Microsoft Power Platform

Power BI- Adv DAX

Lab 03

Creating Calculated table and Columns

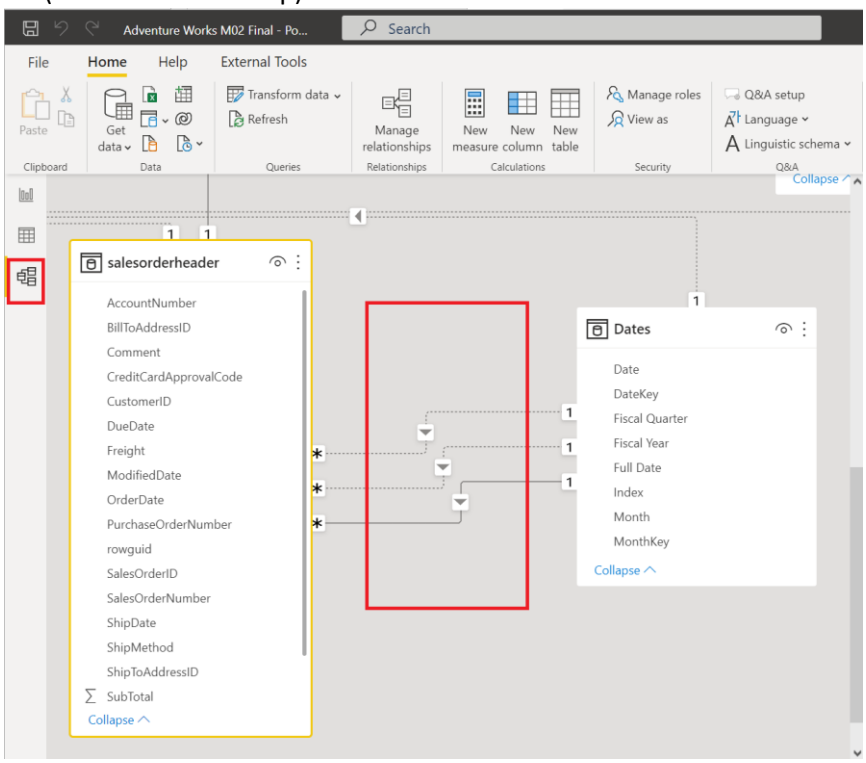
Overview

The estimated time to complete this lab is: 15 Minutes

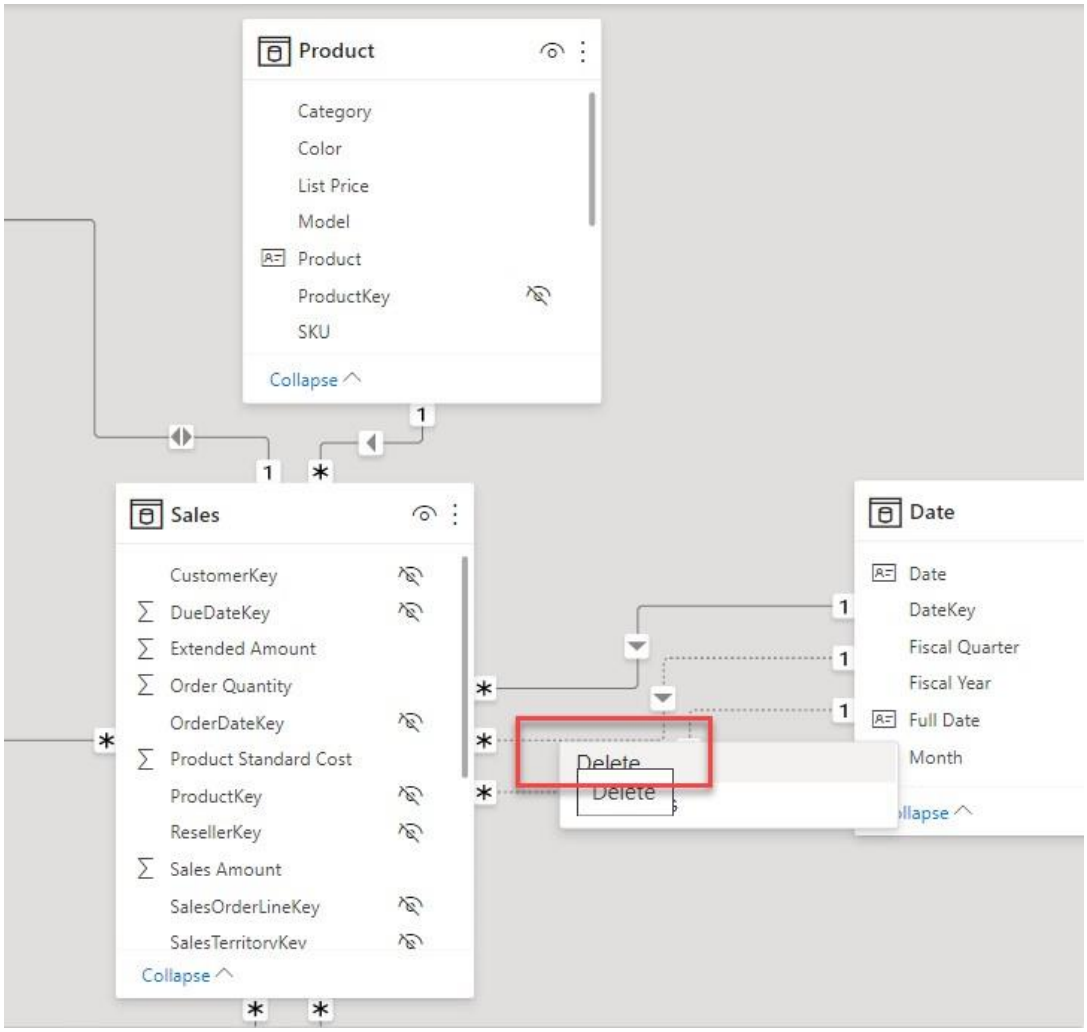
Exercise 1 – Create a Role Playing Dimension Table

The next exercise shows how to duplicate a table.

