Calendar

In the world of Data Analysis, many reports include dates: presented MONTHLY, QUARTERLY, YEARLY, and others format to find more insight about the data along with creating relation among other data field. As data is useless if can't find any insights, creating a calendar table will accelerate the process of finding insights. The necessity of a dates data is beyond doubt. Power BI provides two dimensions for date:

- Default
- Custom

Default:

This date dimension is active by default. Power BI allows this functionality while creating any report. But what if when multiple date field in multiple tables, definitely this will introduce ambiguity by increasing granularity and complexity. The recommended solution is to use a separate table for storing all needed dates.

Note: In the Power BI Desktop, File Menu > Option and Settings > Options in the Options Window, under Current File, Data Load; Time Intelligence: Auto Date/Time

Custom:

There are a couple of ways to create a Calendar Table (also known as Date Table) which will solve the problems that we will face if we use the default Date dimension. Some of the advantages for using custom date dimension are enlisted below:

- ✓ It allows us to easily extract the minimum & the maximum dates for single or multiple columns. For example, we can use the MIN (), MAX () function to obtain the dates.
- ✓ We can define our own hierarchy for dates. Like sometimes we only need Week along with other date fields, the default date dimension will not provide that particular date filed which is needed for our analysis.
- ✓ We can utilize more effectively the Time Intelligence in here. Like, after extracting Min, Max date, we want to do further analysis, for example, we want to compare sales using MTD () function. What if we don't have a particular date for a particular month of a particular year, but we have that exact date in another month of another year. This issue can be solved in this custom date dimension.

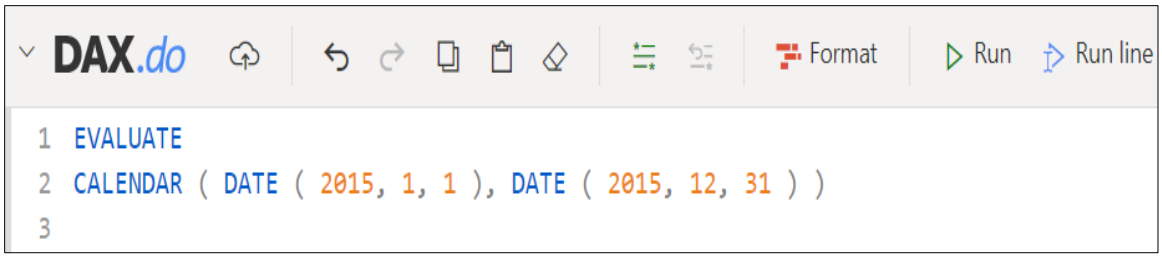
Creating Calendar using DAX, SQL and Excel

1. DAX CALENDAR Function:

Step 1: Open dax.do if you want to use dax studio online. If you have an installed version of dax studio then open it, if not then you can download it from here.

Step 2: Create a static calendar by using CALENDAR function by writing the code below

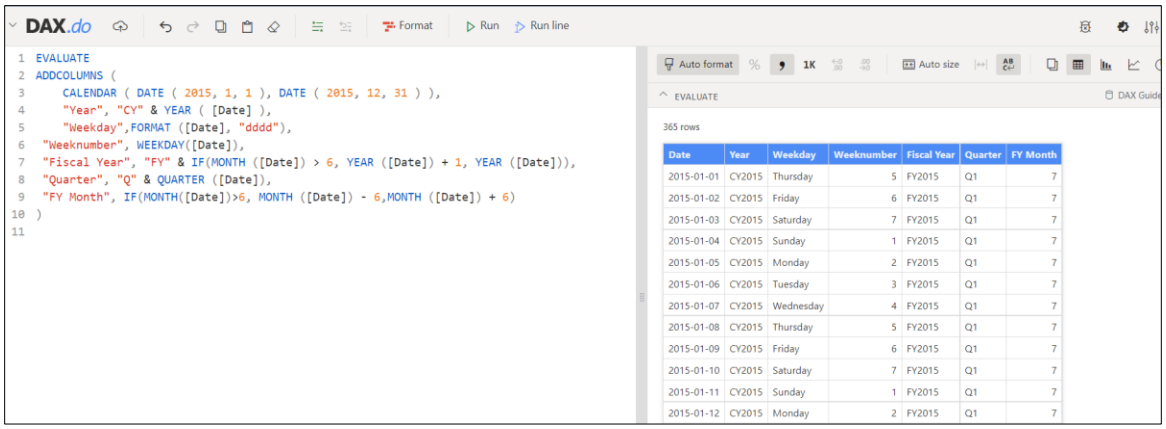
Note: If you are using power bi desktop then no need to use 'EVALUATE'. The keyword is only for dax studio editor.



The above code will generate a calendar from January 2015 to December 2015.

Output:

Step 3: Adding additional columns

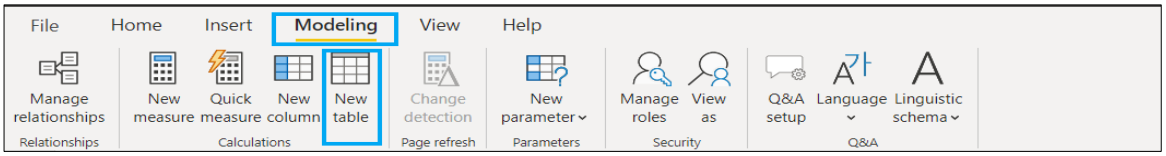


Full Code with comment:

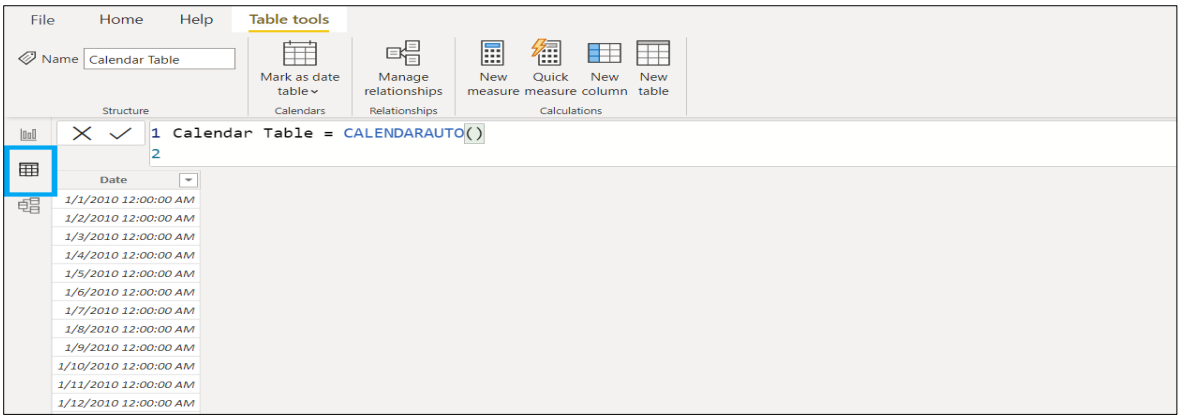
```
EVALUATE
ADDCOLUMNS (
    // creating a custom calender from 2015 january to 2015 december
    CALENDAR (
        DATE ( 2015, 1, 1 ),
        DATE ( 2015, 12, 31 )
    ),
    // adding additional column named 'Year' and 'CY' as prefix
    // DAX YEAR function returns the year of the current date
    "Year", "CY" & YEAR ( [Date] ),
    // adding another column named 'Weekday'
    // DAX FORMAT function formats the date according to the given format
    "Weekday", FORMAT ( [Date], "dddd" ),
    // adding 'Weeknumber' column by using DAX WEEKDAY function that returns
    // the day of the week of the given date.
    "Weeknumber", WEEKDAY ( [Date] ),
    // Using IF condition to calculate the fiscal year
    "Fiscal Year",
        "FY"
            & IF ( MONTH ( [Date] ) > 6, YEAR ( [Date] ) + 1, YEAR ( [Date] ) ),
    // Using DAX QUARTER function to extract quarter from the given data
    "Quarter", "Q" & QUARTER ( [Date] ),
    // Calculating fiscal year month no by using IF condition
    // note that fiscal year starts from July
    "FY Month",
        IF ( MONTH ( [Date] ) > 6, MONTH ( [Date] ) - 6, MONTH ( [Date] ) + 6 ))
```

Creating a Calendar in Power BI Desktop

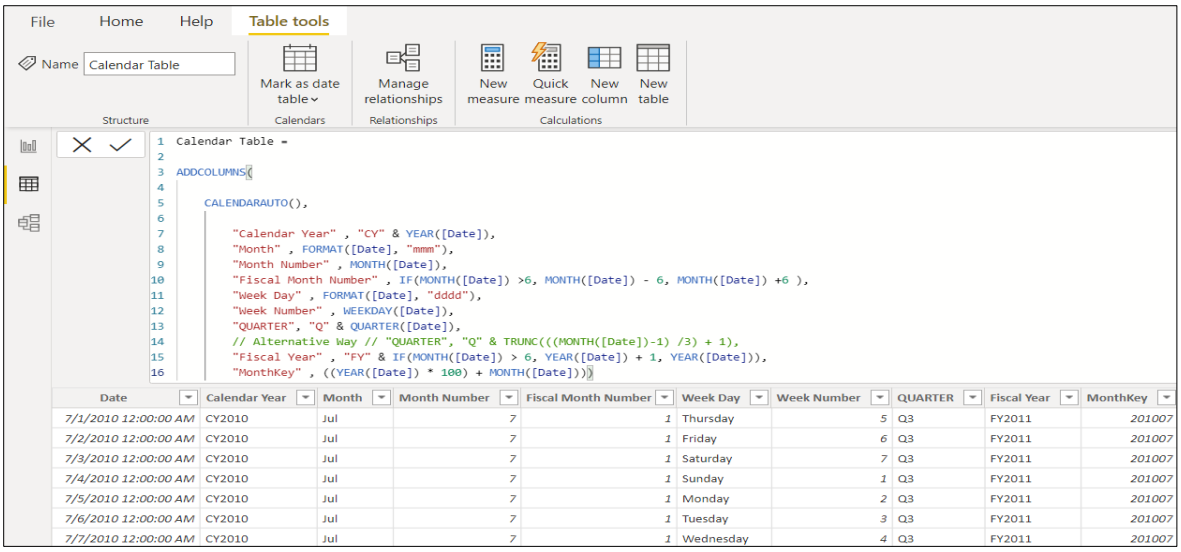
Step 1: In Power BI Desktop, to create a Calendar Table, go to the Modelling section from the ribbon and select New table.



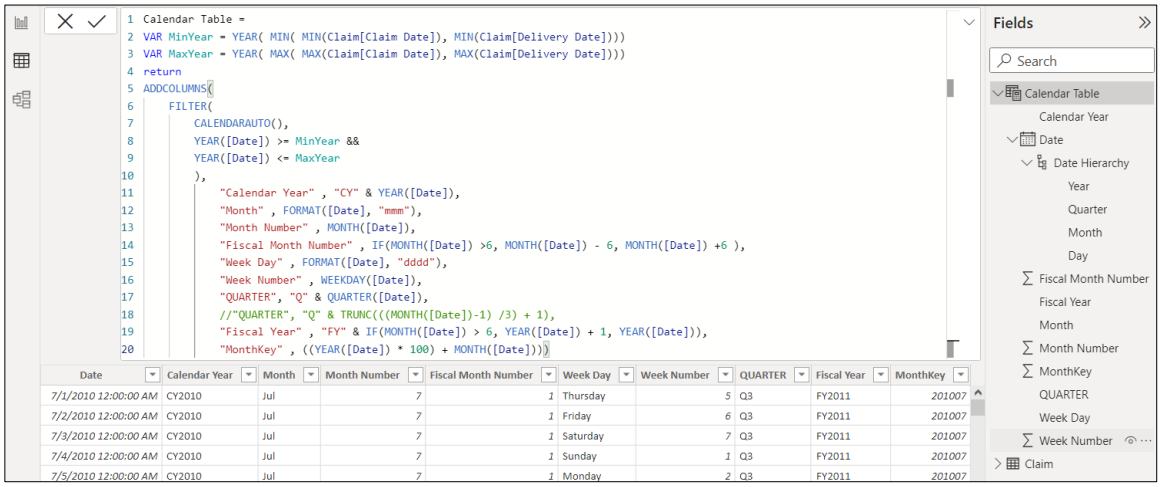
Step 2: There will come a coding section, at first, give a name to the table. There are built-in functions in Power BI Desktop to create Date-Time like CALENDAR() and CALENDARAUTO(). CALENDARAUTO() works based on the time intelligence of Power BI. It will search on its own in the data model to fetch dates from several columns and make a date column.



Step 3: From the Date column, several different columns like Year, Day, Week Number, Fiscal Year could be added using the ADDCOLUMNS() function.



Step 4: To be precise for starting and ending dates an additional filter has been used to avoid unintended dates. Suppose in a sales dataset, there is a column where the customer's birth date is stored. Power BI Time Intelligence will detect and start from that date. But Sales Analysis has no connection with that date. In that case, we manually choose one or multiple columns and declare a Variable (VAR) to return the minimum and maximum value. Filter() function has been applied to the CALENDARAUTO() to avoid any date which is in period of MinYear and MaxYear (Variable to store the min and max year of the respected columns).



