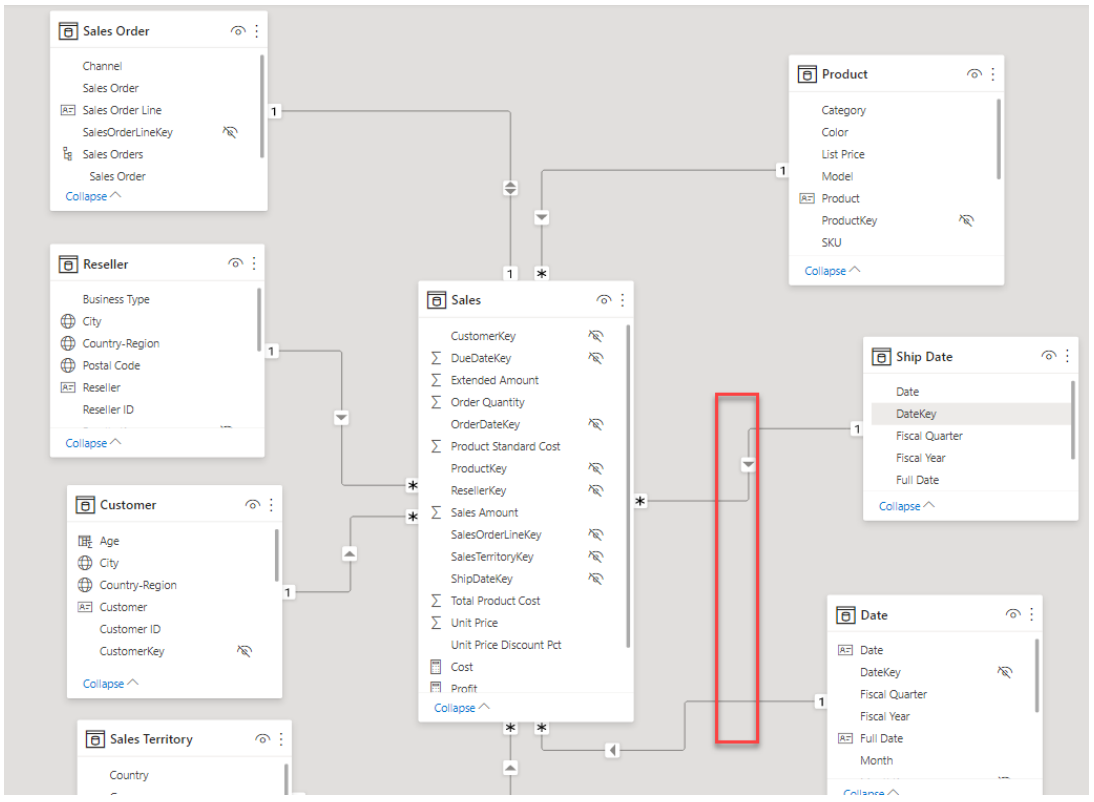
9. Switch to Model view, find the [Ship Date] table and drag it close to Sales table. Hold the DateKey from Ship date table and drop it on Sales[ShipDateKey].



10. Alternatively, you can use Manage relationship dialog to create the relationship. Switch to Model view, click on Manage relationship option from Home -> Relationship option. Then select new and create the relationship.



11. At the end of either of the above two steps, we should see active relationship between the two dates and the sales tables.



12. A calculated table only duplicates data; it doesn't duplicate any model configurations like column visibility or hierarchies. You'll need to configure them for the new table, if required.

13. For example, the **Fiscal Year** column in the Ship Date table can be renamed as **Ship Fiscal Year**.

14. Accordingly, when fields from the Ship Date table are used in visuals, their names are automatically included in captions like the visual title or axis labels.