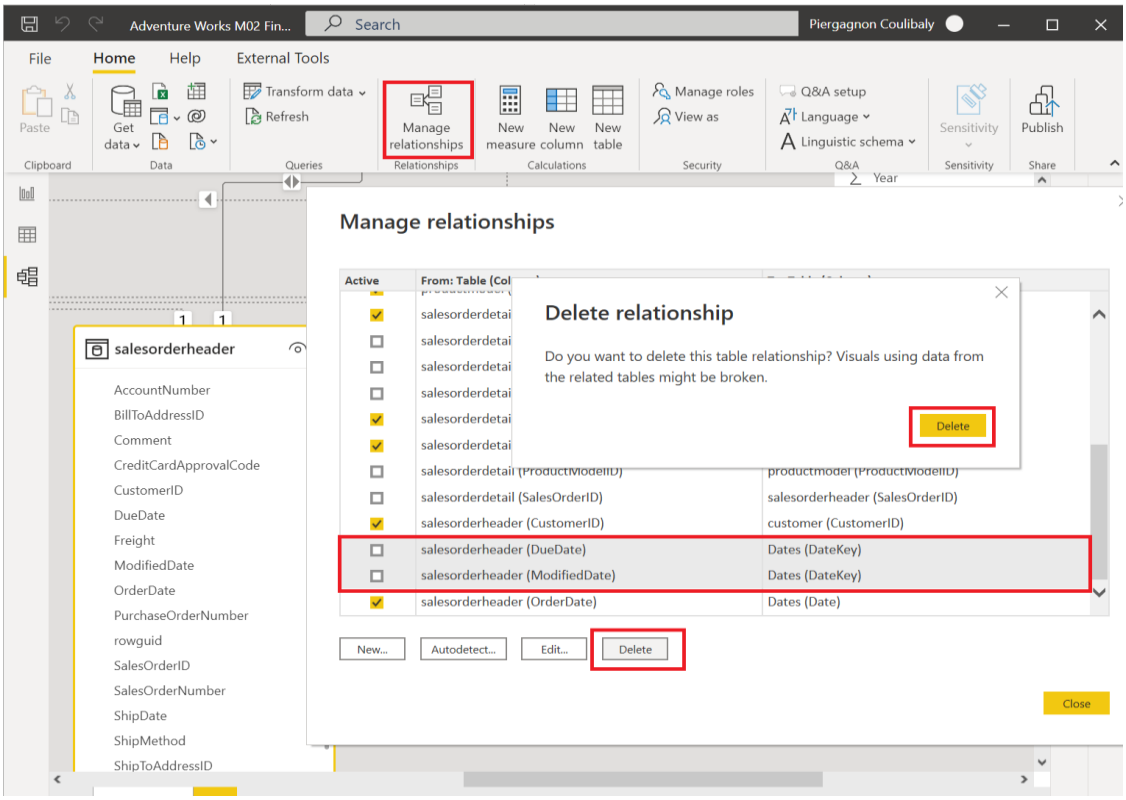
1. Open the **Adventure Works M02.pbix** Power BI Desktop file.
2. Switch to the Model diagram view. Mouse over the 3 relationships between **Sales Order Header** and **Dates**, notice the doted line between **Sales Order Header** and **Dates** (Inactive relationship).



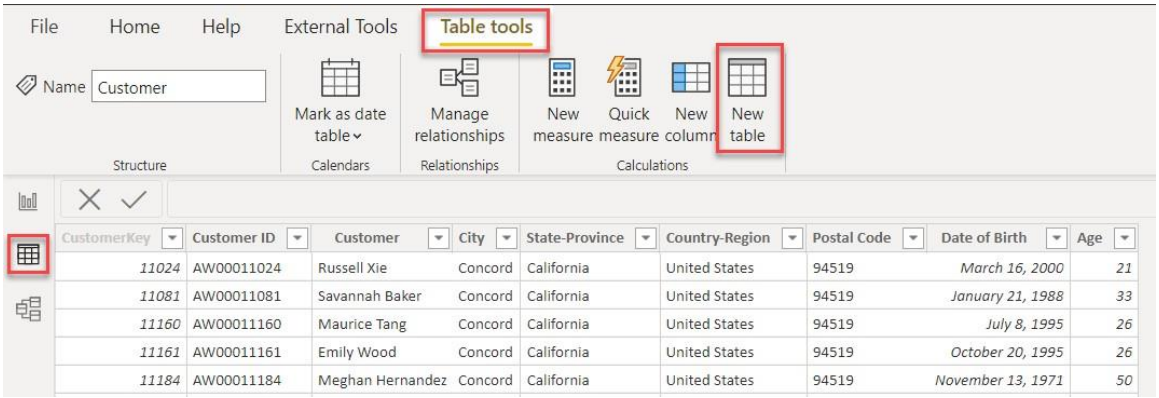
3. Notice that since the active relationship is between Sales Order Header[OrderDate] and Dates[Date], any slicing from the Date would filter Sales Order Header table by Order Date (as opposed to, say, Due Date).
4. To delete an un-needed inactive relationship, right click on the dotted line and select delete.



5. Repeat the above step one more time to delete the second dotted line.
    i Alternatively, you can select Manage relationships from the Home -> Relationships options. Using Ctrl key, select both Sales Order Header(DueDate) - > Dates (Date) and Sales Order Header (ModifiedDate) - > Dates (Date) relationship, then select Delete.