The below code will create a generate a Calendar Table using CALENDARAUTO() DAX function.

```
Calendar Table =
VAR MinYear =
    YEAR ( MIN ( MIN ( Claim[Claim Date] ), MIN ( Claim[Delivery Date] ) ) )
VAR MaxYear =
    YEAR ( MAX ( MAX ( Claim[Claim Date] ), MAX ( Claim[Delivery Date] ) ) )
RETURN
    ADDCOLUMNS (
        FILTER (
            CALENDARAUTO (),
            YEAR ( [Date] ) >= MinYear
                && YEAR ( [Date] ) <= MaxYear
        ),
        "Calendar Year", "CY" & YEAR ( [Date] ),
        "Month", FORMAT ( [Date], "mmm" ),
        "Month Number", MONTH ( [Date] ),
        "Fiscal Month Number",
            IF ( MONTH ( [Date] ) > 6, MONTH ( [Date] ) - 6, MONTH ( [Date] ) + 6
),
        "Week Day", FORMAT ( [Date], "dddd" ),
        "Week Number", WEEKDAY ( [Date] ),
        "QUARTER", "Q" & QUARTER ( [Date] ),
        //"QUARTER", "Q" & TRUNC(((MONTH([Date])-1) /3) + 1),
        "Fiscal Year",
            "FY"
                & IF ( MONTH ( [Date] ) > 6, YEAR ( [Date] ) + 1, YEAR ( [Date] )
),
        "MonthKey",
            (
                ( YEAR ( [Date] ) * 100 )
                    + MONTH ( [Date] )
            )
    )
```

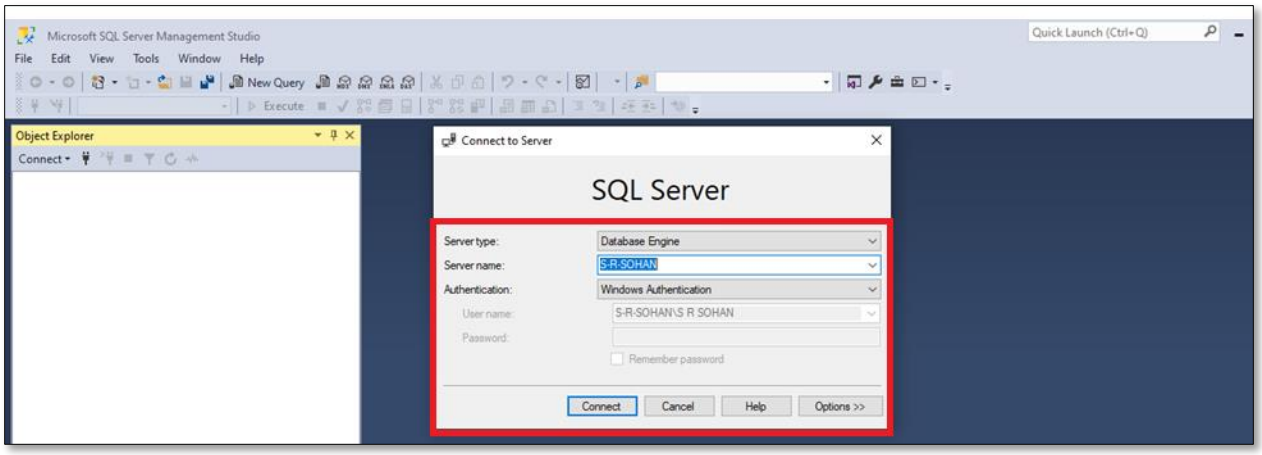
Create a Calendar or Date Table in SQL Server

As, we already know that, some techniques that, optimize query performance of Power BI. One of them is:

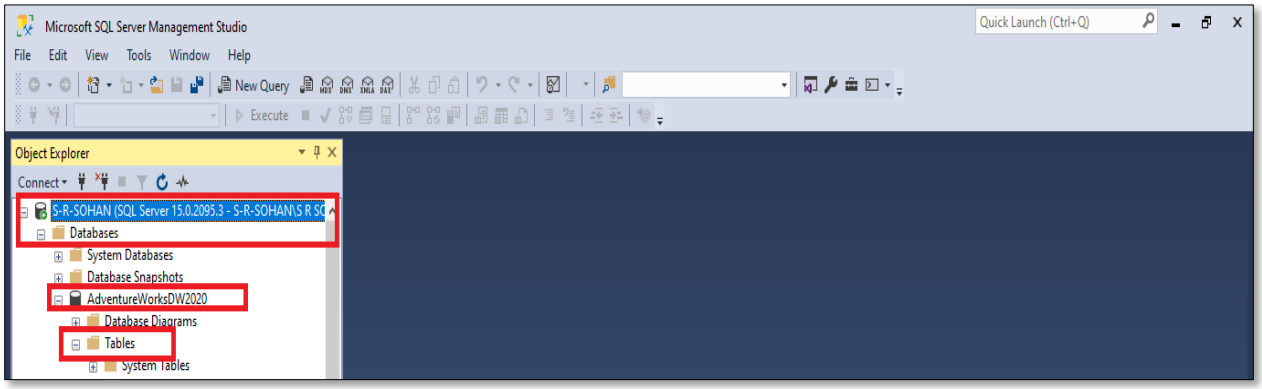
- **Process as much data as possible in the original data source.** Power Query and Power Query Editor allow to process the data; however, the processing power that is required to complete this task might lower performance in other areas of our reports. Generally, **a good practice is to process, as much as possible, in the native data source.**

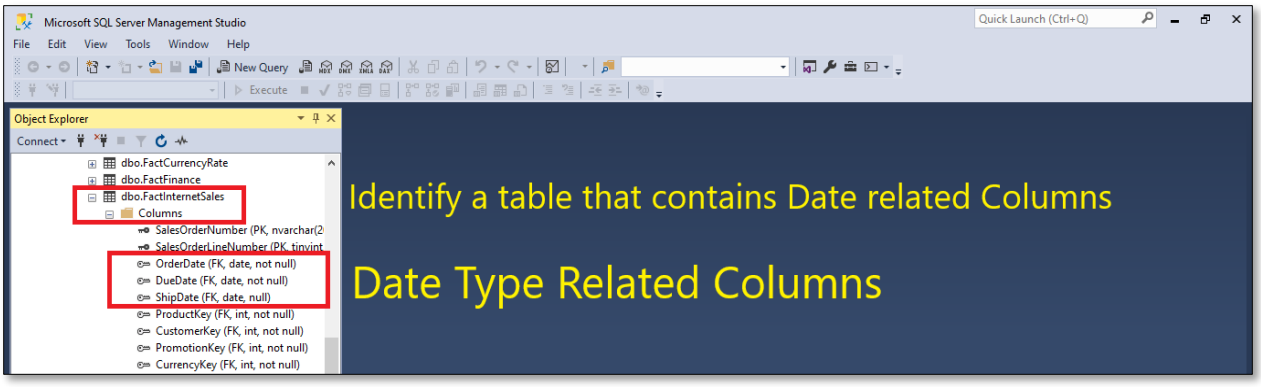
That's why we can create a custom date or calendar table in our **Data Source to optimize Performance of Power BI**. Let's see how to create a Custom Calendar Table in SQL Server.

1. Connect the SQL Server Database.

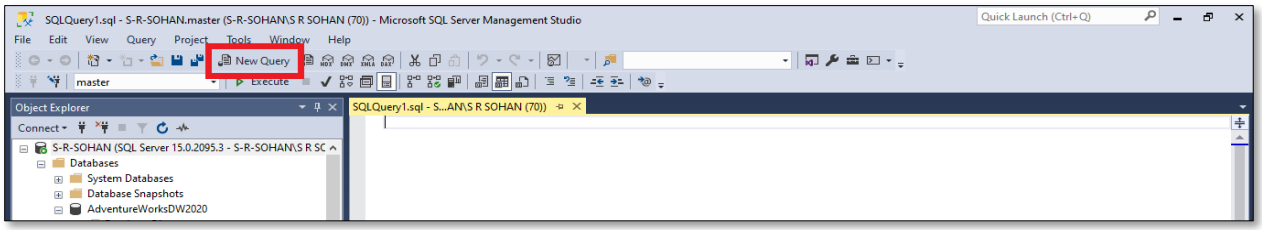


2. Load the desired database

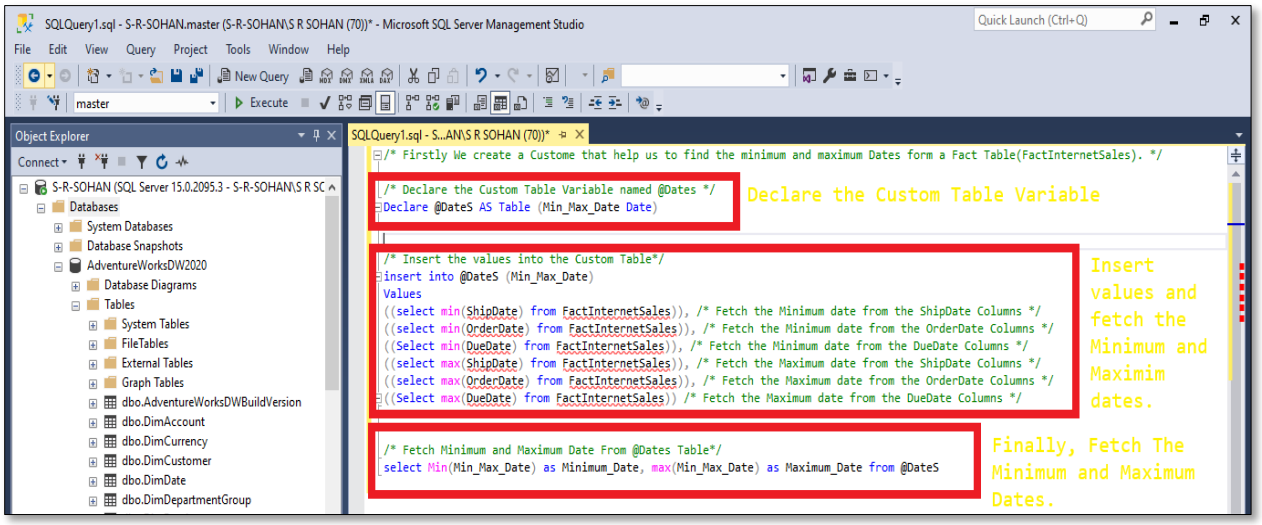




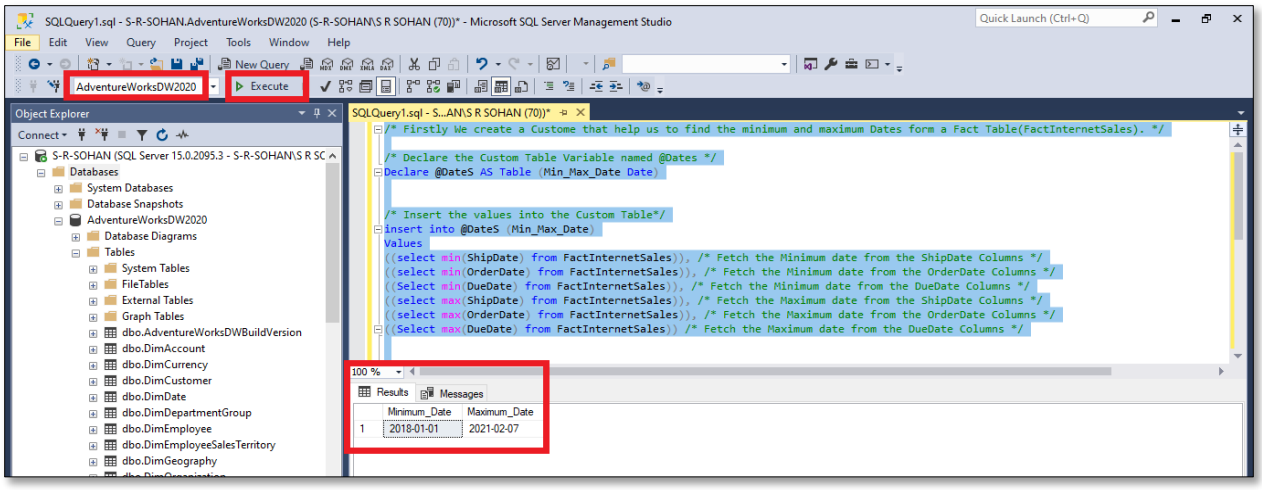
3. Now, find out the date type related column form that tables. After finding out those columns then find the minimum and maximum date from that column.
4. After Identify all dates colums, click the “New Query” option. It will open a new widow, where we can write SQL Code.



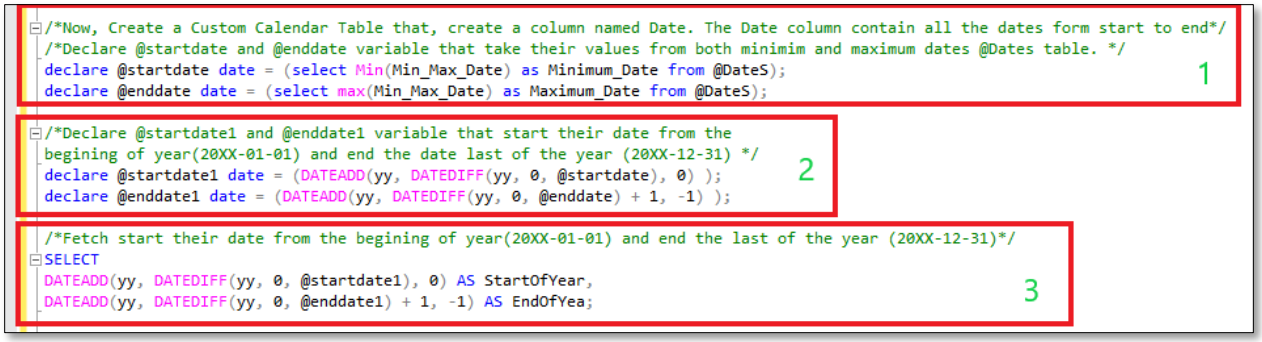
5. Firstly, We create a Custom table that, find the Minimum and Maximum Dates form a Fact Table (such asFactInternetSales).