15. Rename the following columns:

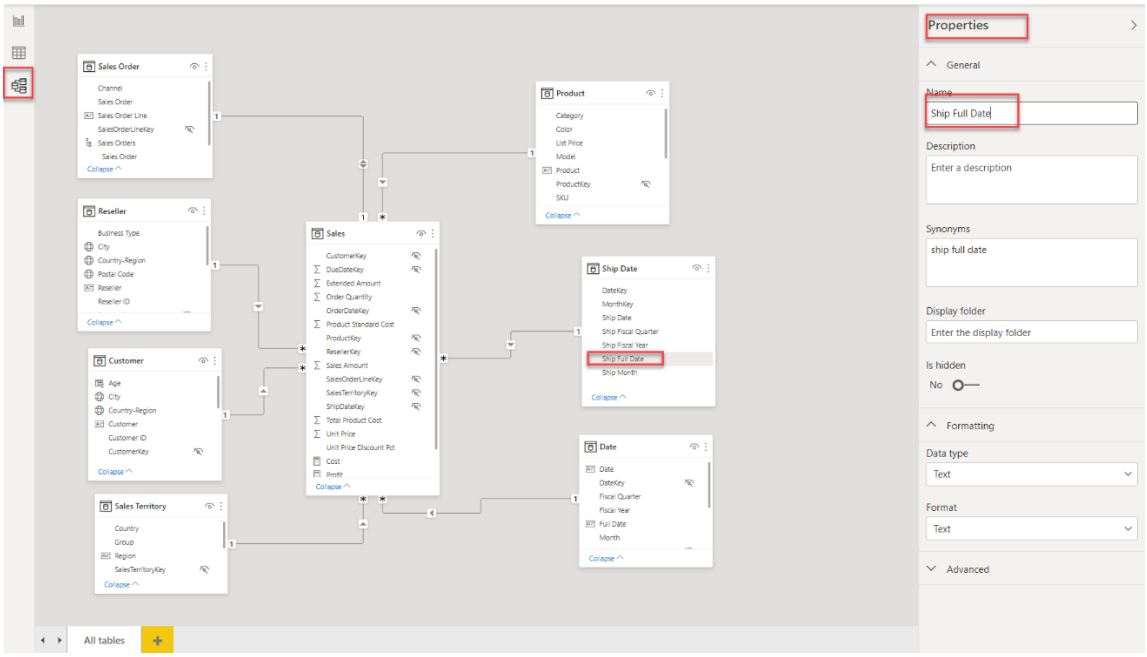
    Date as Ship Date

    Fiscal Year as Ship Fiscal Year

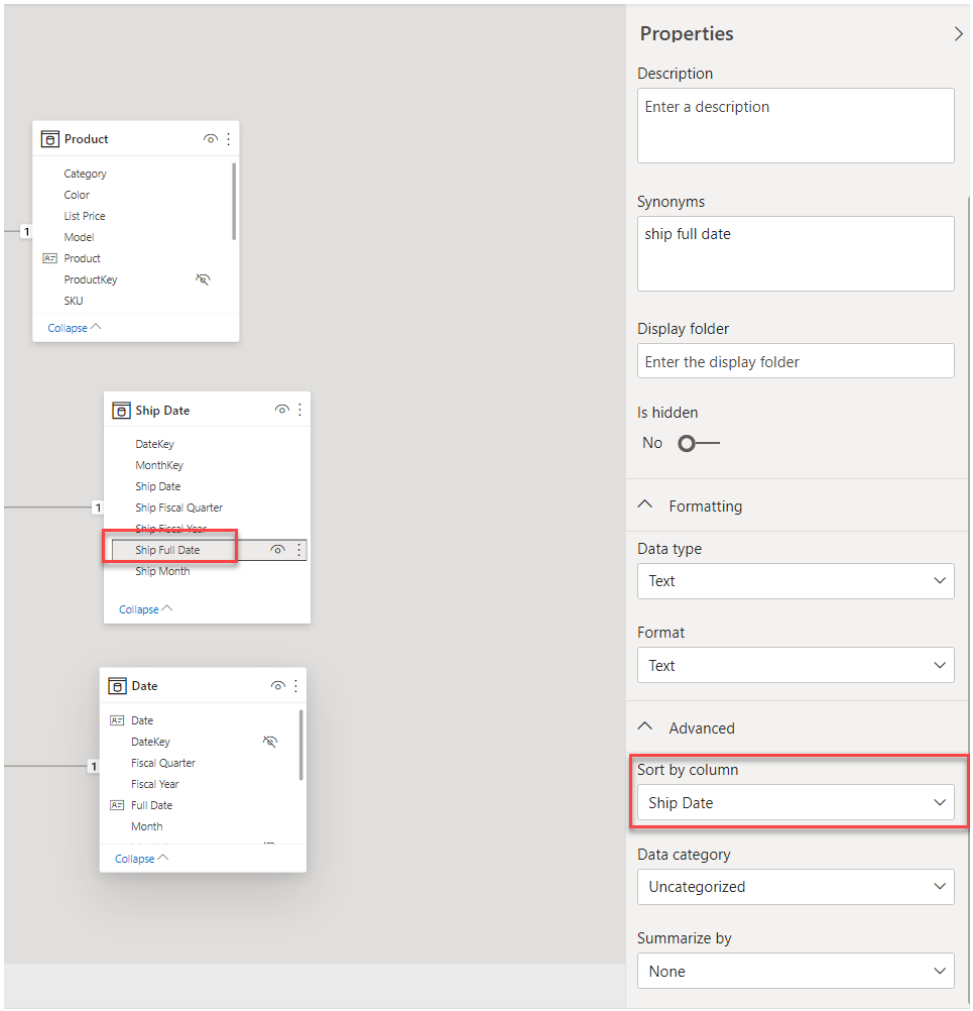
    Fiscal Quarter as Ship Fiscal Quarter

    Month as Ship Month

    Full Date as Ship Full Date



16. Sort the **Ship Full Date** column by the **Ship Date** column.



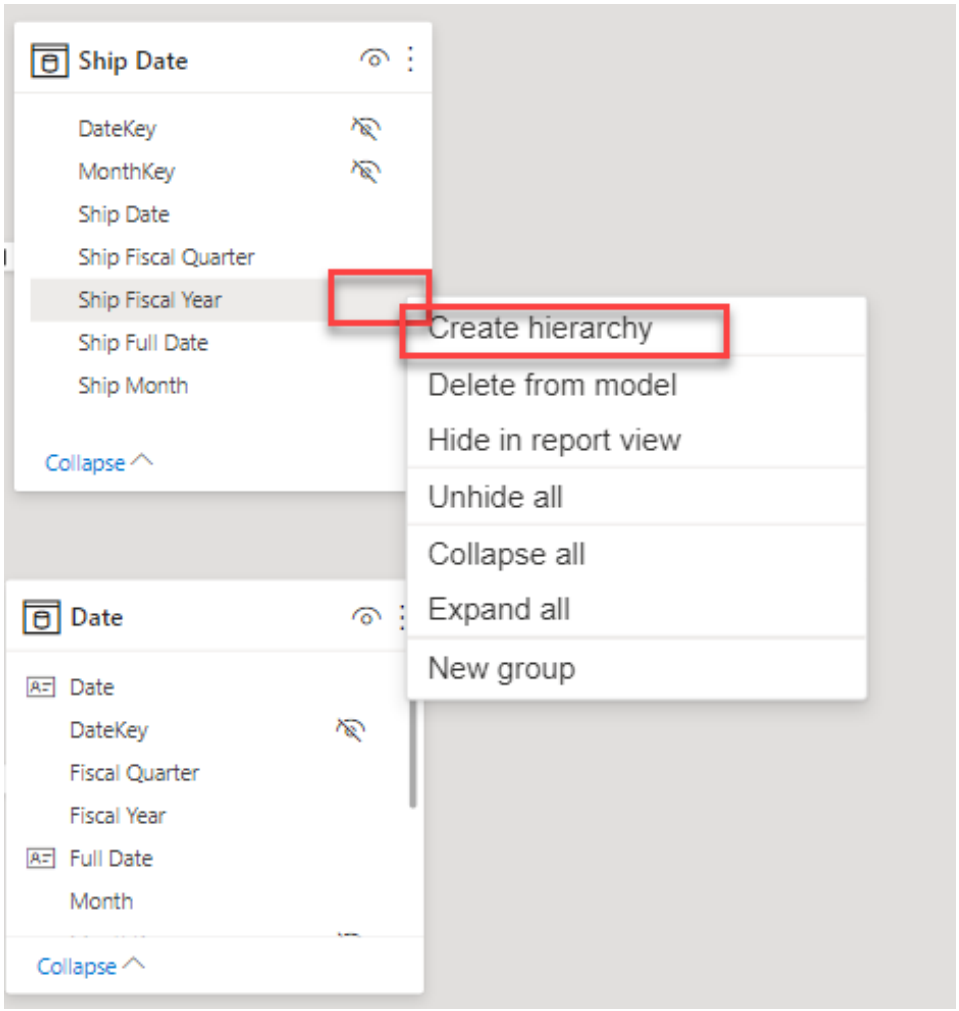
17. Sort the **Ship Month** column by the **MonthKey** column.
18. Hide the **MonthKey** column.
19. Create a hierarchy named **Fiscal** with the following levels:

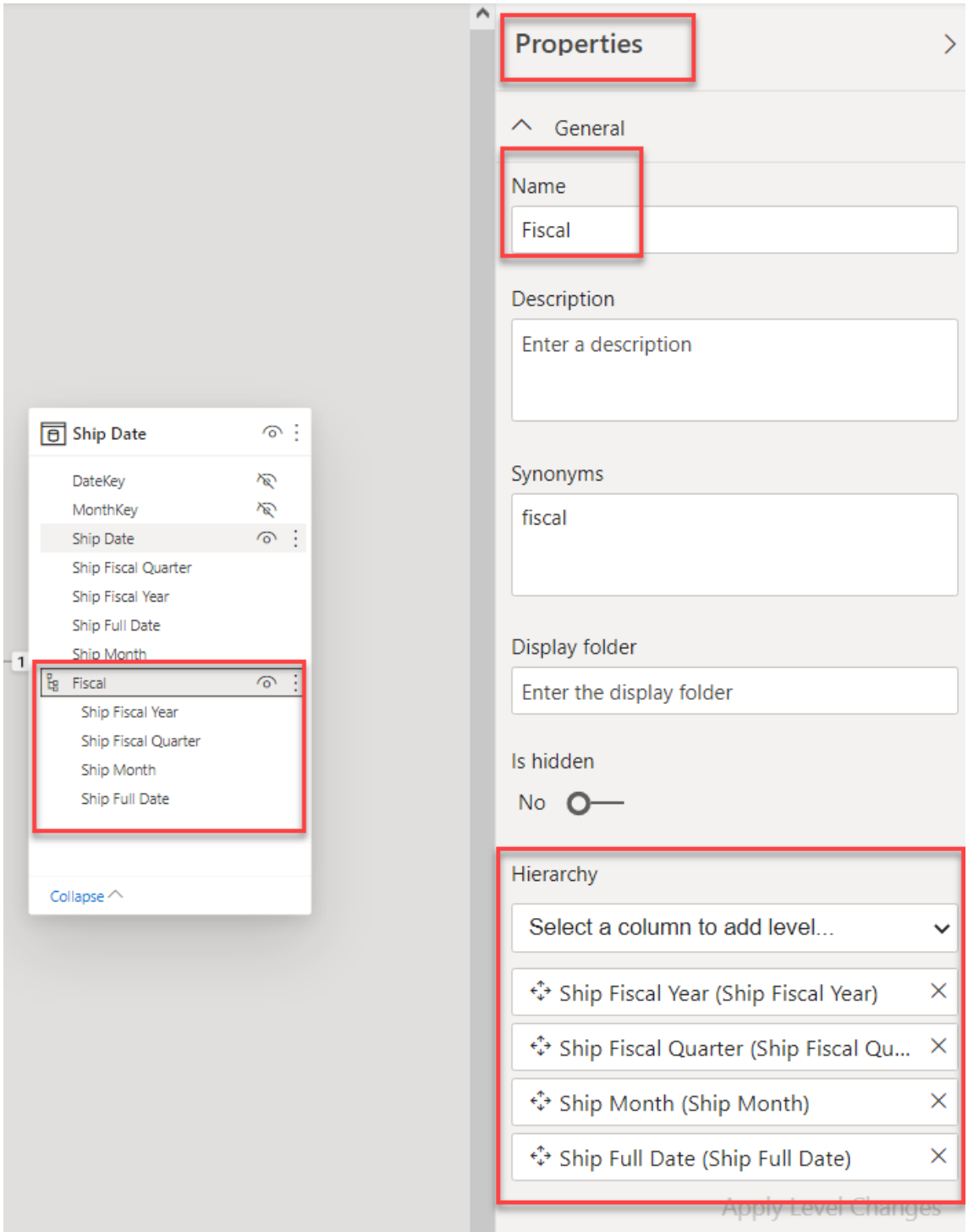
    Ship Fiscal Year

    Ship Fiscal Quarter

    Ship Month

    Ship Full Date


Ship Date
DateKey
MonthKey
Ship Date
Ship Fiscal Quarter
Ship Fiscal Year
Ship Full Date
Ship Month
Collapse
Create hierarchy
Delete from model
Hide in report view
Unhide all
Collapse all
Expand all
New group
Date
Date
DateKey
Fiscal Quarter
Fiscal Year
Full Date
Month
Collapse