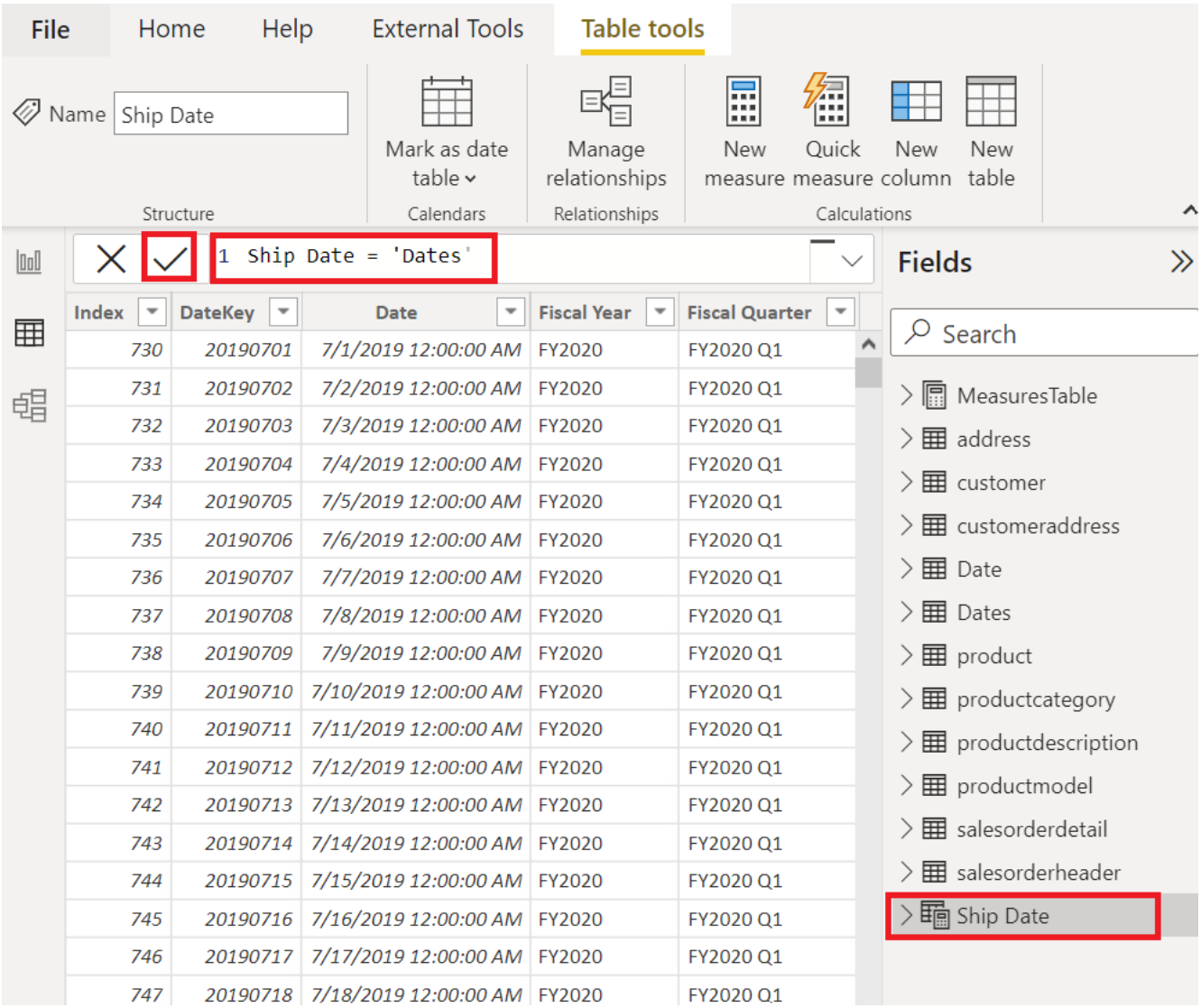


6. Switch the view to Data. From “Table tools” menu select “new table”.



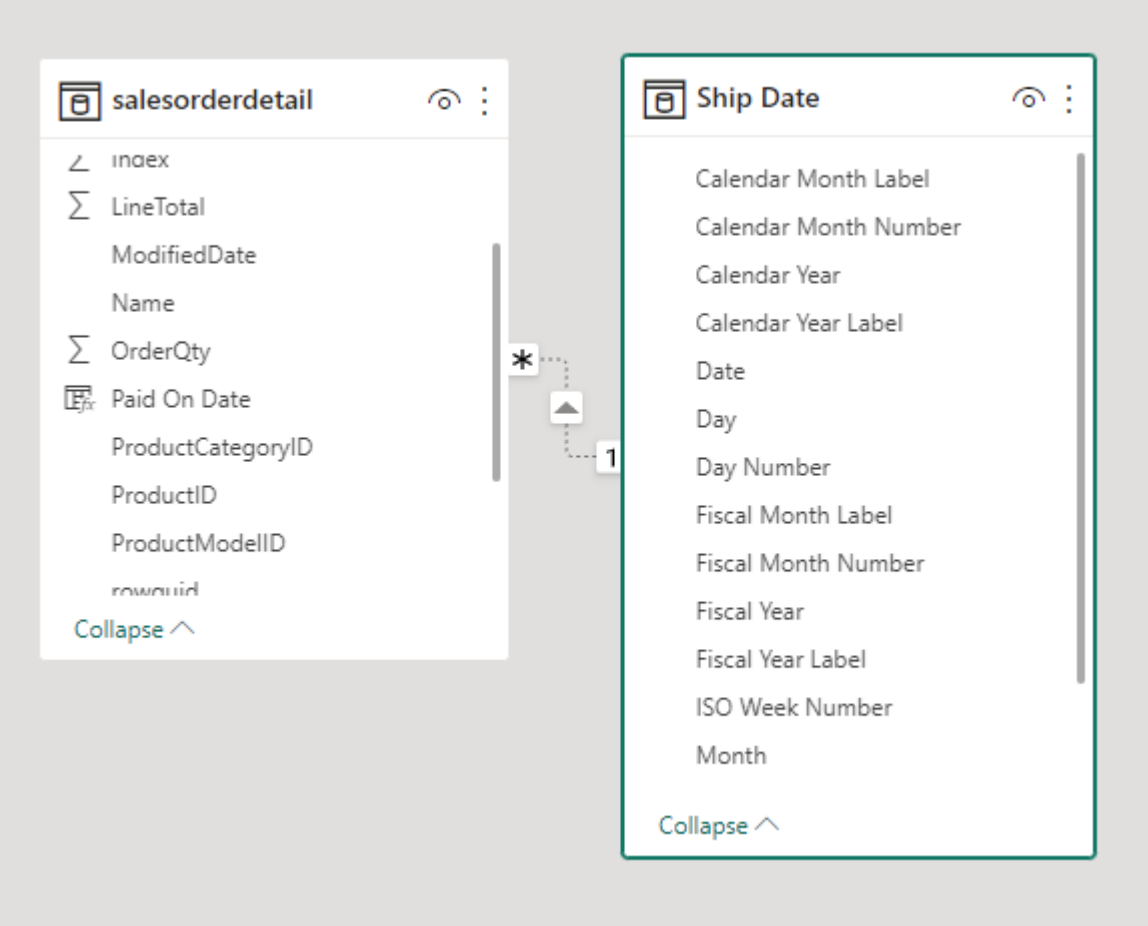
CustomerKey	Customer ID	Customer	City	State-Province	Country-Region	Postal Code	Date of Birth	Age
11024	AW00011024	Russell Xie	Concord	California	United States	94519	March 16, 2000	21
11081	AW00011081	Savannah Baker	Concord	California	United States	94519	January 21, 1988	33
11160	AW00011160	Maurice Tang	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

7. Notice a formula bar appears, change the formula as follows and select Ok or hit enter key from the keyboard. `Ship Date = 'Dates'`