6. Now, execute the query, after executing the query we will gate the minimum and maximum result from the Fact table (FactInternetSales).



7. Now, create a custom table that, take Minimum_Date and Maximum_Date from the @Dates Table. This Table option describe below step by step:
    1. **Step 01:** Declare @startdate and @enddate variables that store Minimum_Date and Maximum_Date respectively.
    2. **Step 02:** Declare another variable @startdate1 and @enddate1 that, will start the date from the beginning of the year (20XX-01-01) and end the date last of the year (20XX-12-31). That's very helpful for the time intelligence analysis.
    3. **Step 03:** Fetch the Dates from the start of the year and end of the year from the previous step.



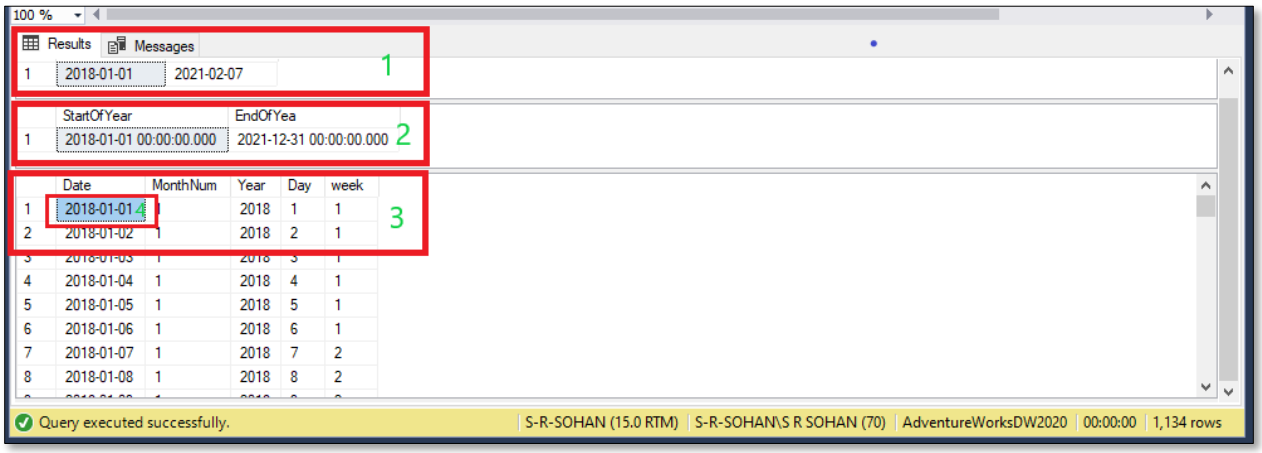
8. Create another Custom table Called Calendar
    1. **Step 01:** Calendar table that, fetch all dates recursively from start of the year to end of the year.
    2. **Step 02:** Fetch that's all of those dates from start of the year and of the year.

```sql
/*Create Calendar Table that, fetch recursively all dates form the
@startdate1(begining of year(20XX-01-01) to @enddate1last of the year (20XX-12-31) */
with Calendar as
(select @startdate1 as [Date]
UNION ALL
select DATEADD(dd,1,[Date])
from Calendar
where DATEADD(dd,1,[Date]) <= @enddate1)

/* Select the Date Column from Calendar table */
Select [Date],MONTH([Date]) as MonthNum, YEAR([Date]) as [Year], DAY([Date]) as [Day], Datepart(week,[Date]) as [week] from Calendar
/* By Default, Maximum Recusion limit 100 times, that's why we set the limit Infinity[0 means infinity] Recursion Function*/
Option (MAXRECURSION 0) /* Recusion Function */
```

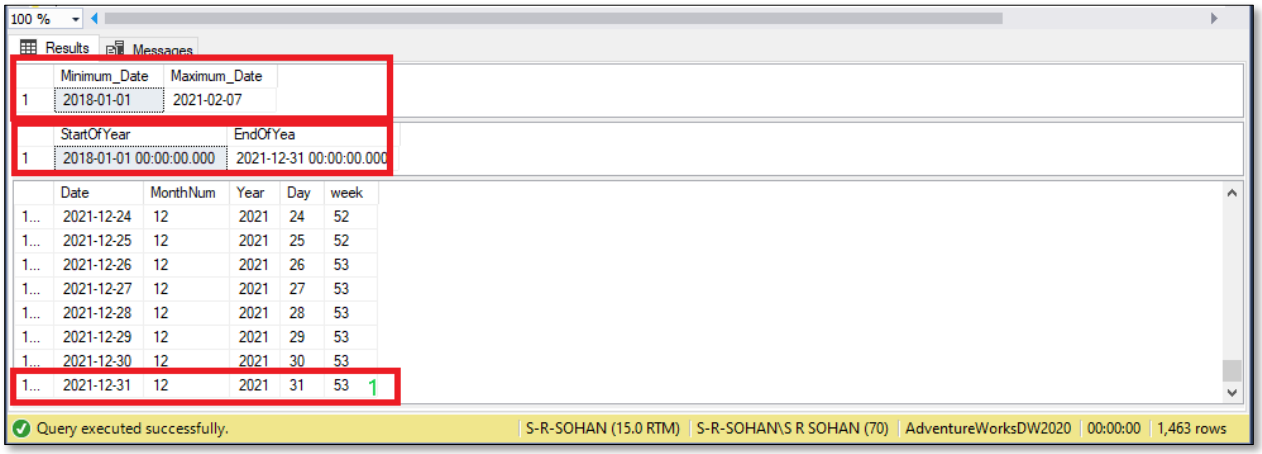
1

2

9. After Executing the SQL query, it will return 3 parts.
    1. **Step 01:** Fetch the Minimum and Maximum Date from the Fact Table (FactInternetSales).
    2. **Step 02:** Convert the Minimum and Maximum Date into start of the year and end of the year.
    3. **Step 03:** Check is Minimum_Date converts into Start of the year (2018-01-01). As, it's by default date is (2018-01-01) so that, we don't differentiate properly. If we will put any random date such as (20XX-06-1) it will give the same result such as (20XX-01-01)



10. Now, Checking is Maximum_Date converts into end of the year. That's means, Maximum_Date (2021-02-07) is replaced EndOfYear (2012-12-31).



Finally, Successfully Create a Date or Calendar table on our Data Source. It will increase the performance of Power BI.

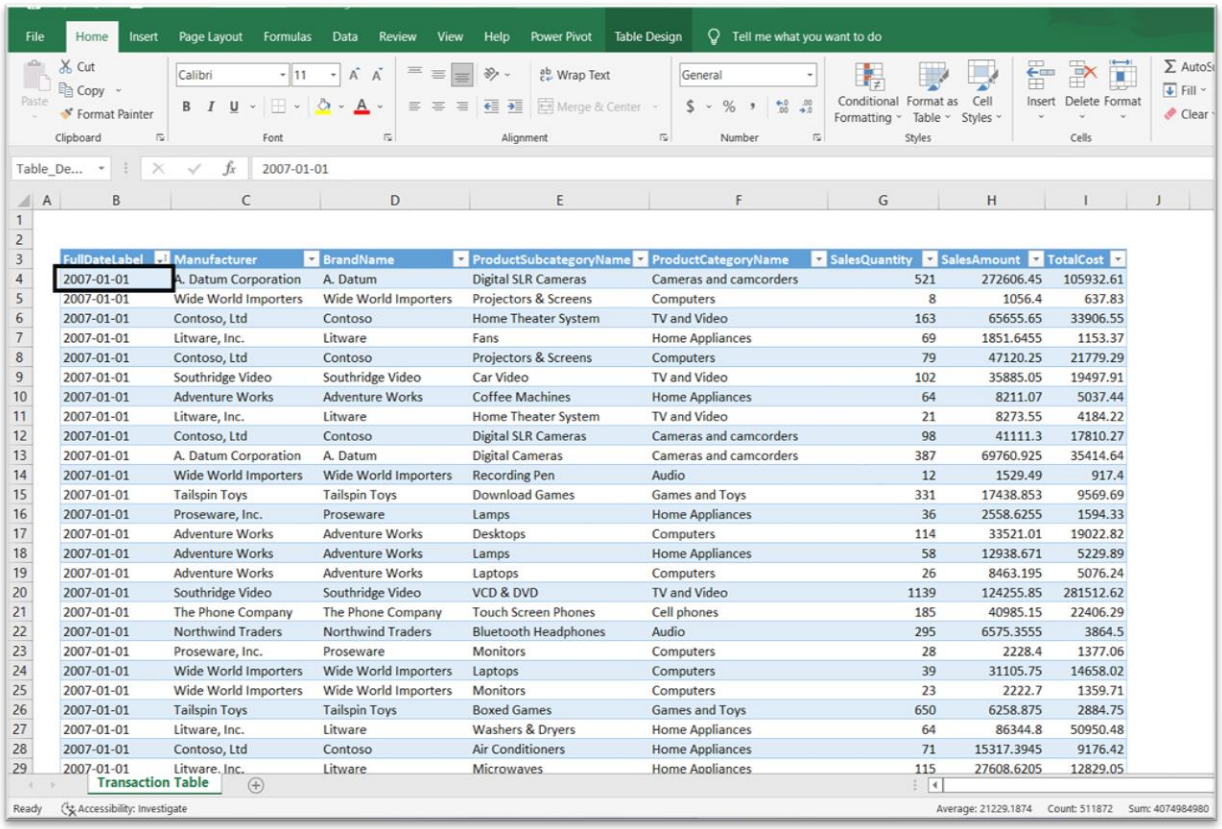
SQL Query

```
/* Firstly We create a Custome that help us to find the minimum and maximum Dates form a Fact Table(FactInternetSales). */

/* Declare the Custom Table Variable named @Dates */
Declare @DateS AS Table (Min_Max_Date Date)

/* Insert the values into the Custom Table*/
insert into @DateS (Min_Max_Date)
Values
((select min(ShipDate) from FactInternetSales)), /* Fetch the Minimum date from the ShipDate Columns */
((select min(OrderDate) from FactInternetSales)), /* Fetch the Minimum date from the OrderDate Columns */
((Select min(DueDate) from FactInternetSales)), /* Fetch the Minimum date from the DueDate Columns */
((select max(ShipDate) from FactInternetSales)), /* Fetch the Maximum date from the ShipDate Columns */
((select max(OrderDate) from FactInternetSales)), /* Fetch the Maximum date from the OrderDate Columns */
((Select max(DueDate) from FactInternetSales)) /* Fetch the Maximum date from the DueDate Columns */

/* Fetch Minimum and Maximum Date From @Dates Table*/
select Min(Min_Max_Date) as Minimum_Date, max(Min_Max_Date) as Maximum_Date from @DateS

/*Now, Create a Custom Calendar Table that, create a column named Date. The Date column contain all the dates form start to end*/
/*Declare @startdate and @enddate variable that take their values from both minimim and maximum dates @Dates table. */
declare @startdate date = (select Min(Min_Max_Date) as Minimum_Date from @DateS);
declare @enddate date = (select max(Min_Max_Date) as Maximum_Date from @DateS);

/*Declare @startdate1 and @enddate1 variable that start their date from the
begining of year(20XX-01-01) and end the date last of the year (20XX-12-31) */
declare @startdate1 date = (DATEADD(yy, DATEDIFF(yy, 0, @startdate), 0) );
declare @enddate1 date = (DATEADD(yy, DATEDIFF(yy, 0, @enddate) + 1, -1) );

/*Fetch start their date from the begining of year(20XX-01-01) and end the last of the year (20XX-12-31)*/
SELECT
DATEADD(yy, DATEDIFF(yy, 0, @startdate1), 0) AS StartOfYear,
DATEADD(yy, DATEDIFF(yy, 0, @enddate1) + 1, -1) AS EndOfYea;

/*Create Calendar Table that, fetch recursively all dates form the
@startdate1(begining of year(20XX-01-01) to @enddate1last of the year (20XX-12-31) */
with Calendar as
(select @startdate1 as [Date]
UNION ALL
select DATEADD(dd,1,[Date])
from Calendar
where DATEADD(dd,1,[Date]) <= @enddate1)

/* Select the Date Column from Calendar table */
Select [Date],MONTH([Date]) as MonthNum, YEAR([Date]) as [Year], DAY([Date]) as [Day], Datepart(week,[Date]) as
[week] from Calendar
/* By Default, Maximum Recusion limit 100 times, that's why we set the limit Infinity[0 means infinity] Recursion Function*/
Option (MAXRECURSION 0) /* Recusion Function */
```