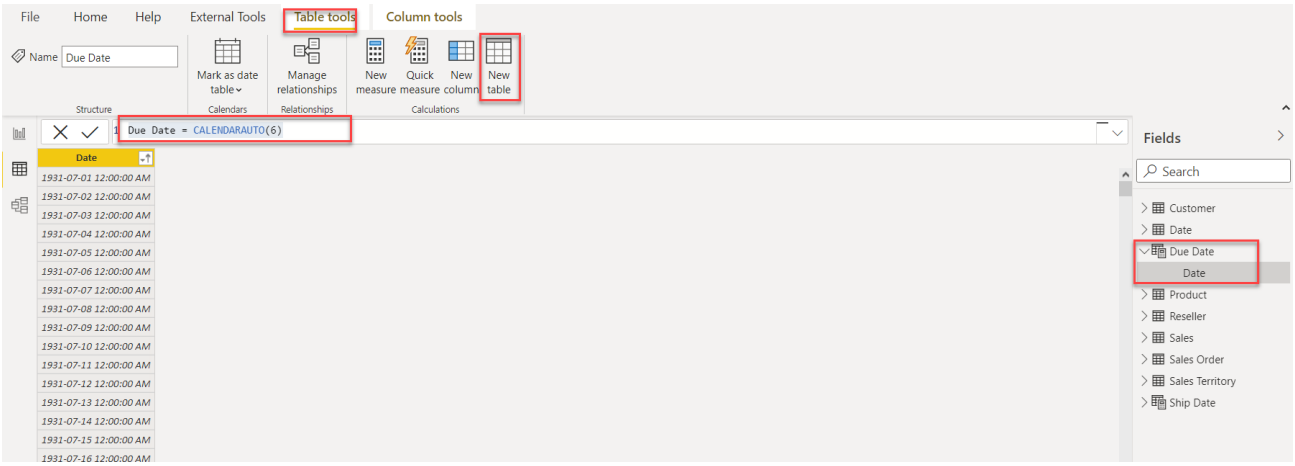

Properties
General
Name
Fiscal
Description
Enter a description
Synonyms
fiscal
Display folder
Enter the display folder
Is hidden
No
Hierarchy
Select a column to add level...
Ship Fiscal Year (Ship Fiscal Year)
Ship Fiscal Quarter (Ship Fiscal Qu...
Ship Month (Ship Month)
Ship Full Date (Ship Full Date)
Apply Level Changes
Ship Date
DateKey
MonthKey
Ship Date
Ship Fiscal Quarter
Ship Fiscal Year
Ship Full Date
Ship Month
Fiscal
Ship Fiscal Year
Ship Fiscal Quarter
Ship Month
Ship Full Date
Collapse

Exercise 2 – Create a Date Table

The next exercise shows how to add a Date Table using Dax.

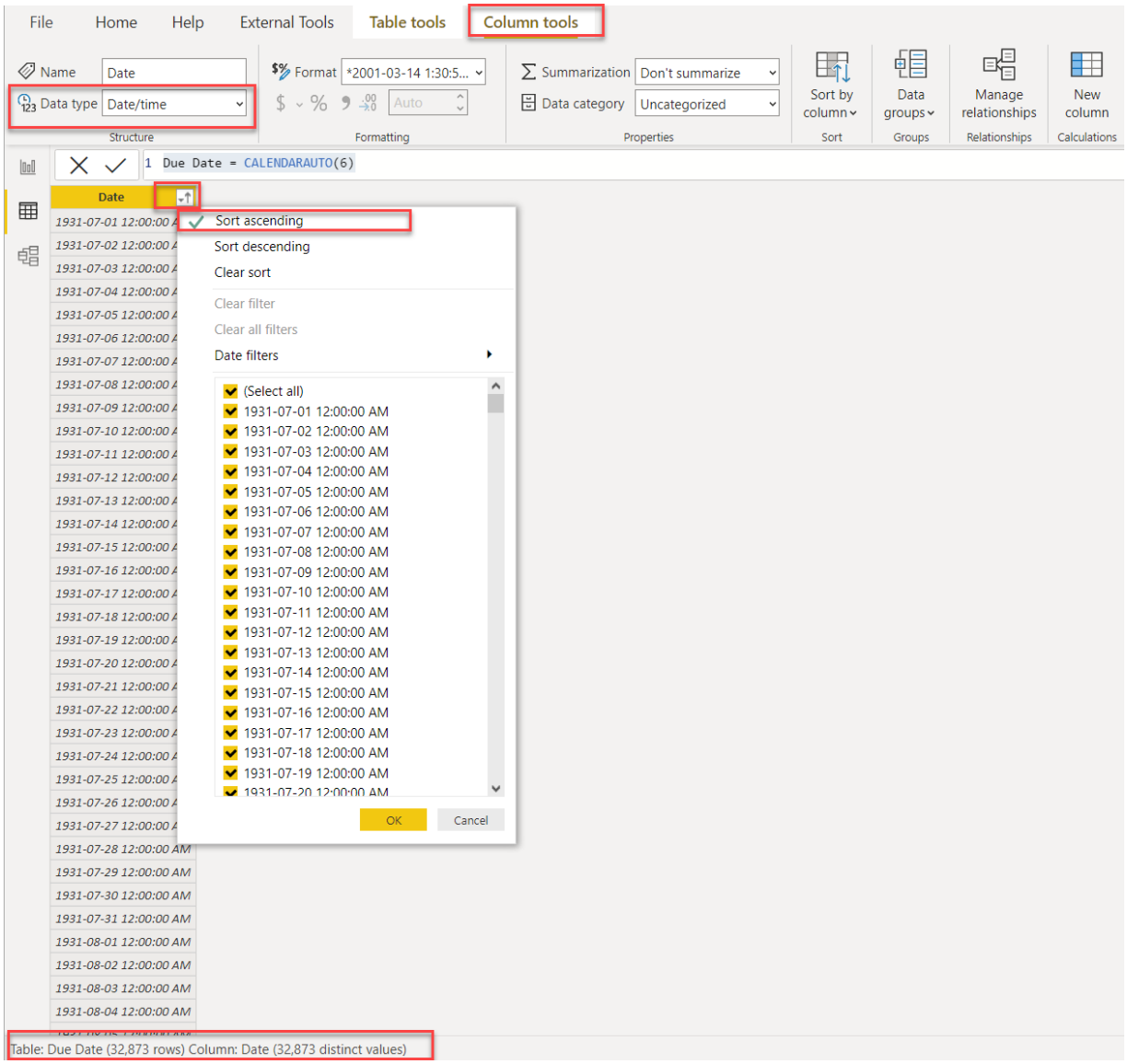
1. Continue with the file from Exercise 1.
2. In the next example, a second calculated table will be created, this time by using the CALENDARAUTO DAX function.
3. Create the Due Date calculated table by using the following definition.

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



4. The CALENDARAUTO function takes a single optional argument, which is the last month number of the year, and returns a single-column table.
5. If you don't pass in a month number, it's assumed to be 12 (for December).
6. For example, at Adventure Works, their financial year ends on June 30 of each year, so the value 6 (for June) is passed in.
7. The function scans all date and date/time columns in your model to determine the earliest and latest stored date values. Due to this, this expression could take few seconds.
8. It then produces a complete set of dates that span all dates in your model, ensuring that full years of dates are loaded.
9. For example, if the earliest date that is stored in your model is October 15, 2017, then the first date that is returned by the CALENDARAUTO function would be July 1, 2017.
10. If the latest date that is stored in the model is June 15, 2020, then the last date that is returned by the CALENDARAUTO function would be June 30, 2020.
11. Effectively, the CALENDARAUTO function guarantees that the following requirements to *mark a date table* are met:
    a) The table must include a column of data type Date.
    b) The column must contain complete years.
    c) The column must not have missing dates.
12. Next, switch to data view, and then in the **Fields** pane, select the Due Date table.