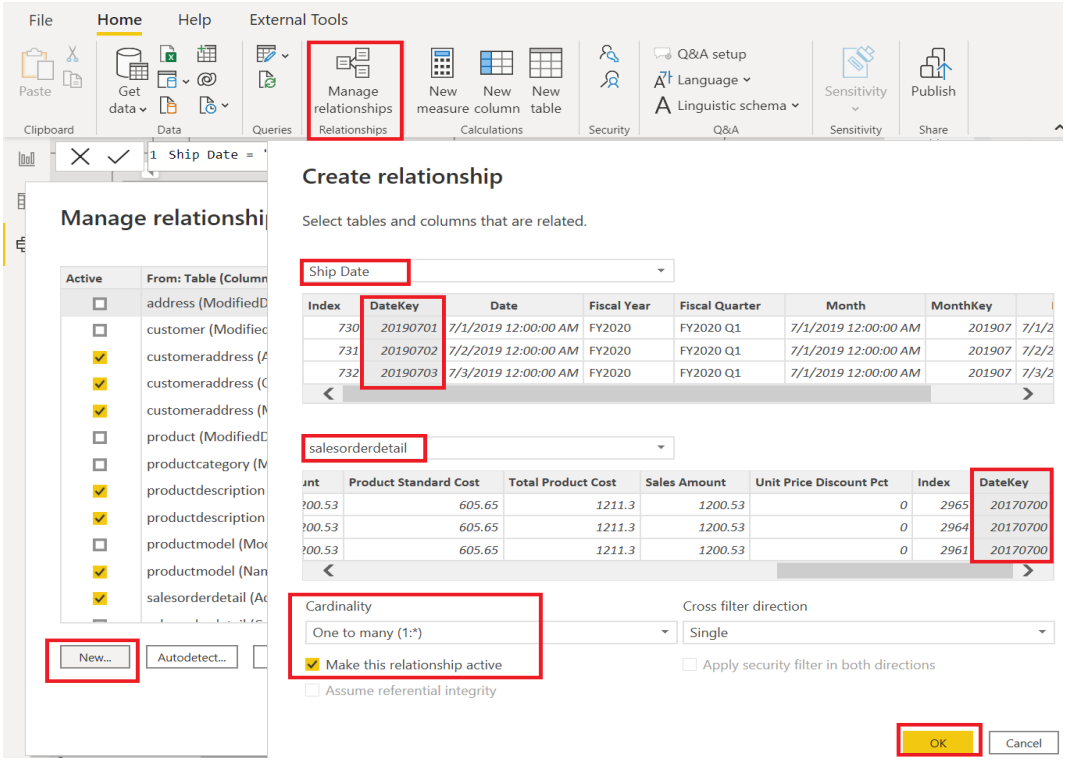


Index	DateKey	Date	Fiscal Year	Fiscal Quarter
730	20190701	7/1/2019 12:00:00 AM	FY2020	FY2020 Q1
731	20190702	7/2/2019 12:00:00 AM	FY2020	FY2020 Q1
732	20190703	7/3/2019 12:00:00 AM	FY2020	FY2020 Q1
733	20190704	7/4/2019 12:00:00 AM	FY2020	FY2020 Q1
734	20190705	7/5/2019 12:00:00 AM	FY2020	FY2020 Q1
735	20190706	7/6/2019 12:00:00 AM	FY2020	FY2020 Q1
736	20190707	7/7/2019 12:00:00 AM	FY2020	FY2020 Q1
737	20190708	7/8/2019 12:00:00 AM	FY2020	FY2020 Q1
738	20190709	7/9/2019 12:00:00 AM	FY2020	FY2020 Q1
739	20190710	7/10/2019 12:00:00 AM	FY2020	FY2020 Q1
740	20190711	7/11/2019 12:00:00 AM	FY2020	FY2020 Q1
741	20190712	7/12/2019 12:00:00 AM	FY2020	FY2020 Q1
742	20190713	7/13/2019 12:00:00 AM	FY2020	FY2020 Q1
743	20190714	7/14/2019 12:00:00 AM	FY2020	FY2020 Q1
744	20190715	7/15/2019 12:00:00 AM	FY2020	FY2020 Q1
745	20190716	7/16/2019 12:00:00 AM	FY2020	FY2020 Q1
746	20190717	7/17/2019 12:00:00 AM	FY2020	FY2020 Q1
747	20190718	7/18/2019 12:00:00 AM	FY2020	FY2020 Q1

8. Switch to Model view, find the [Ship Date] table and drag it close to **Sales Order Detail** table. Hold the Date column from Ship date table and drop it on **Sales Order Detail** [Due Date].



i 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.



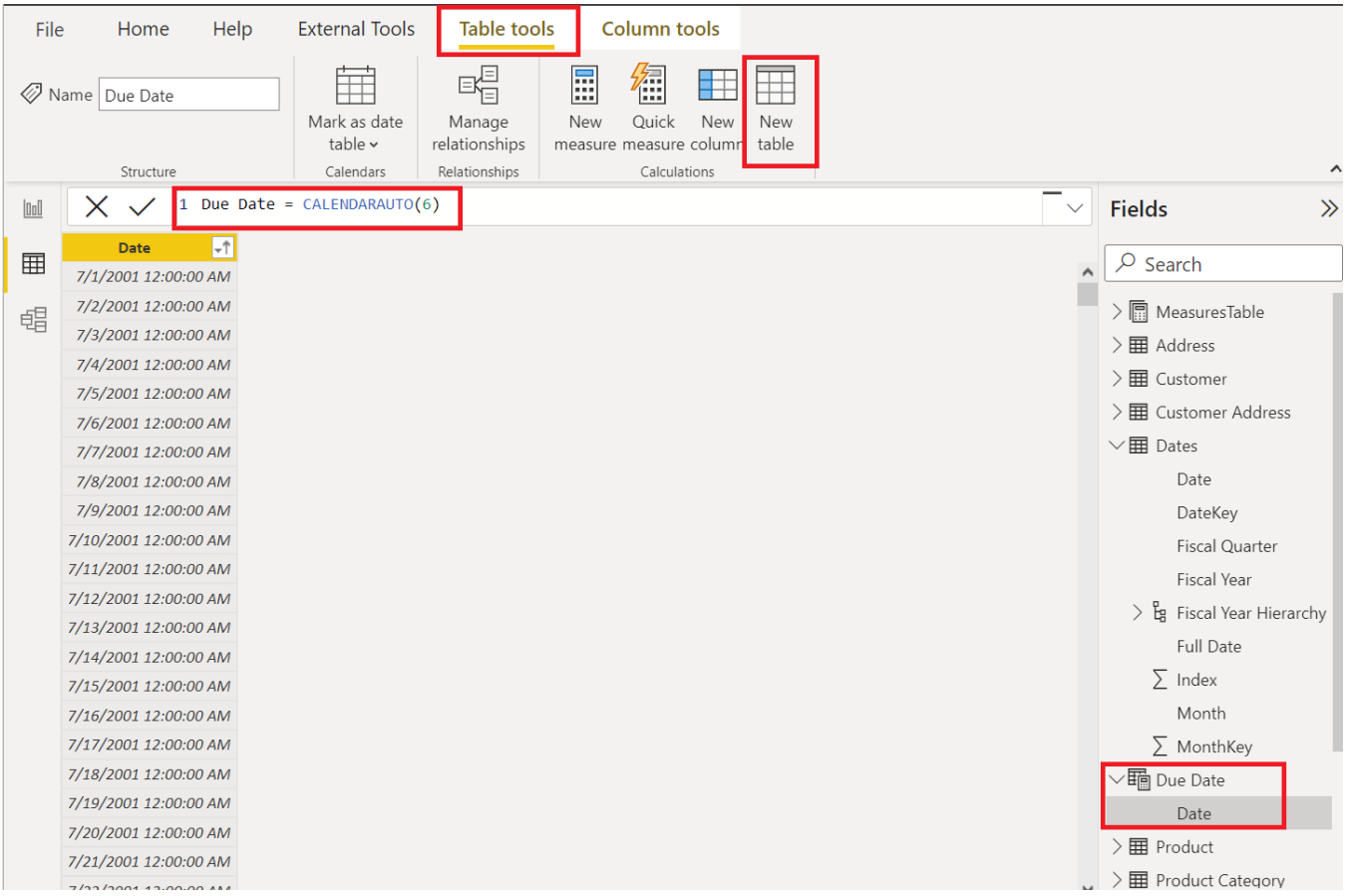
9. At the end of either of the above two steps, we should see active relationship between the two dates and the Sales Order Detail tables; one connecting directly and one passing through the SalesOrderHeader table.
10. A calculated table only duplicates data; it doesn't duplicate any model configurations like column visibility or hierarchies. You'll need to configure them for the new table, if required.