Dimension Calendar Table in Excel and load it into Power BI

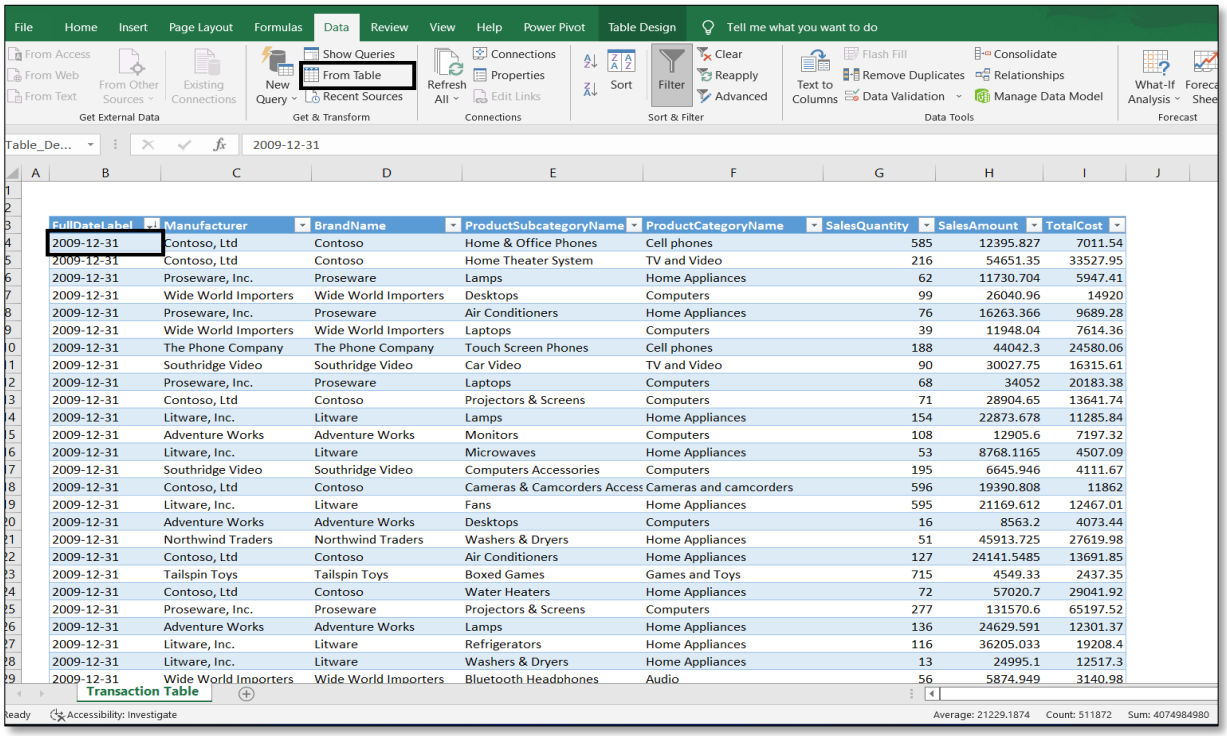
We often need to create a calendar table for time series analysis. We can do it in several ways like in the Source Database or in Power Query. We can easily create a calendar table using Power Pivot in Excel and load the model into power Bi for further use by following the steps.

Suppose we have a Transaction table that has a date column named 'FullDateLabel'. The column ranges from 01/01/2007 to 31/12/2009. We want to make a calendar table based on the 'FullDatelabel'.



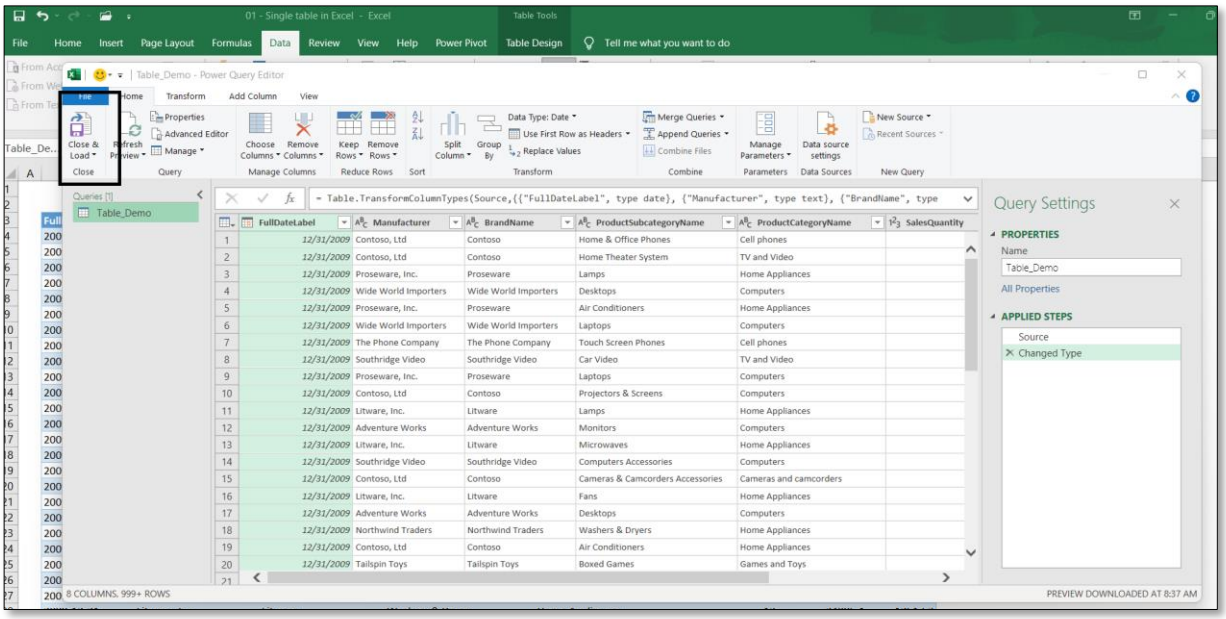
FullDateLabel	Manufacturer	BrandName	ProductSubcategoryName	ProductCategoryName	SalesQuantity	SalesAmount	TotalCost
2007-01-01	A. Datum Corporation	A. Datum	Digital SLR Cameras	Cameras and camcorders	521	272606.45	105932.61
2007-01-01	Wide World Importers	Wide World Importers	Projectors & Screens	Computers	8	1056.4	637.83
2007-01-01	Contoso, Ltd	Contoso	Home Theater System	TV and Video	163	65655.65	33906.55
2007-01-01	Litware, Inc.	Litware	Fans	Home Appliances	69	1851.6455	1153.37
2007-01-01	Contoso, Ltd	Contoso	Projectors & Screens	Computers	79	47120.25	21779.29
2007-01-01	Southridge Video	Southridge Video	Car Video	TV and Video	102	35885.05	19497.91
2007-01-01	Adventure Works	Adventure Works	Coffee Machines	Home Appliances	64	8211.07	5037.44
2007-01-01	Litware, Inc.	Litware	Home Theater System	TV and Video	21	8273.55	4184.22
2007-01-01	Contoso, Ltd	Contoso	Digital SLR Cameras	Cameras and camcorders	98	41111.3	17810.27
2007-01-01	A. Datum Corporation	A. Datum	Digital Cameras	Cameras and camcorders	387	69760.925	35414.64
2007-01-01	Wide World Importers	Wide World Importers	Recording Pen	Audio	12	1529.49	917.4
2007-01-01	Tailspin Toys	Tailspin Toys	Download Games	Games and Toys	331	17438.853	9569.69
2007-01-01	Proseware, Inc.	Proseware	Lamps	Home Appliances	36	2558.6255	1594.33
2007-01-01	Adventure Works	Adventure Works	Desktops	Computers	114	33521.01	19022.82
2007-01-01	Adventure Works	Adventure Works	Lamps	Home Appliances	58	12938.671	5229.89
2007-01-01	Adventure Works	Adventure Works	Laptops	Computers	26	8463.195	5076.24
2007-01-01	Southridge Video	Southridge Video	VCD & DVD	TV and Video	1139	124255.85	281512.62
2007-01-01	The Phone Company	The Phone Company	Touch Screen Phones	Cell phones	185	40985.15	22406.29
2007-01-01	Northwind Traders	Northwind Traders	Bluetooth Headphones	Audio	295	6575.3555	3864.5
2007-01-01	Proseware, Inc.	Proseware	Monitors	Computers	28	2228.4	1377.06
2007-01-01	Wide World Importers	Wide World Importers	Laptops	Computers	39	31105.75	14658.02
2007-01-01	Wide World Importers	Wide World Importers	Monitors	Computers	23	2222.7	1359.71
2007-01-01	Tailspin Toys	Tailspin Toys	Boxed Games	Games and Toys	650	6258.875	2884.75
2007-01-01	Litware, Inc.	Litware	Washers & Dryers	Home Appliances	64	86344.8	50950.48
2007-01-01	Contoso, Ltd	Contoso	Air Conditioners	Home Appliances	71	15317.3945	9176.42
2007-01-01	Litware, Inc.	Litware	Microwaves	Home Appliances	115	27608.6205	12829.05

1. First of all, we have to load the table to Power Query. From 'Data' tab select 'From Table'. For simplicity, we are not making any other Transformation here.