13. Now, review the column of dates (D/M/Y). You might want to order them to see the earliest date in the first row by selecting the arrow inside the **Date** column header and then sorting in ascending order.



14. Ordering or filtering columns doesn't change how the values are stored.
15. These functions in the data view help you explore and understand the data.
16. Now that the **Date** column is selected, review the message in the status bar (located in the lower-left corner). It describes how many rows that the table stores and how many distinct values are found in the selected column.
17. When the table rows and distinct values are the same, it means that the column contains unique values.
18. That factor is important for two reasons: It satisfies the requirements to mark a date table, and it allows this column to be used in a model relationship as the one-side.
19. The Due Date calculated table will recalculate each time a table that contains a date column refreshes.

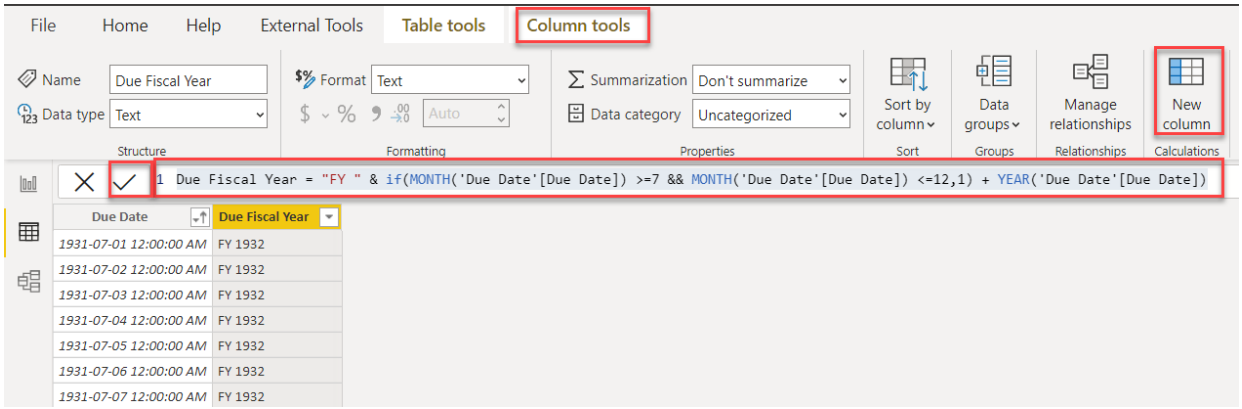
20. In other words, when a row is loaded into the Sales table with an order date of July 1, 2020, the Due Date table will automatically extend to include dates through to the end of the next year: June 30, 2021.
21. The Due Date table requires additional columns to support the known filtering and grouping requirements, specifically by year, quarter, and month.

Exercise 3 – Add calculated columns

The next exercise shows how to add calculated columns.

1. Continue with the file from Exercise 2.
2. In data view, in the **Fields** pane, ensure that the Due Date table is selected.
3. Before you create a calculated column, first rename the **Date** column to **Due Date**.
4. Now, you can add a calculated column to the Due Date table.
5. To create a calculated column, in the **Table tools** contextual ribbon, from inside the **Calculations** group, select "**New column**".
6. In the formula box, enter the following calculated column definition and then press the **Enter** key.

```
Due Fiscal Year = "FY " & if(MONTH('Due Date'[Due Date]) >=7 && MONTH('Due Date'[Due Date]) <=12,1) + YEAR('Due Date'[Due Date])
```



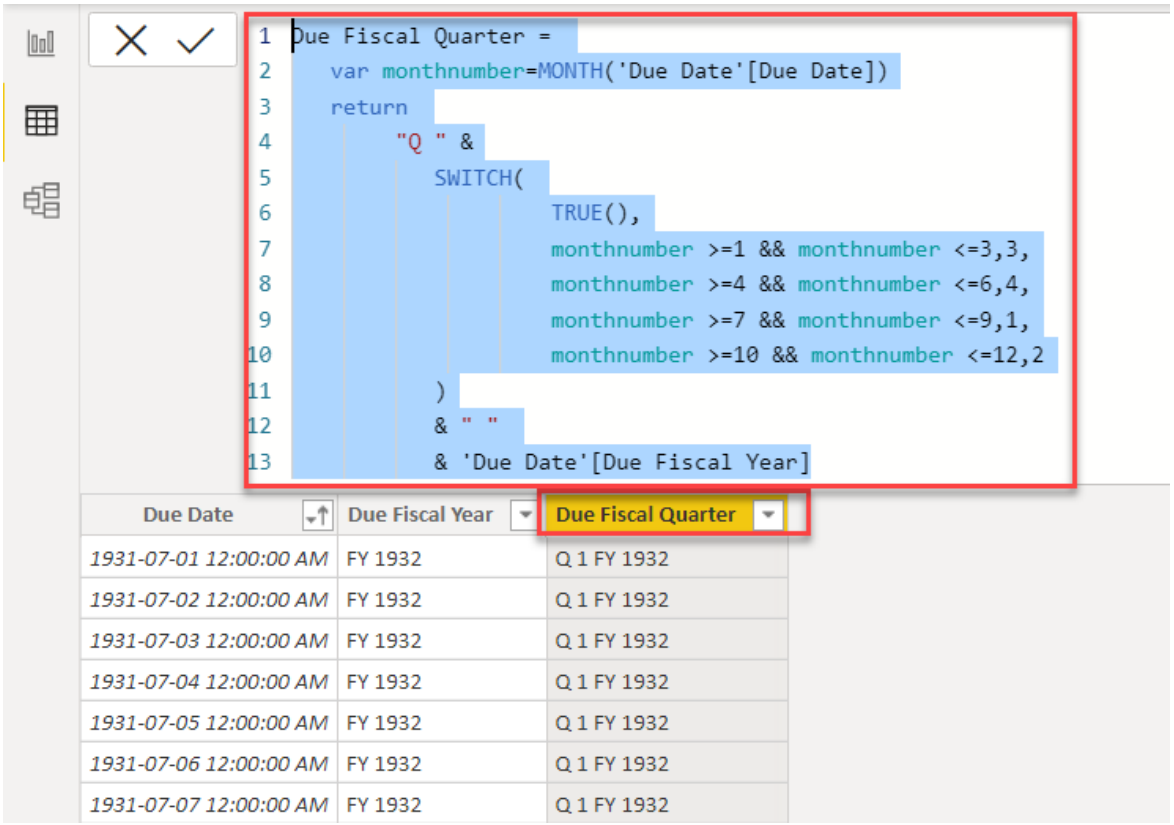
7. The following steps describe how Microsoft Power BI evaluates the calculate column formula:
    a) The addition operator (+) is evaluated before the text concatenation operator (&).
    b) The YEAR DAX function returns the whole number value of the due date year.
    c) The IF DAX function returns the value when the due date month number is 7-12 (July to December); otherwise, it returns BLANK. (For example, because the Adventure Works financial year is July-June, the last six months of the calendar year will use the next calendar year as their financial year.)
    d) The year value is added to the value that is returned by the IF function, which is the value one or BLANK. If the value is BLANK, it's implicitly converted to zero (0) to allow the addition to produce the fiscal year value.