Exercise 2 – Create a Date Table

The next exercise shows how to add a Date Table using DAX Code rather than simply referencing another table.

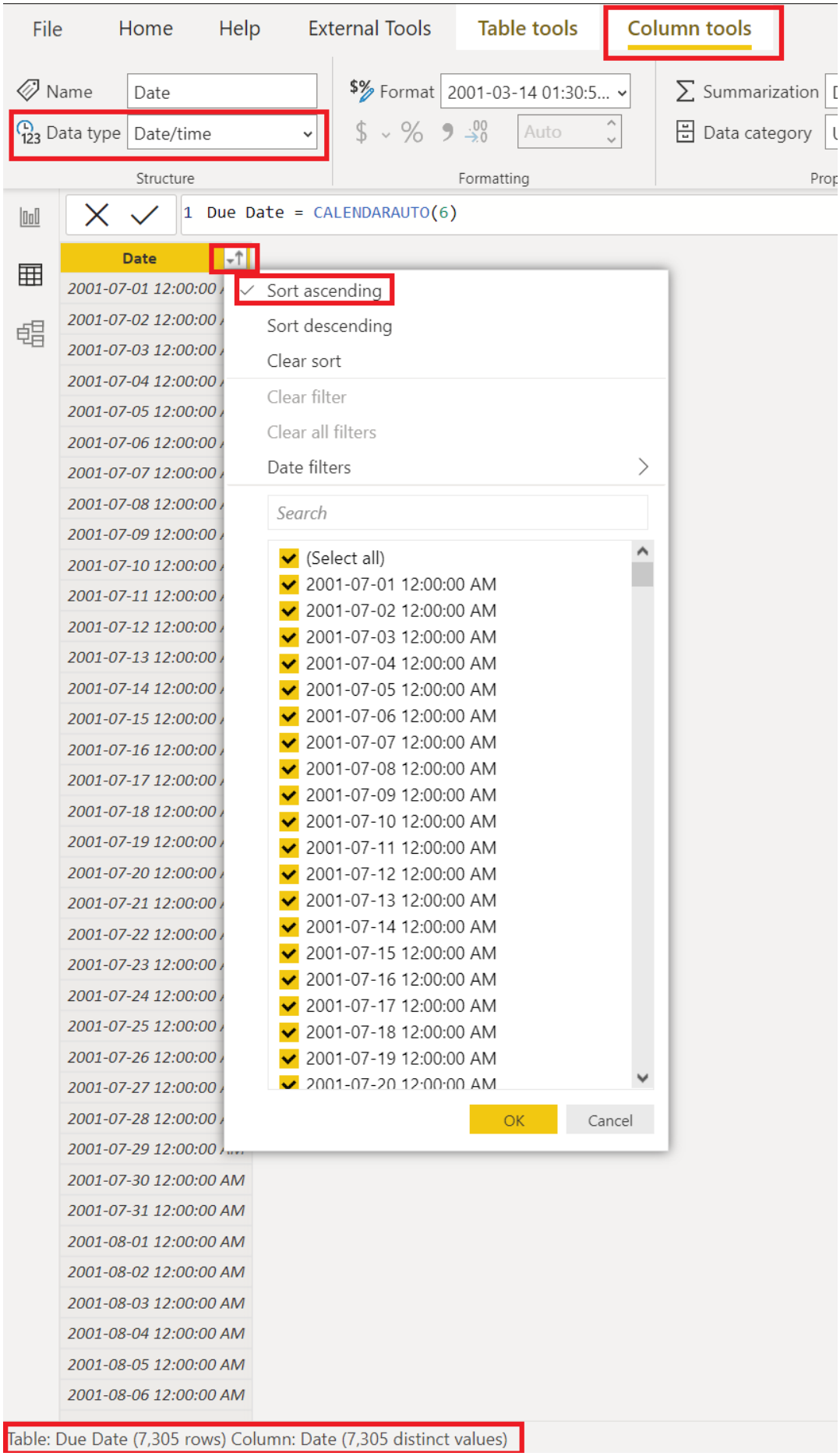
1. Continue with the file from Exercise 1.
2. In the next example, a second calculated table will be created, this time by using the CALENDARAUTO DAX function.
3. Create the Due Date calculated table by using the following definition.

```
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.


File
Home
Help
External Tools
Table tools
Column tools
Name
Date
Data type
Date/time
Format
2001-03-14 01:30:5...
Summarization
Data category
Structure
Formatting
1 Due Date = CALENDARAUTO(6)
Date
Sort ascending
Sort descending
Clear sort
Clear filter
Clear all filters
Date filters
Search
(Select all)
2001-07-01 12:00:00 AM
2001-07-02 12:00:00 AM
2001-07-03 12:00:00 AM
2001-07-04 12:00:00 AM
2001-07-05 12:00:00 AM
2001-07-06 12:00:00 AM
2001-07-07 12:00:00 AM
2001-07-08 12:00:00 AM
2001-07-09 12:00:00 AM
2001-07-10 12:00:00 AM
2001-07-11 12:00:00 AM
2001-07-12 12:00:00 AM
2001-07-13 12:00:00 AM
2001-07-14 12:00:00 AM
2001-07-15 12:00:00 AM
2001-07-16 12:00:00 AM
2001-07-17 12:00:00 AM
2001-07-18 12:00:00 AM
2001-07-19 12:00:00 AM
OK
Cancel
2001-07-30 12:00:00 AM
2001-07-31 12:00:00 AM
2001-08-01 12:00:00 AM
2001-08-02 12:00:00 AM
2001-08-03 12:00:00 AM
2001-08-04 12:00:00 AM
2001-08-05 12:00:00 AM
2001-08-06 12:00:00 AM
Table: Due Date (7,305 rows) Column: Date (7,305 distinct values)