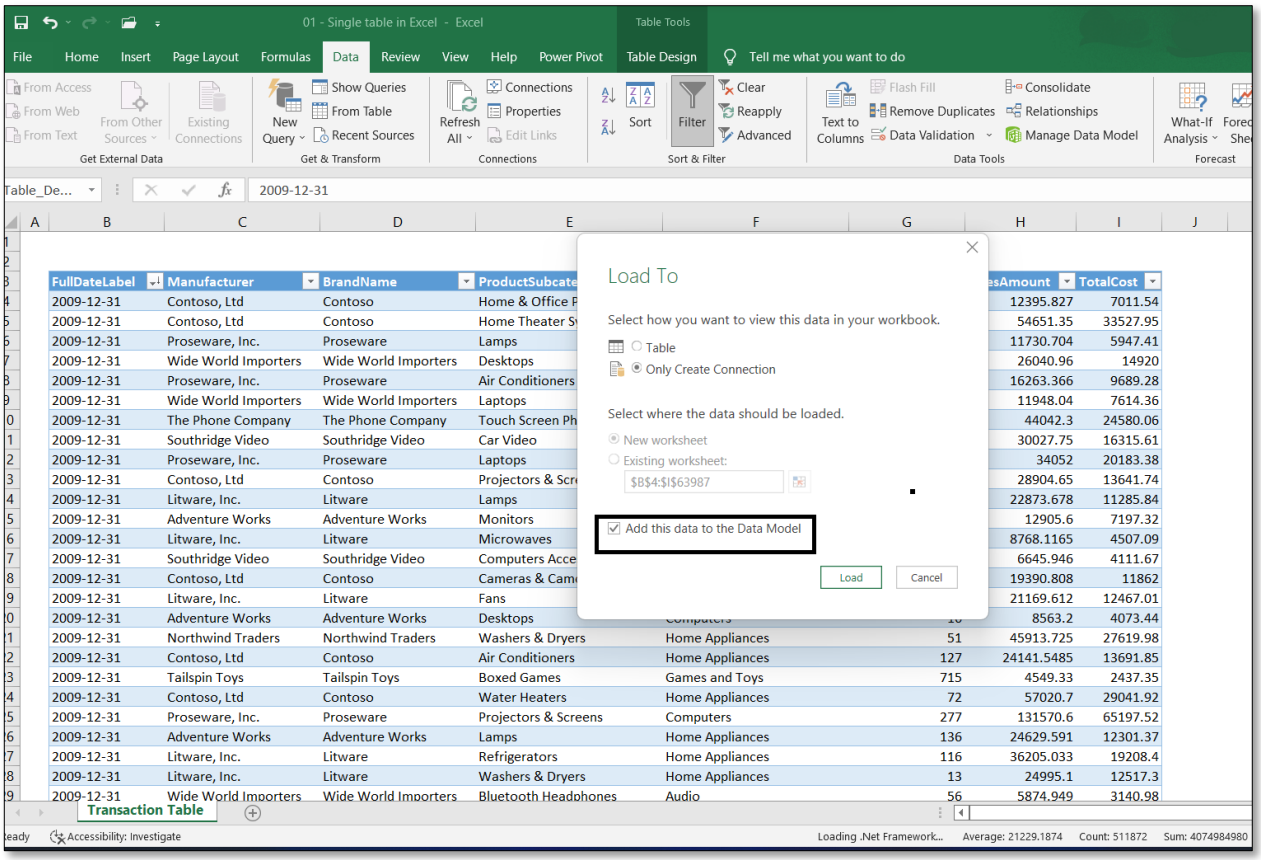


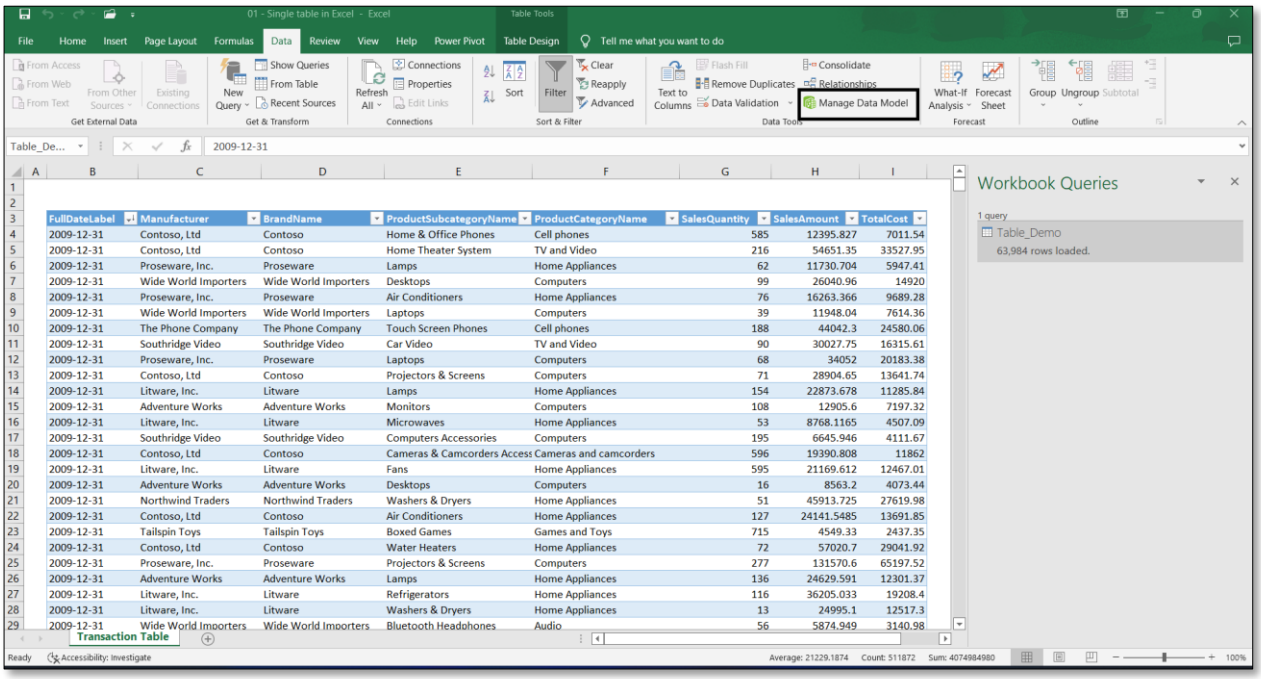
FullDateLabel	Manufacturer	BrandName	ProductSubcategoryName	ProductCategoryName	SalesQuantity	SalesAmount	TotalCost
2009-12-31	Contoso, Ltd	Contoso	Home & Office Phones	Cell phones	585	12395.827	7011.54
2009-12-31	Contoso, Ltd	Contoso	Home Theater System	TV and Video	216	54651.35	33527.95
2009-12-31	Proseware, Inc.	Proseware	Lamps	Home Appliances	62	11730.704	5947.41
2009-12-31	Wide World Importers	Wide World Importers	Desktops	Computers	99	26040.96	14920
2009-12-31	Proseware, Inc.	Proseware	Air Conditioners	Home Appliances	76	16263.366	9689.28
2009-12-31	Wide World Importers	Wide World Importers	Laptops	Computers	39	11948.04	7614.36
2009-12-31	The Phone Company	The Phone Company	Touch Screen Phones	Cell phones	188	44042.3	24580.06
2009-12-31	Southridge Video	Southridge Video	Car Video	TV and Video	90	30027.75	16315.61
2009-12-31	Proseware, Inc.	Proseware	Laptops	Computers	68	34052	20183.38
2009-12-31	Contoso, Ltd	Contoso	Projectors & Screens	Computers	71	28904.65	13641.74
2009-12-31	Litware, Inc.	Litware	Lamps	Home Appliances	154	22873.678	11285.84
2009-12-31	Adventure Works	Adventure Works	Monitors	Computers	108	12905.6	7197.32
2009-12-31	Litware, Inc.	Litware	Microwaves	Home Appliances	53	8768.1165	4507.09
2009-12-31	Southridge Video	Southridge Video	Computers Accessories	Computers	195	6645.946	4111.67
2009-12-31	Contoso, Ltd	Contoso	Cameras & Camcorders Access	Cameras and camcorders	596	19390.808	11862
2009-12-31	Litware, Inc.	Litware	Fans	Home Appliances	595	21169.612	12467.01
2009-12-31	Adventure Works	Adventure Works	Desktops	Computers	16	8563.2	4073.44
2009-12-31	Northwind Traders	Northwind Traders	Washers & Dryers	Home Appliances	51	45913.725	27619.98
2009-12-31	Contoso, Ltd	Contoso	Air Conditioners	Home Appliances	127	24141.5485	13691.85
2009-12-31	Tailspin Toys	Tailspin Toys	Boxed Games	Games and Toys	715	4549.33	2437.35
2009-12-31	Contoso, Ltd	Contoso	Water Heaters	Home Appliances	72	57020.7	29041.92
2009-12-31	Proseware, Inc.	Proseware	Projectors & Screens	Computers	277	131570.6	65197.52
2009-12-31	Adventure Works	Adventure Works	Lamps	Home Appliances	136	24629.591	12301.37
2009-12-31	Litware, Inc.	Litware	Refrigerators	Home Appliances	116	36205.033	19208.4
2009-12-31	Litware, Inc.	Litware	Washers & Dryers	Home Appliances	13	24995.1	12517.3
2009-12-31	Wide World Importers	Wide World Importers	Bluetooth Headphones	Audio	56	5874.949	3140.98

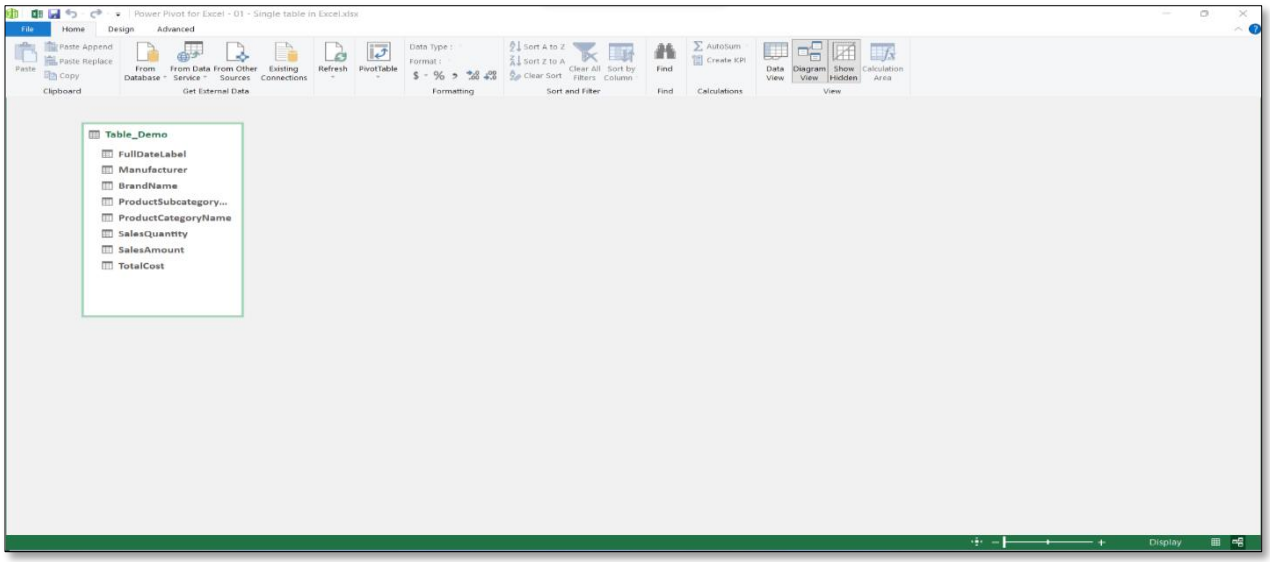
2. Select the ‘Close and load To’ Under the ‘Home’ tab to load it in the model.



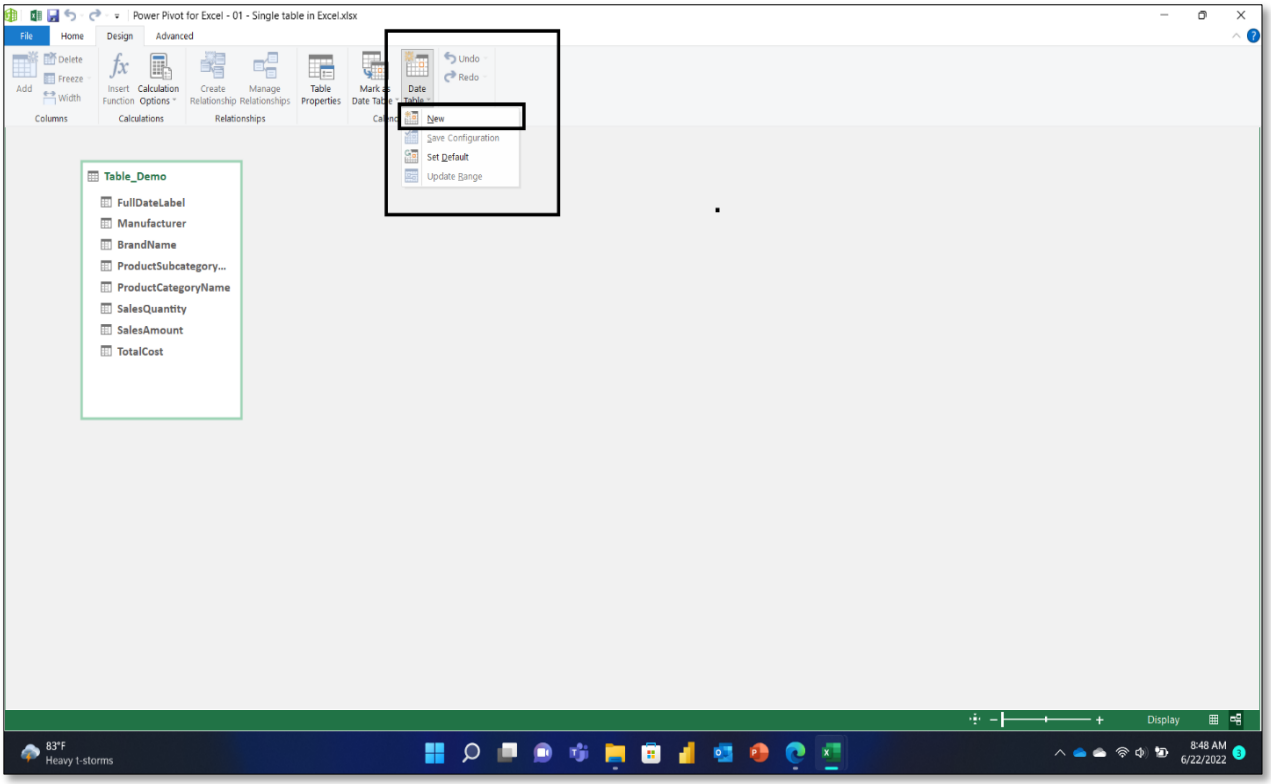
3. We can load it as a Table or Only Create a Connection. Make Sure to check ‘Add This data to Data Model’ and then load.
4. After loading the Queries select ‘Manage Data Model’ to open Power Pivot.


01 - Single table in Excel - Excel
Table Tools
File Home Insert Page Layout Formulas Data Review View Help Power Pivot Table Design
Tell me what you want to do
Get External Data
Get & Transform
Connections
Sort & Filter
Data Tools
Forecast
Table_De...
2009-12-31
Load To
Select how you want to view this data in your workbook.
Table
Only Create Connection
Select where the data should be loaded.
New worksheet
Existing worksheet:
$B$4:$I$63987
Add this data to the Data Model
Load
Cancel
Transaction Table
Ready
Accessibility: Investigate
Loading .Net Framework...
Average: 21229.1874
Count: 511872
Sum: 4074984980

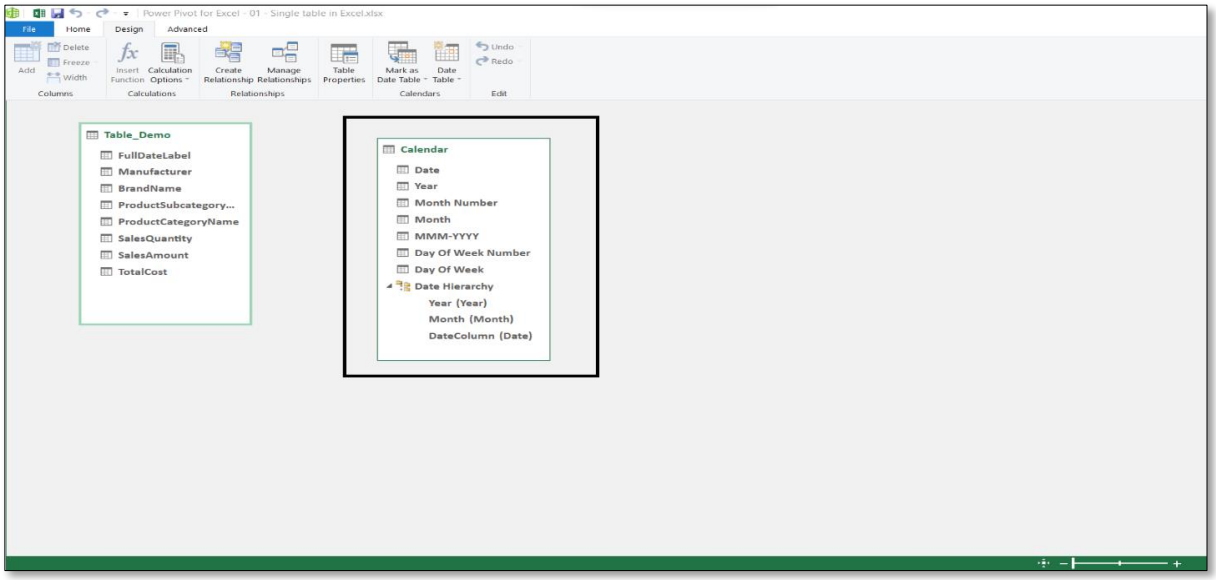

Workbook Queries
1 query
Table_Demo
63,984 rows loaded.
Transaction Table
Ready
Accessibility: Investigate
Average: 21229.1874
Count: 511872
Sum: 4074984980