e) The literal text value "FY" concatenated with the fiscal year value, which is implicitly converted to text.

8. Add a second calculated column by using the following definition:

```
Due Fiscal Quarter =
   var monthnumber=MONTH('Due Date'[Due Date])
   return
        "Q " &
           SWITCH(
                    TRUE(),
                    monthnumber >=1 && monthnumber <=3,3,
                    monthnumber >=4 && monthnumber <=6,4,
                    monthnumber >=7 && monthnumber <=9,1,
                    monthnumber >=10 && monthnumber <=12,2
           )
           & " "
           & 'Due Date'[Due Fiscal Year]
```

9. The calculated column definition adds the **Due Fiscal Quarter** column to the Due Date table.
10. The Switch function returns the quarter number (Quarter 1 is July-September), and the result is concatenated to the **Due Fiscal Year** column value and the literal text **Q**.



Due Date	Due Fiscal Year	Due Fiscal Quarter
1931-07-01 12:00:00 AM	FY 1932	Q 1 FY 1932
1931-07-02 12:00:00 AM	FY 1932	Q 1 FY 1932
1931-07-03 12:00:00 AM	FY 1932	Q 1 FY 1932
1931-07-04 12:00:00 AM	FY 1932	Q 1 FY 1932
1931-07-05 12:00:00 AM	FY 1932	Q 1 FY 1932
1931-07-06 12:00:00 AM	FY 1932	Q 1 FY 1932
1931-07-07 12:00:00 AM	FY 1932	Q 1 FY 1932

11. Add a third calculated column by using the following definition

```
Due Month = FORMAT('Due Date'[Due Date],"mmmm, yyyy")
```

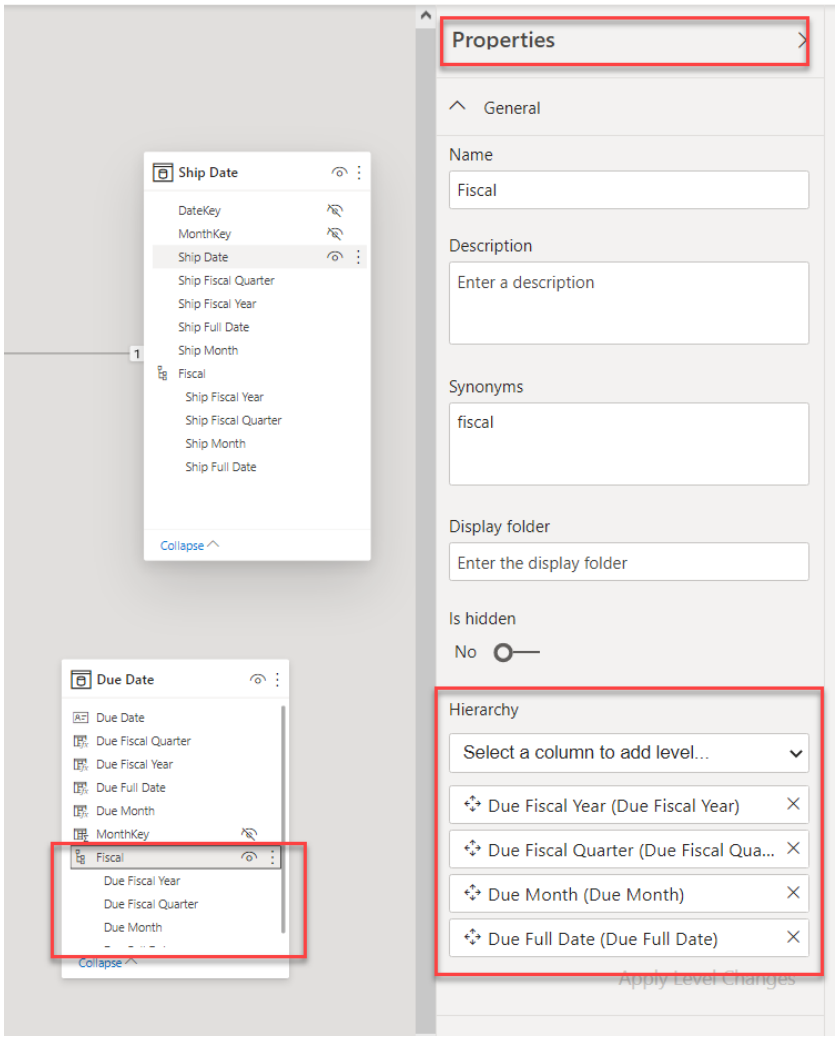
12. The calculated column definition adds the **Due Month** column to the Due Date table.
13. The FORMAT DAX function converts the **Due Date** column value to text by using a format string. In this case, the format string produces a label that describes the year and abbreviated month name.
14. Add a fourth calculated column by using the following definition:

```
Due Full Date = FORMAT('Due Date'[Due Date],"yyyy mmmm, dd")
```