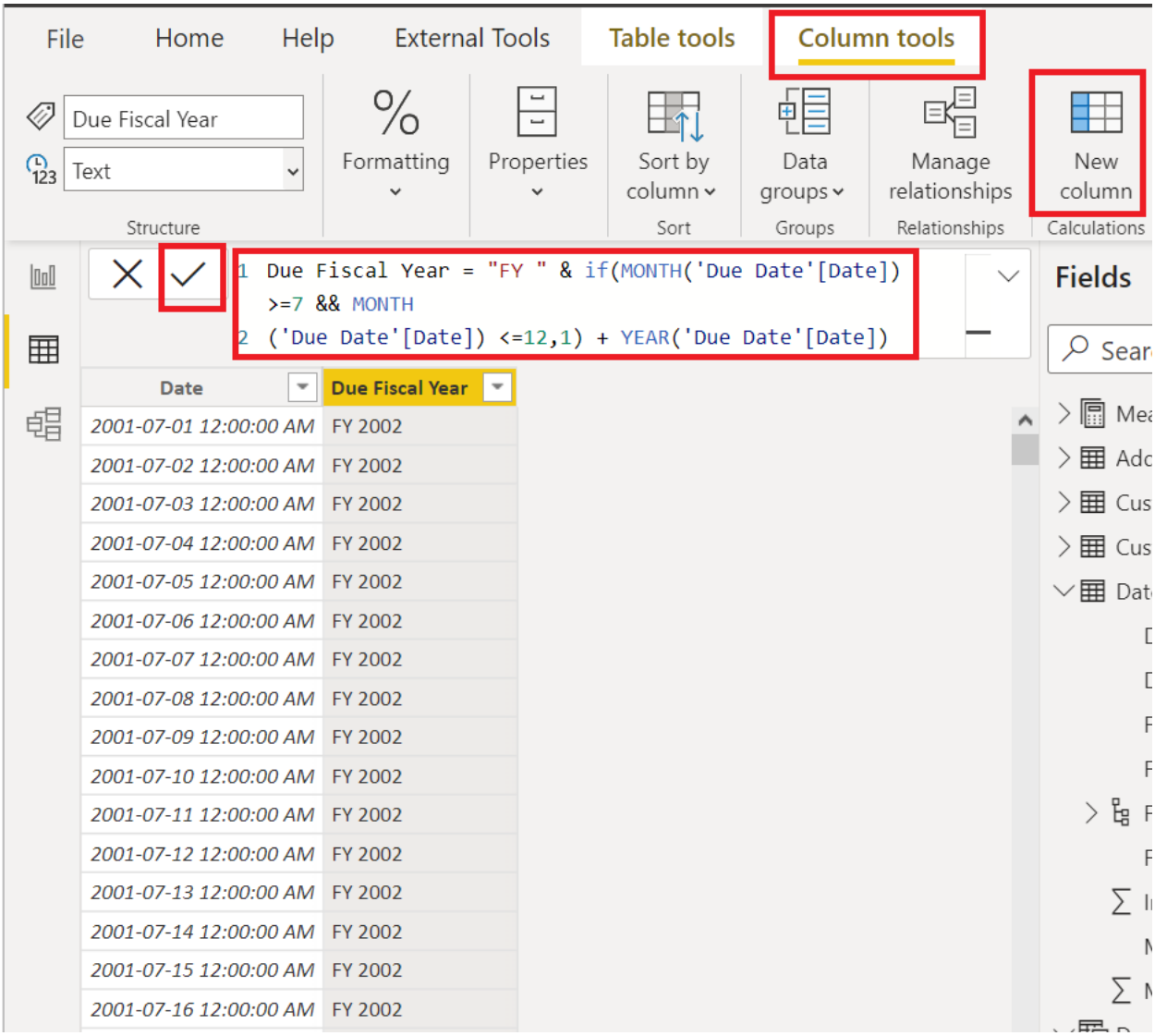
14. Ordering or filtering columns doesn't change how the values are stored.
15. These functions in the data view help you explore and understand the data.
16. Now that the **Date** column is selected, review the message in the status bar (located in the lower-left corner). It describes how many rows 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 SalesOrderDetail 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. To create a calculated column, in the **Table tools** contextual ribbon, from inside the **Calculations** group, select "**New column"**.
4. In the formula box, enter the following calculated column definition and then press the **Enter** key.

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



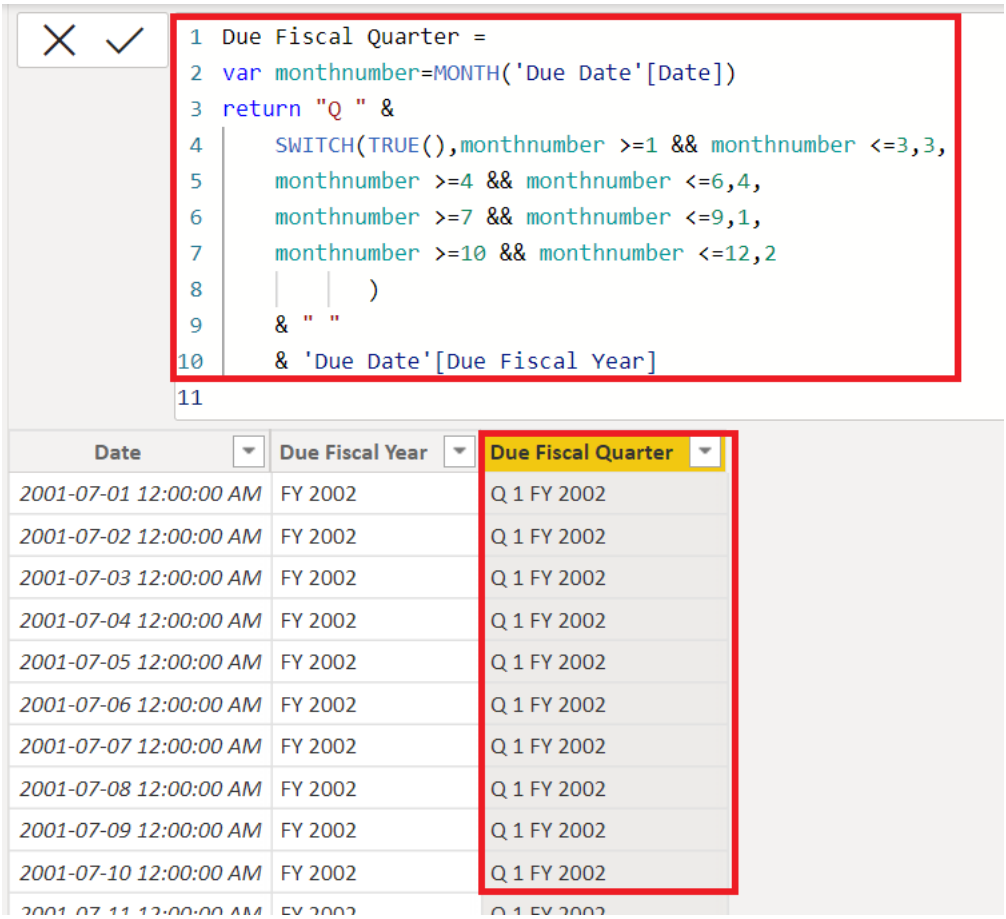
5. The following steps describe how Microsoft Power BI evaluates the calculate column formula:
    a) The addition operator (+) is evaluated before the text concatenation operator (&).
    b) The YEAR DAX function returns the whole number value of the due date year.
    c) The IF DAX function returns 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.

6. 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 JulySeptember), and the result is concatenated to the **Due Fiscal Year** column value and the literal text **Q**.



Date	Due Fiscal Year	Due Fiscal Quarter
2001-07-01 12:00:00 AM	FY 2002	Q 1 FY 2002