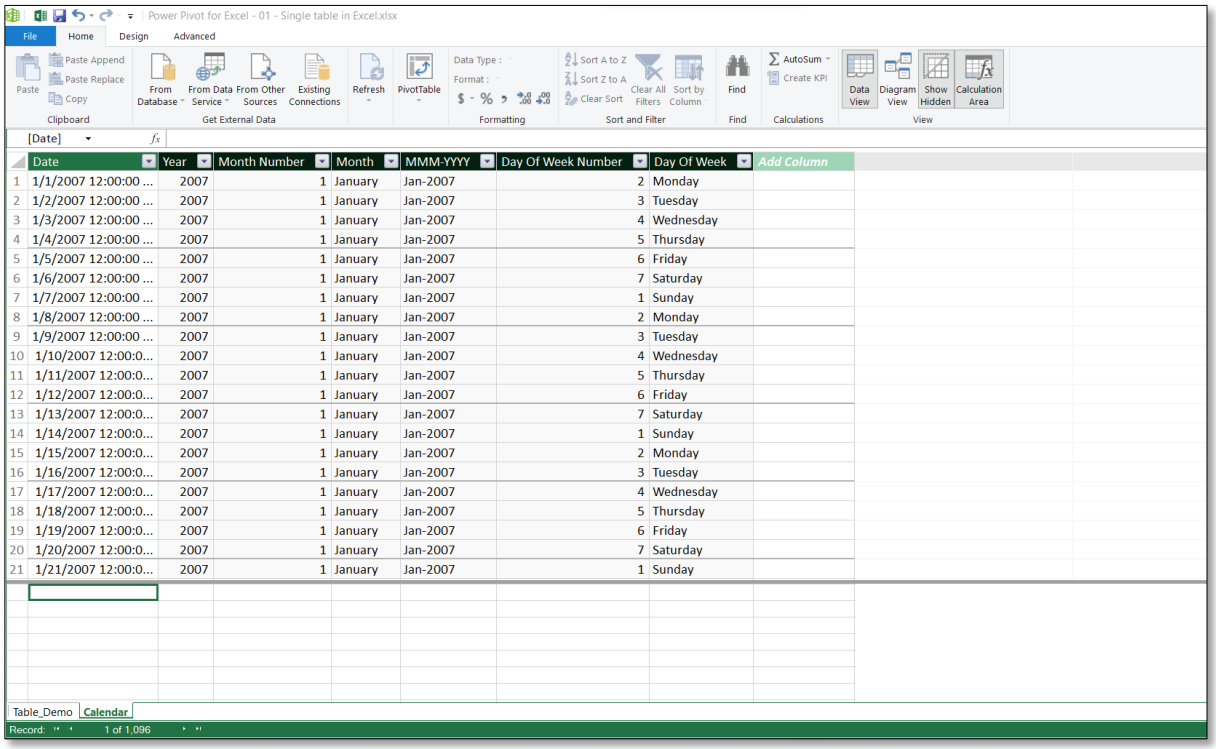


5. In power Pivot from Design Tab Select ‘New’ under ‘Date Table’ tab



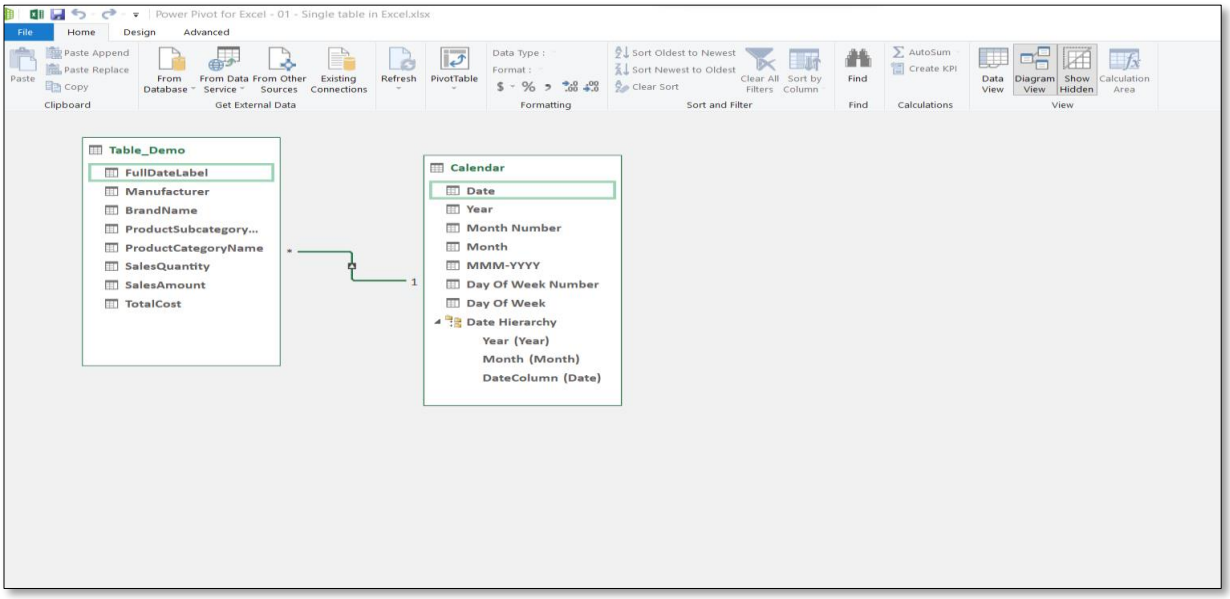
6. Power Pivot will automatically generate the calendar table with some additional columns. We can also customize the columns here.



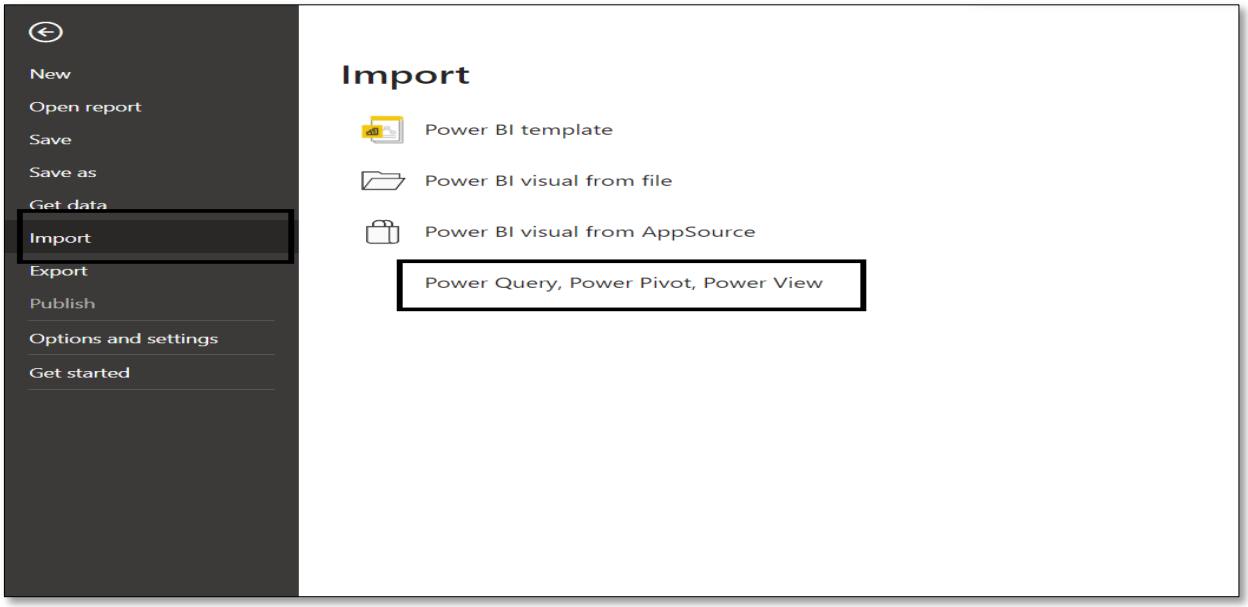


	Date	Year	Month Number	Month	MMM-YYYY	Day Of Week Number	Day Of Week	Add Column
1	1/1/2007 12:00:00 ...	2007	1	January	Jan-2007	2	Monday	
2	1/2/2007 12:00:00 ...	2007	1	January	Jan-2007	3	Tuesday	
3	1/3/2007 12:00:00 ...	2007	1	January	Jan-2007	4	Wednesday	
4	1/4/2007 12:00:00 ...	2007	1	January	Jan-2007	5	Thursday	
5	1/5/2007 12:00:00 ...	2007	1	January	Jan-2007	6	Friday	
6	1/6/2007 12:00:00 ...	2007	1	January	Jan-2007	7	Saturday	
7	1/7/2007 12:00:00 ...	2007	1	January	Jan-2007	1	Sunday	
8	1/8/2007 12:00:00 ...	2007	1	January	Jan-2007	2	Monday	
9	1/9/2007 12:00:00 ...	2007	1	January	Jan-2007	3	Tuesday	
10	1/10/2007 12:00:0...	2007	1	January	Jan-2007	4	Wednesday	
11	1/11/2007 12:00:0...	2007	1	January	Jan-2007	5	Thursday	
12	1/12/2007 12:00:0...	2007	1	January	Jan-2007	6	Friday	
13	1/13/2007 12:00:0...	2007	1	January	Jan-2007	7	Saturday	
14	1/14/2007 12:00:0...	2007	1	January	Jan-2007	1	Sunday	
15	1/15/2007 12:00:0...	2007	1	January	Jan-2007	2	Monday	
16	1/16/2007 12:00:0...	2007	1	January	Jan-2007	3	Tuesday	
17	1/17/2007 12:00:0...	2007	1	January	Jan-2007	4	Wednesday	
18	1/18/2007 12:00:0...	2007	1	January	Jan-2007	5	Thursday	
19	1/19/2007 12:00:0...	2007	1	January	Jan-2007	6	Friday	
20	1/20/2007 12:00:0...	2007	1	January	Jan-2007	7	Saturday	
21	1/21/2007 12:00:0...	2007	1	January	Jan-2007	1	Sunday	

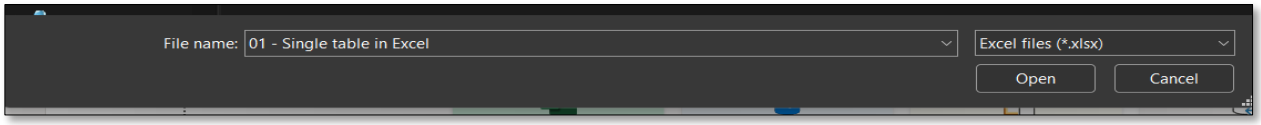
7. After that, we make a relationship between the two tables using the date column. Just Drag one column and drop it onto the Date column from the Calendar table.



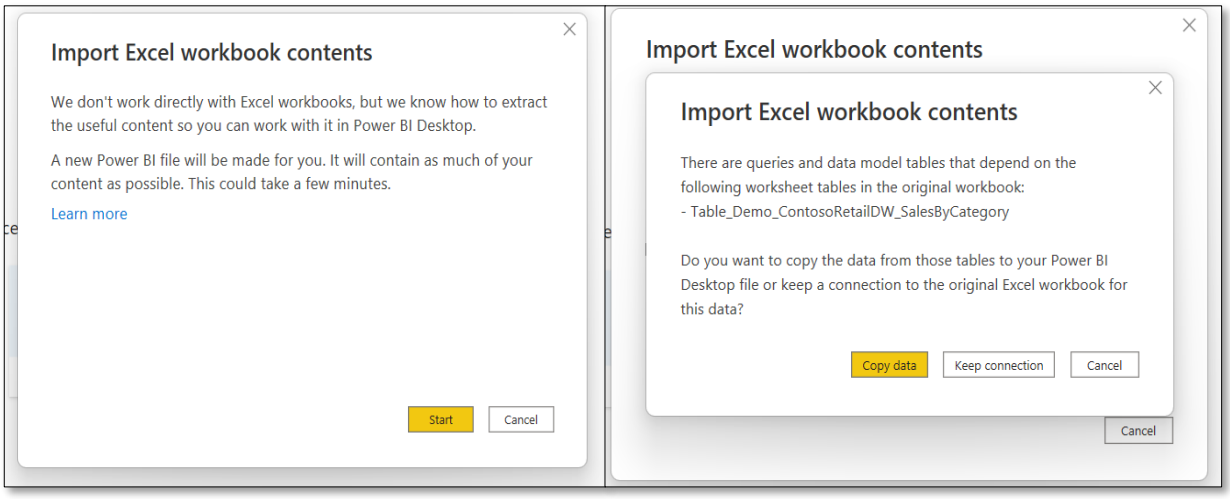
8. In order to import the model into power Bi first save and close the Excel file and then open Power Bi. From Import select 'Power Query, Power Pivot, Power View' option and select the file.