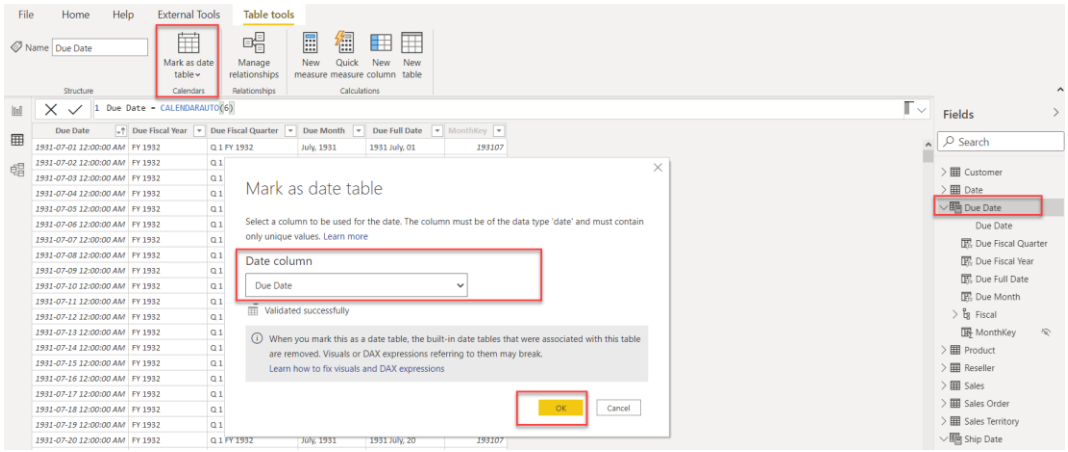
15. Add a fifth calculated column by using the following definition:

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

16. It produces a numeric value that can be used to sort the **Due Month** text values in chronological order.
17. Verify that the Due Date table has six columns. The first column was added when the calculated table was created, and the other five columns were added as calculated columns.
18. To complete the design of the Due Date table, you can:
    a) Sort the **Due Full Date** column by the **Due Date** column.
    b) Sort the **Due Month** column by the **MonthKey** column.
    c) Hide the **MonthKey** column.
    d) Create a hierarchy named **Fiscal** with the following levels:
        1. Due Fiscal Year
        2. Due Fiscal Quarter
        3. Due Month
        4. Due Full Date



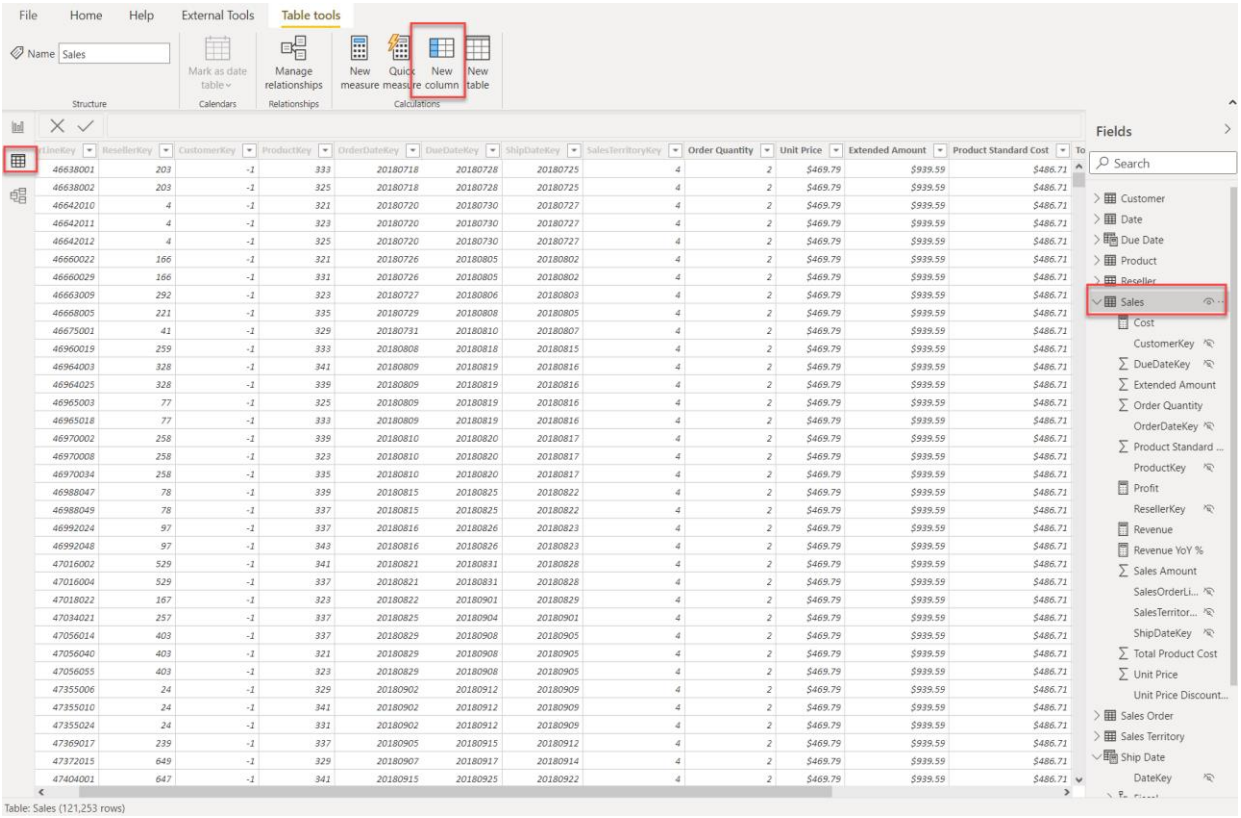
19. Mark the Due Date table as a date table by using the **Due Date** column.



Exercise 4 – Add calculated columns in sales table

The next exercise shows how to add calculated columns.