2001-07-02 12:00:00 AM	FY 2002	Q 1 FY 2002
2001-07-03 12:00:00 AM	FY 2002	Q 1 FY 2002
2001-07-04 12:00:00 AM	FY 2002	Q 1 FY 2002
2001-07-05 12:00:00 AM	FY 2002	Q 1 FY 2002
2001-07-06 12:00:00 AM	FY 2002	Q 1 FY 2002
2001-07-07 12:00:00 AM	FY 2002	Q 1 FY 2002
2001-07-08 12:00:00 AM	FY 2002	Q 1 FY 2002
2001-07-09 12:00:00 AM	FY 2002	Q 1 FY 2002
2001-07-10 12:00:00 AM	FY 2002	Q 1 FY 2002
2001-07-11 12:00:00 AM	FY 2002	Q 1 FY 2002

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

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

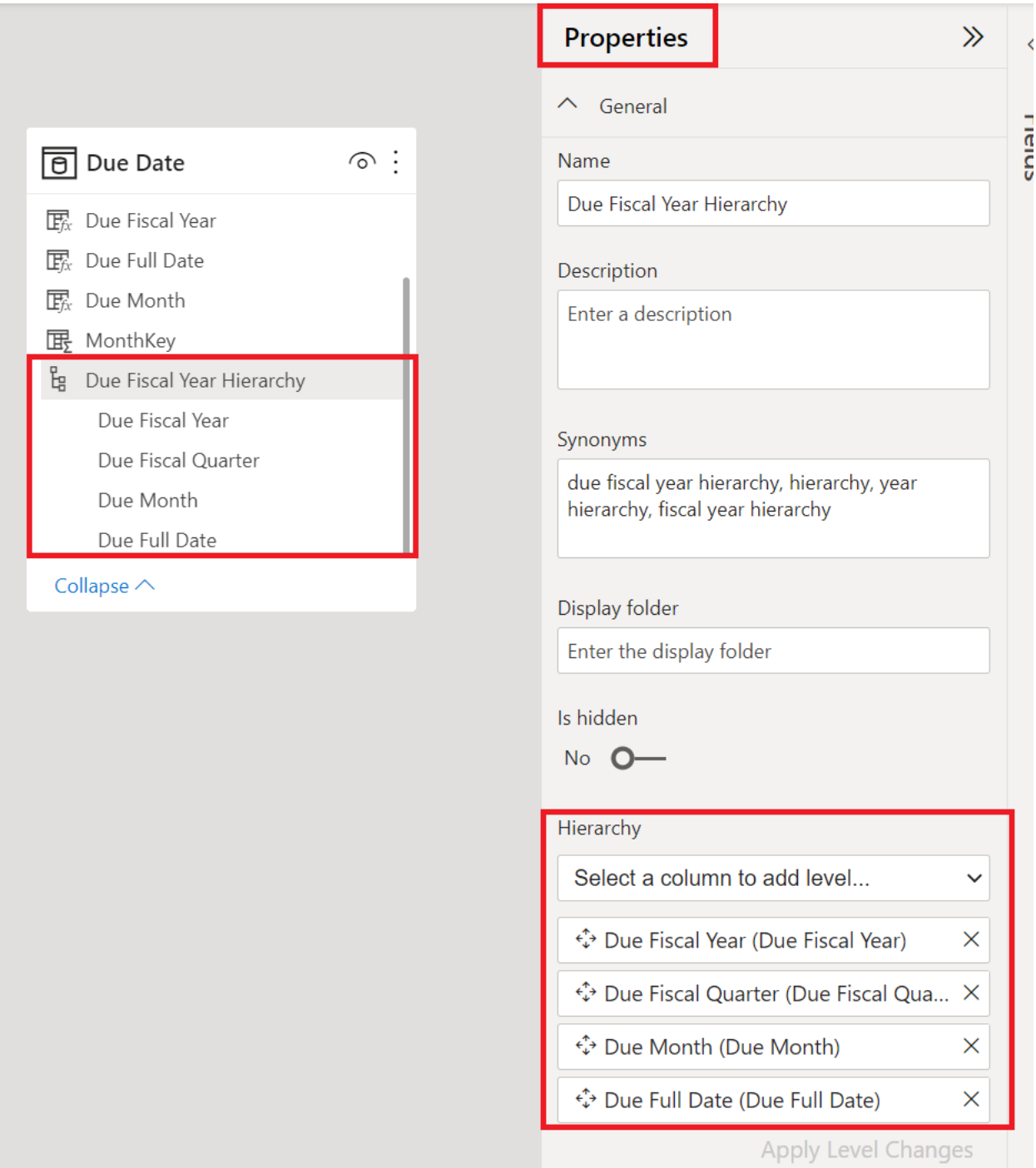
12. The calculated column definition adds the **Due Month** column to the Due Date table.
13. The FORMAT DAX function converts the **Date** column value to text by using a format string. In this case, the format string produces a label that describes the year and abbreviated month name.
14. Add a fourth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

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

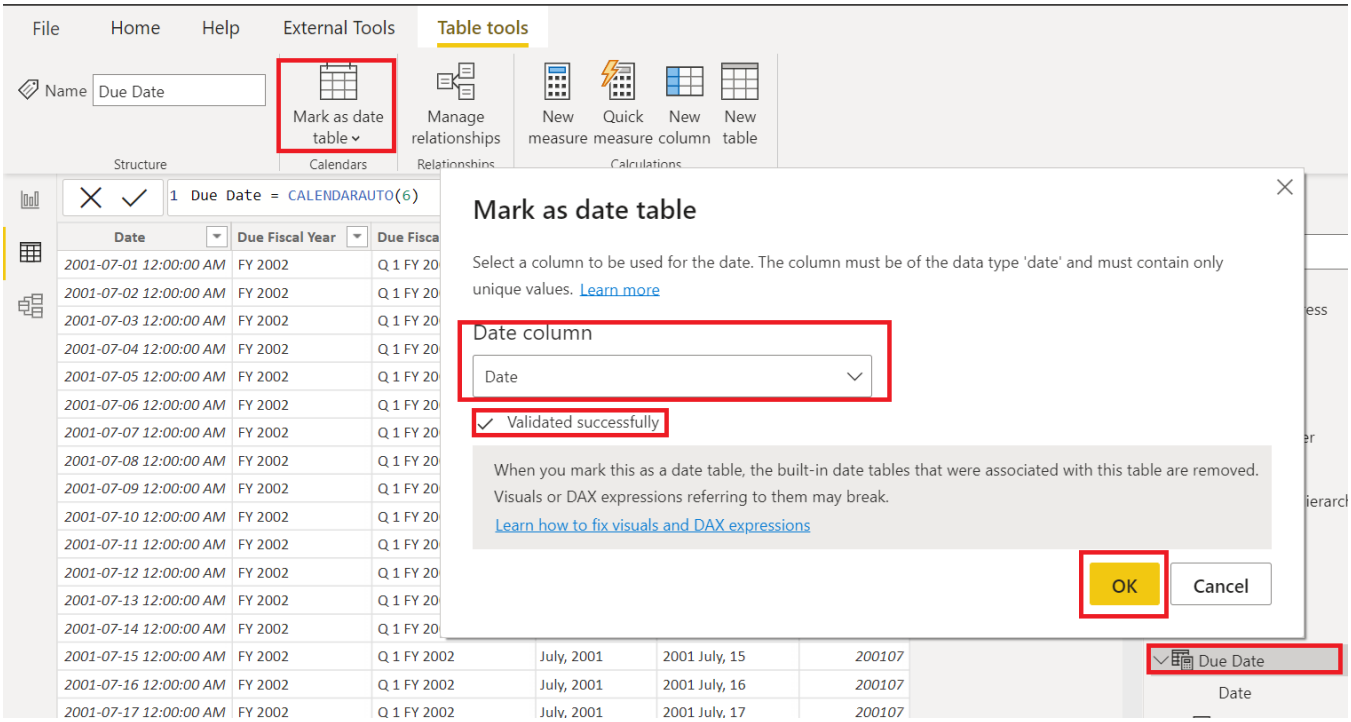
15. Add a fifth calculated column by using the following definition:

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



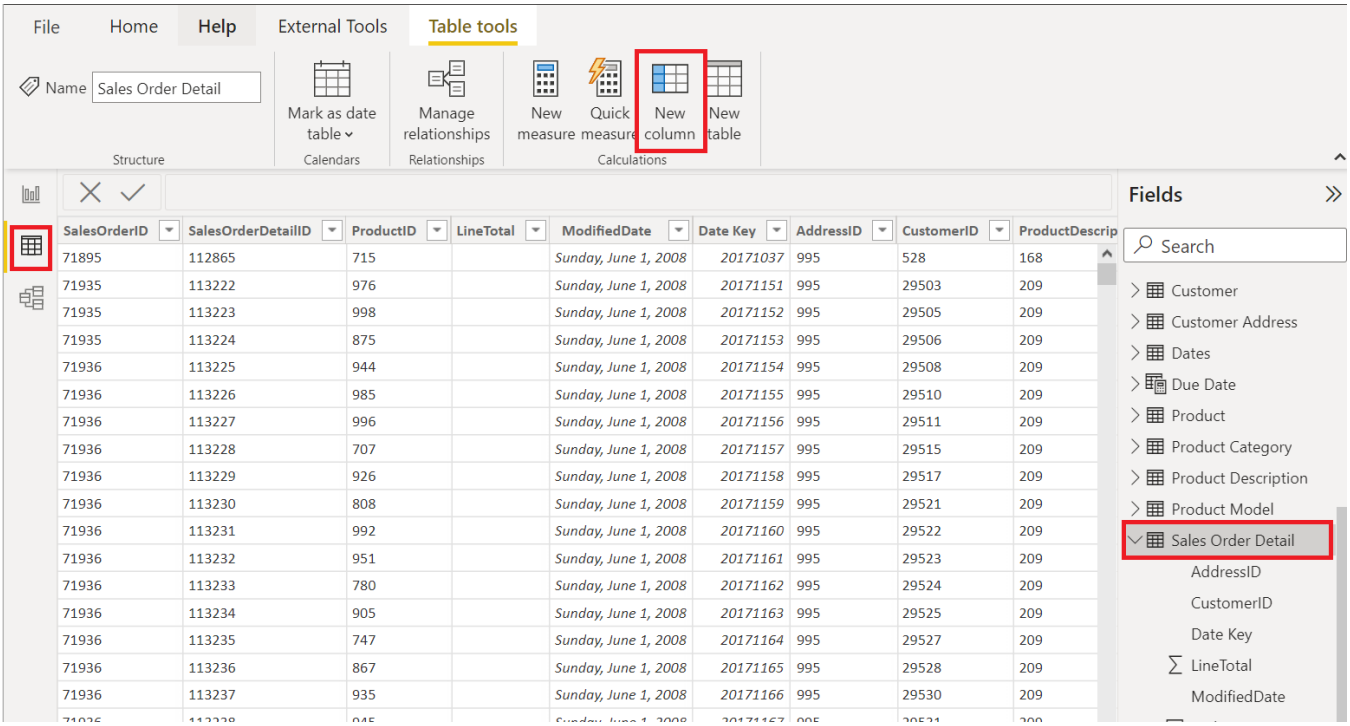
19. Mark the Due Date table as a date table by using the **Due Date** column.



Exercise 4 – Add calculated columns in Sales Order Detail 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 Order Detail table is selected. From “Table tools” select “New Column”