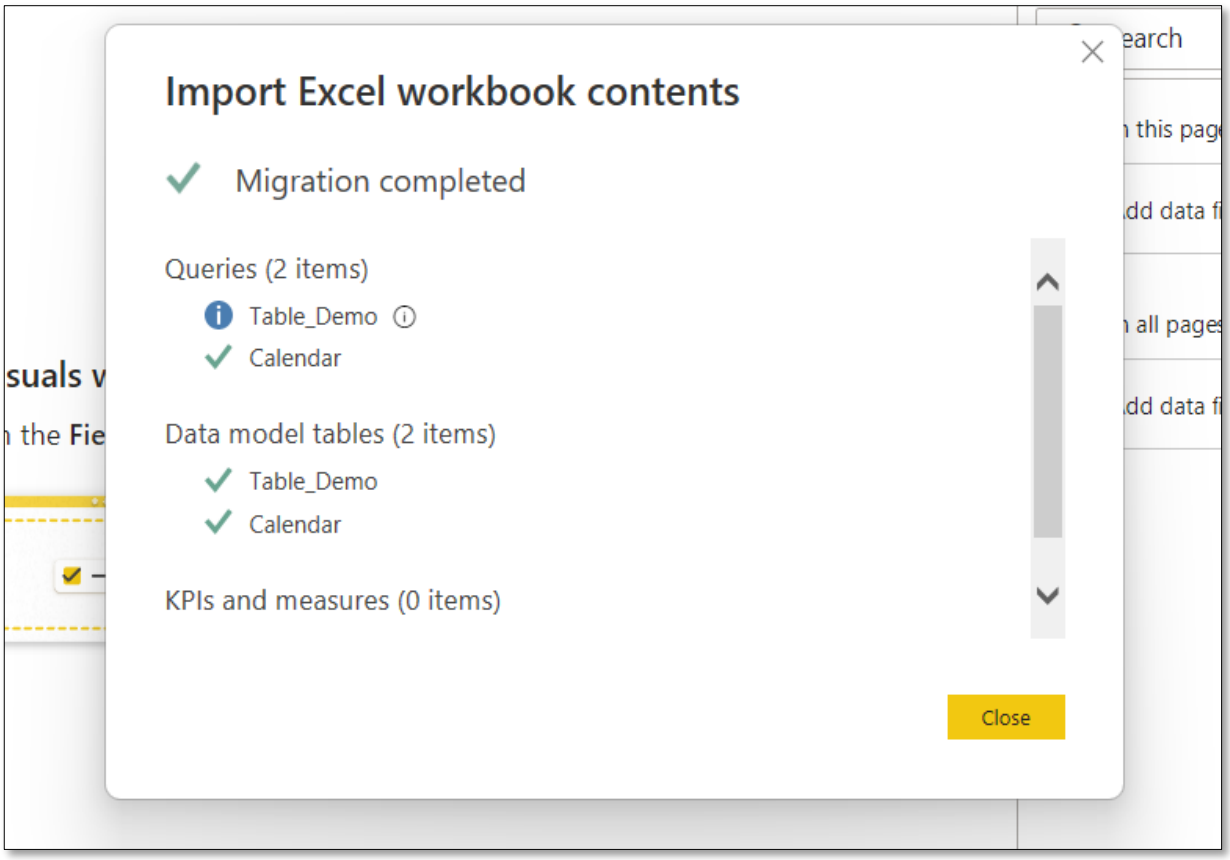


Select the File

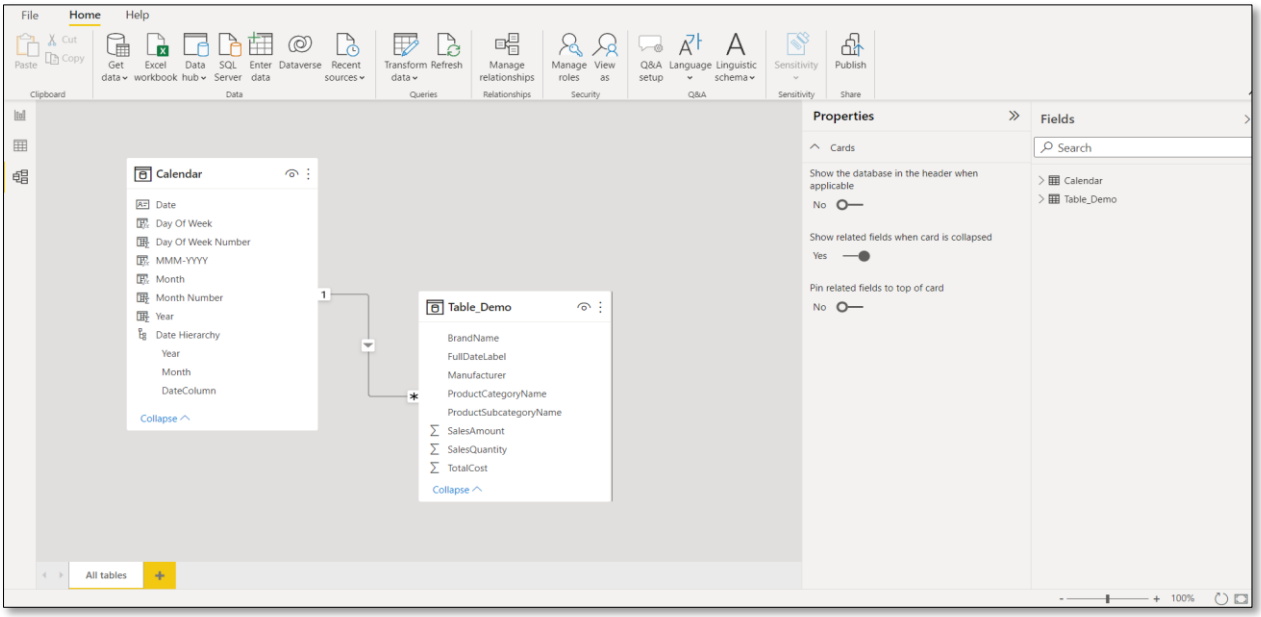


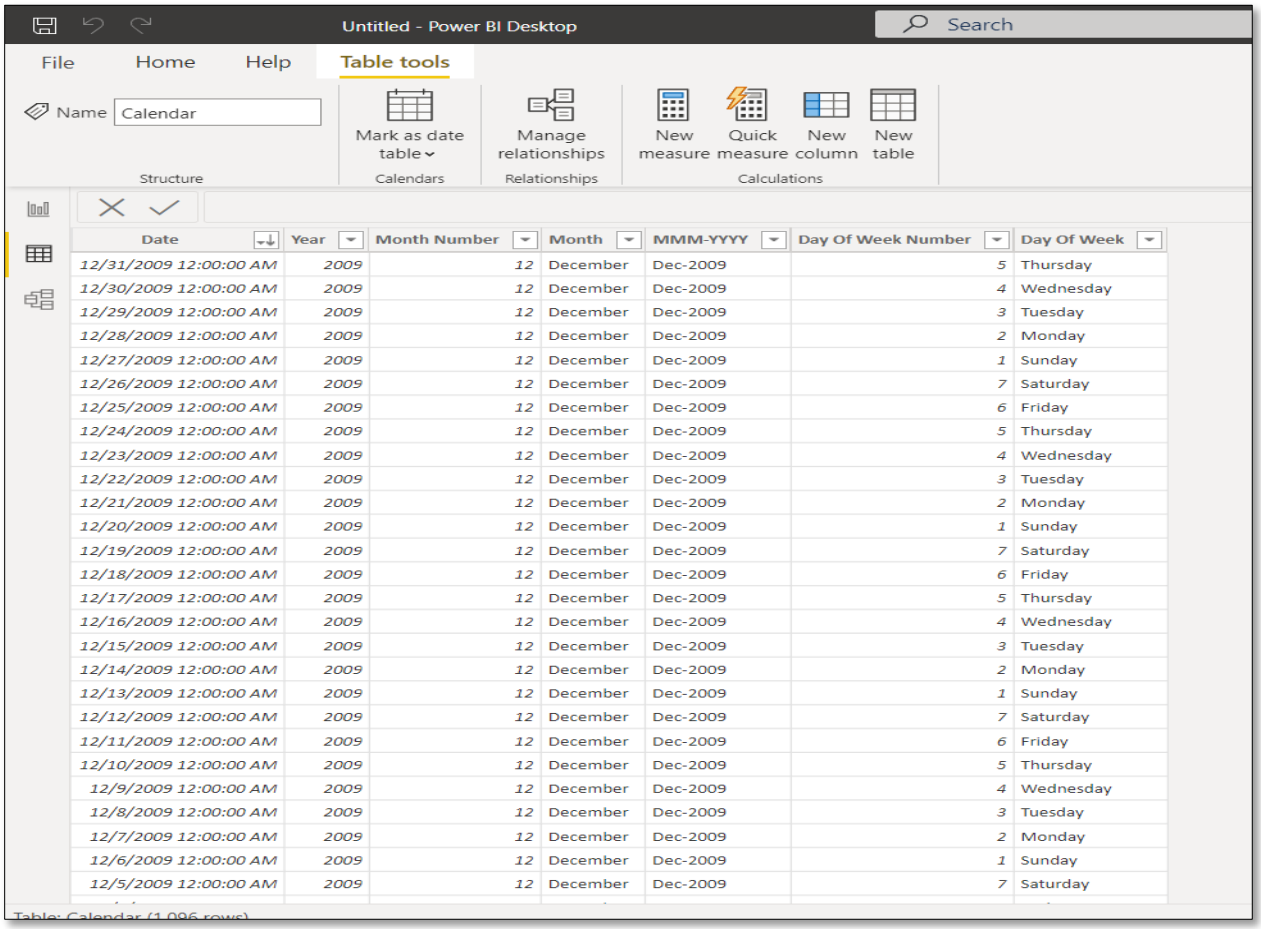
9. Power Bi will automatically load the model in it. You can copy the whole data or you can keep the only connection.





Finally, we get a Dimension Calendar Table which is created in Excel and loaded into Power Bi.





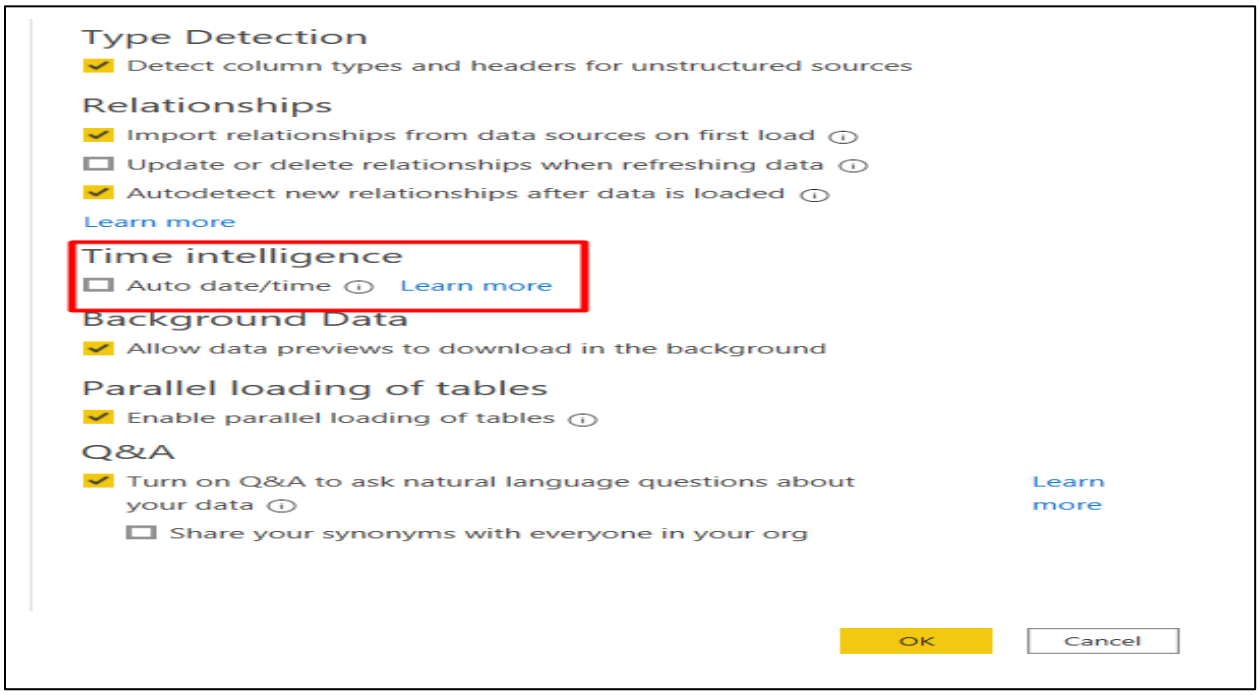
Date	Year	Month Number	Month	MMM-YYYY	Day Of Week Number	Day Of Week
12/31/2009 12:00:00 AM	2009	12	December	Dec-2009	5	Thursday
12/30/2009 12:00:00 AM	2009	12	December	Dec-2009	4	Wednesday
12/29/2009 12:00:00 AM	2009	12	December	Dec-2009	3	Tuesday
12/28/2009 12:00:00 AM	2009	12	December	Dec-2009	2	Monday
12/27/2009 12:00:00 AM	2009	12	December	Dec-2009	1	Sunday
12/26/2009 12:00:00 AM	2009	12	December	Dec-2009	7	Saturday
12/25/2009 12:00:00 AM	2009	12	December	Dec-2009	6	Friday
12/24/2009 12:00:00 AM	2009	12	December	Dec-2009	5	Thursday
12/23/2009 12:00:00 AM	2009	12	December	Dec-2009	4	Wednesday
12/22/2009 12:00:00 AM	2009	12	December	Dec-2009	3	Tuesday
12/21/2009 12:00:00 AM	2009	12	December	Dec-2009	2	Monday
12/20/2009 12:00:00 AM	2009	12	December	Dec-2009	1	Sunday
12/19/2009 12:00:00 AM	2009	12	December	Dec-2009	7	Saturday
12/18/2009 12:00:00 AM	2009	12	December	Dec-2009	6	Friday
12/17/2009 12:00:00 AM	2009	12	December	Dec-2009	5	Thursday
12/16/2009 12:00:00 AM	2009	12	December	Dec-2009	4	Wednesday
12/15/2009 12:00:00 AM	2009	12	December	Dec-2009	3	Tuesday
12/14/2009 12:00:00 AM	2009	12	December	Dec-2009	2	Monday
12/13/2009 12:00:00 AM	2009	12	December	Dec-2009	1	Sunday
12/12/2009 12:00:00 AM	2009	12	December	Dec-2009	7	Saturday
12/11/2009 12:00:00 AM	2009	12	December	Dec-2009	6	Friday
12/10/2009 12:00:00 AM	2009	12	December	Dec-2009	5	Thursday
12/9/2009 12:00:00 AM	2009	12	December	Dec-2009	4	Wednesday
12/8/2009 12:00:00 AM	2009	12	December	Dec-2009	3	Tuesday
12/7/2009 12:00:00 AM	2009	12	December	Dec-2009	2	Monday
12/6/2009 12:00:00 AM	2009	12	December	Dec-2009	1	Sunday
12/5/2009 12:00:00 AM	2009	12	December	Dec-2009	7	Saturday

Table: Calendar (1,096 rows)

Create Calendar Table Using BRAVO

Bravo for Power BI is a powerful toolkit that helps to analyze your models, format measures, create date tables, and export data.