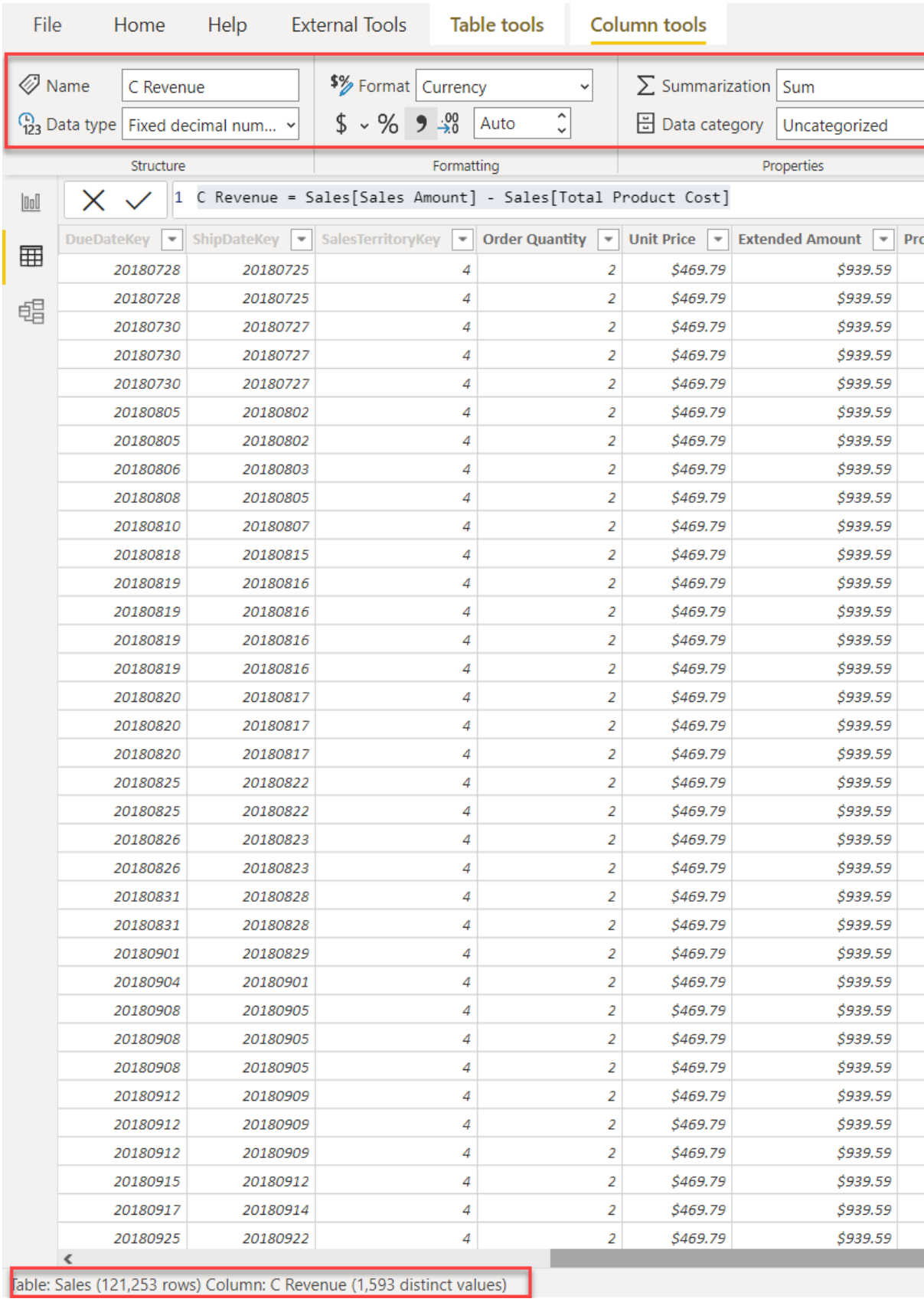
1. Continue with the file from Exercise 3.
2. Switch to data view and ensure that Sales table is selected. From “Table tools” select “New Column”



3. Use the following DAX Expression in the formula bar and select Ok.

```
C Revenue = Sales[Sales Amount] - Sales[Total Product Cost]
```

4. Notice the result in the data view and formatting option.



5. Save the Power BI File as Adventure Works DW 2020 M03.pbix

Terms of Use

By using this hands-on lab, you agree to the following terms:

The technology/functionality described in this hands-on lab is provided by Microsoft Corporation in a "sandbox" testing environment for purposes of obtaining your feedback and to provide you with a learning experience. You may only use the hands-on lab to evaluate such technology features and functionality and provide feedback to Microsoft. You may not use it for any other purpose. Without written permission, you may not modify, copy, distribute, transmit, display, perform, reproduce, publish, license, create derivative works from, transfer, or sell this hands-on lab or any portion thereof.

COPYING OR REPRODUCTION OF THE HANDS-ON LAB (OR ANY PORTION OF IT) TO ANY OTHER SERVER OR LOCATION FOR FURTHER REPRODUCTION OR REDISTRIBUTION WITHOUT WRITTEN PERMISSION IS EXPRESSLY PROHIBITED.

THIS HANDS-ON LAB PROVIDES CERTAIN SOFTWARE TECHNOLOGY/PRODUCT FEATURES AND FUNCTIONALITY, INCLUDING POTENTIAL NEW FEATURES AND CONCEPTS, IN A SIMULATED ENVIRONMENT WITHOUT COMPLEX SET-UP OR INSTALLATION FOR THE PURPOSE DESCRIBED ABOVE. THE TECHNOLOGY/CONCEPTS REPRESENTED IN THIS HANDS-ON LAB MAY NOT REPRESENT FULL FEATURE FUNCTIONALITY AND MAY NOT WORK THE WAY A FINAL VERSION MAY WORK. WE ALSO MAY NOT RELEASE A FINAL VERSION OF SUCH FEATURES OR CONCEPTS. YOUR EXPERIENCE WITH USING SUCH FEATURES AND FUNCITONALITY IN A PHYSICAL ENVIRONMENT MAY ALSO BE DIFFERENT.

FEEDBACK If you give feedback about the technology features, functionality and/or concepts described in this hands-on lab to Microsoft, you give to Microsoft, without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software or documentation to third parties because we include your feedback in them. These rights survive this agreement.

MICROSOFT CORPORATION HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS WITH REGARD TO THE HANDS-ON LAB, INCLUDING ALL WARRANTIES AND CONDITIONS OF MERCHANTABILITY, WHETHER EXPRESS, IMPLIED OR STATUTORY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. MICROSOFT DOES NOT MAKE ANY ASSURANCES OR REPRESENTATIONS WITH REGARD TO THE ACCURACY OF THE RESULTS, OUTPUT THAT DERIVES FROM USE OF THE VIRTUAL LAB, OR SUITABILITY OF THE INFORMATION CONTAINED IN THE VIRTUAL LAB FOR ANY PURPOSE.

DISCLAIMER This lab contains only a portion of new features and enhancements in Microsoft Power BI. Some of the features might change in future releases of the product.