SalesOrderID	SalesOrderDetailID	ProductID	LineTotal	ModifiedDate	Date Key	AddressID	CustomerID	ProductDescrip
71895	112865	715		Sunday, June 1, 2008	20171037	995	528	168
71935	113222	976		Sunday, June 1, 2008	20171151	995	29503	209
71935	113223	998		Sunday, June 1, 2008	20171152	995	29505	209
71935	113224	875		Sunday, June 1, 2008	20171153	995	29506	209
71936	113225	944		Sunday, June 1, 2008	20171154	995	29508	209
71936	113226	985		Sunday, June 1, 2008	20171155	995	29510	209
71936	113227	996		Sunday, June 1, 2008	20171156	995	29511	209
71936	113228	707		Sunday, June 1, 2008	20171157	995	29515	209
71936	113229	926		Sunday, June 1, 2008	20171158	995	29517	209
71936	113230	808		Sunday, June 1, 2008	20171159	995	29521	209
71936	113231	992		Sunday, June 1, 2008	20171160	995	29522	209
71936	113232	951		Sunday, June 1, 2008	20171161	995	29523	209
71936	113233	780		Sunday, June 1, 2008	20171162	995	29524	209
71936	113234	905		Sunday, June 1, 2008	20171163	995	29525	209
71936	113235	747		Sunday, June 1, 2008	20171164	995	29527	209
71936	113236	867		Sunday, June 1, 2008	20171165	995	29528	209
71936	113237	935		Sunday, June 1, 2008	20171166	995	29530	209

3. Use the following DAX Expression in the formula bar and select Ok. `C Revenue = SalesOrderDetail[UnitPrices] * SalesOrderDetail[OrderQty]`

4. Notice the result in the data view and formatting option.
5. Save the Power BI File as Adventure Works M03.pbix