To manage the date table, first, the Auto date/time option under the Time intelligence must be disabled.



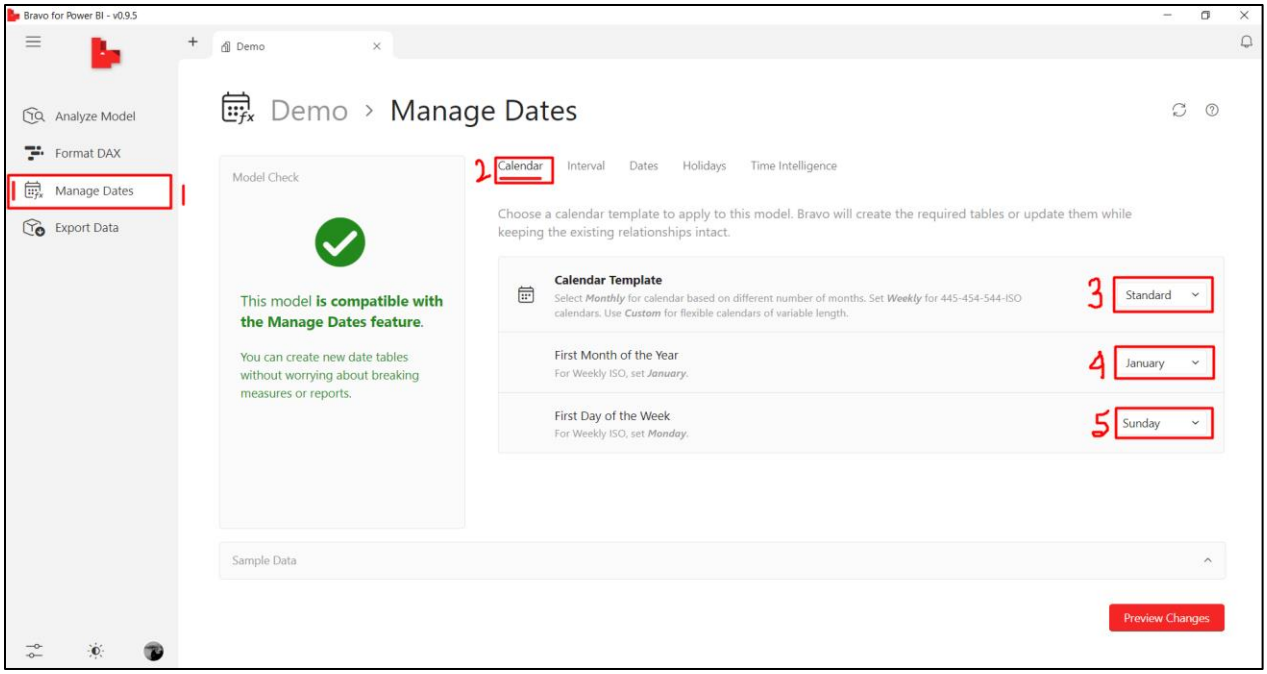
Note: To disable the Auto date/time option under the Time intelligence (go to Power BI Desktop > File > Options and Settings > Options > Data Load > Time intelligence > Auto date/time for new Files)

Go to the Manage Date ribbon and pick the date table. After picking the date table follow the steps below.

Step 1:

Download BRAVO from Bravo for Power BI from GitHub After installing BRAVO, open Power BI Desktop then under External Tool Tab Select Bravo. Or connect to Power BI Desktop from BRAVO and select the dataset you want to work with. Then follow the steps below.

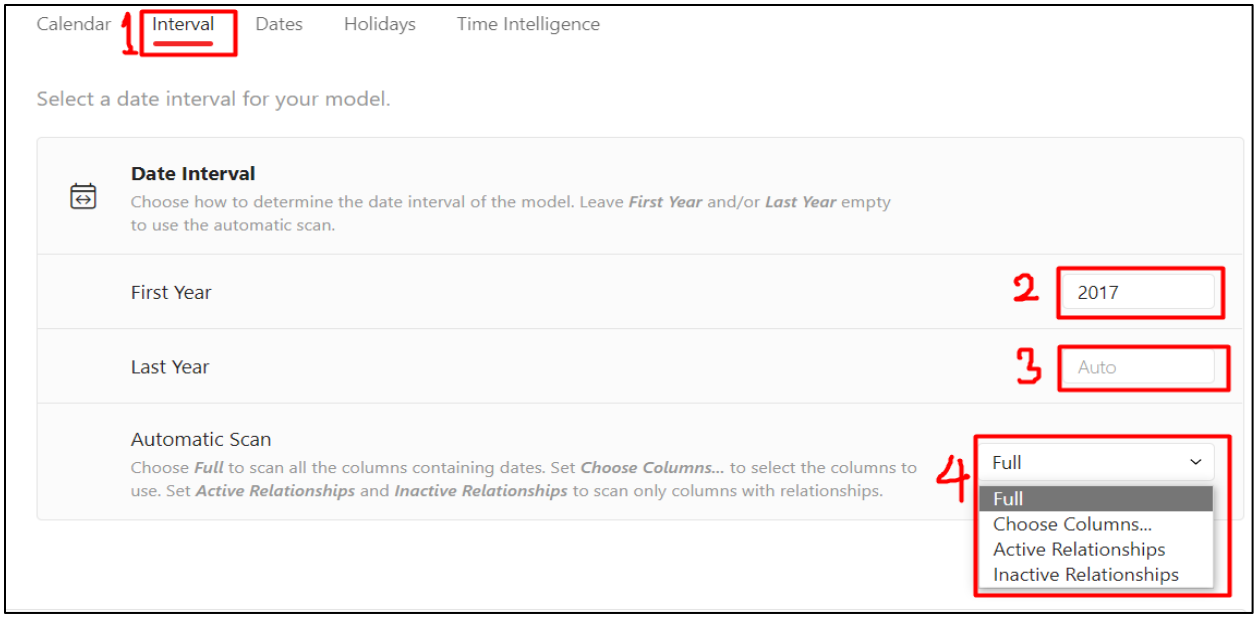
- Choose Calendar under Manage Date (Marked as 1). As the date table is already selected, this feature will show the overall view for Calendar (Marked as 2) setting.



- Choose Standard (Marked as 3) for Standard type calendar, Monthly or Weekly for monthly or weekly base calendar or select Custom for a custom calendar
- The same way applies for selecting the First Month of the Year (Marked as 4) and First Day of the Week (Marked as 5).

Step 2:

This "Interval" ribbon (Marked as 1) helps to choose date column (Automatically & Manually) as marked 4 with additional options where First Year (Marked as 2) & Last Year (Marked as 3) can be defined.

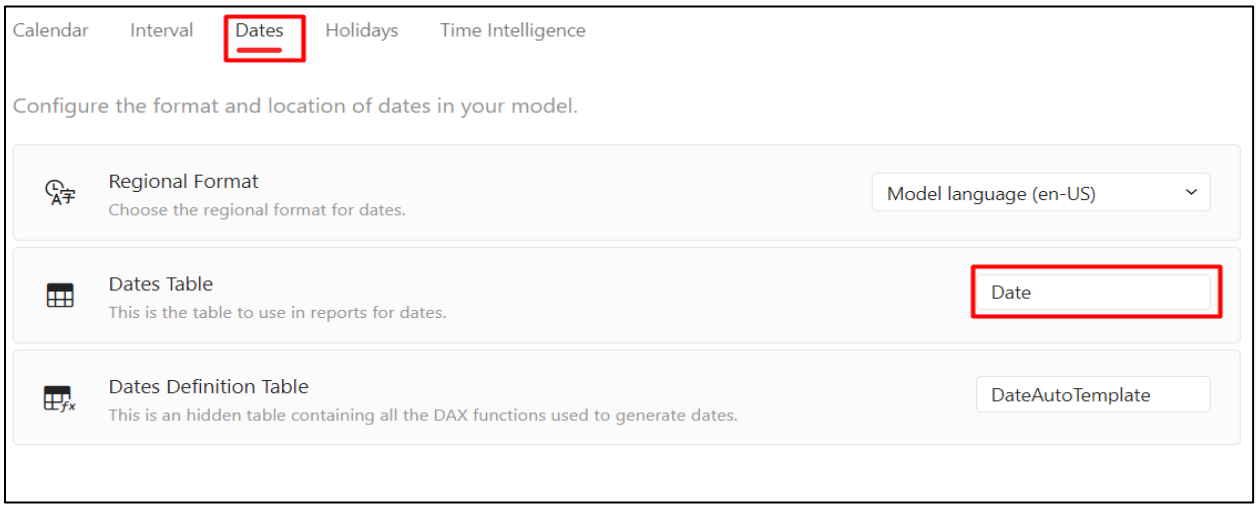


This will create a calendar from January 2017 to December 2021. Select Full in Automatic Scan to scan entire dataset or Select Choose Columns to scan a specific column.

Note: Left last year empty to create a calendar till the current date.

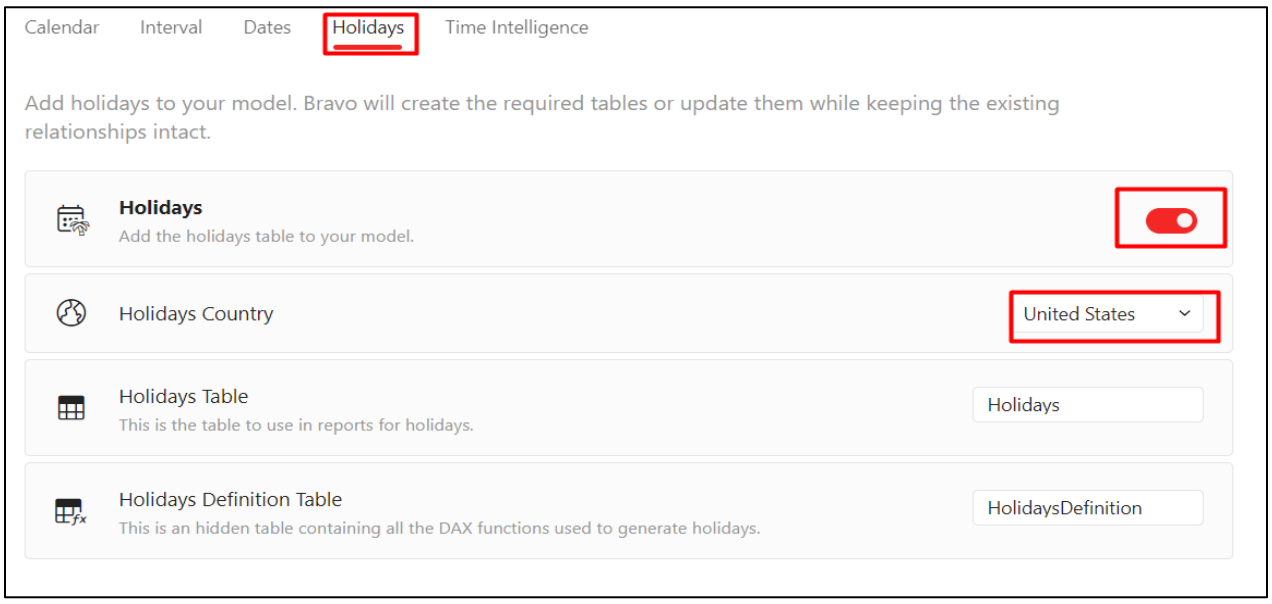
Step 3:

Regional format, table name along with date definition table will be found in the Dates setting. This helps to choose the regional format for the date table. Change the Regional Format field if you want a specific format for the table. To rename the table, edit the Dates Table field.



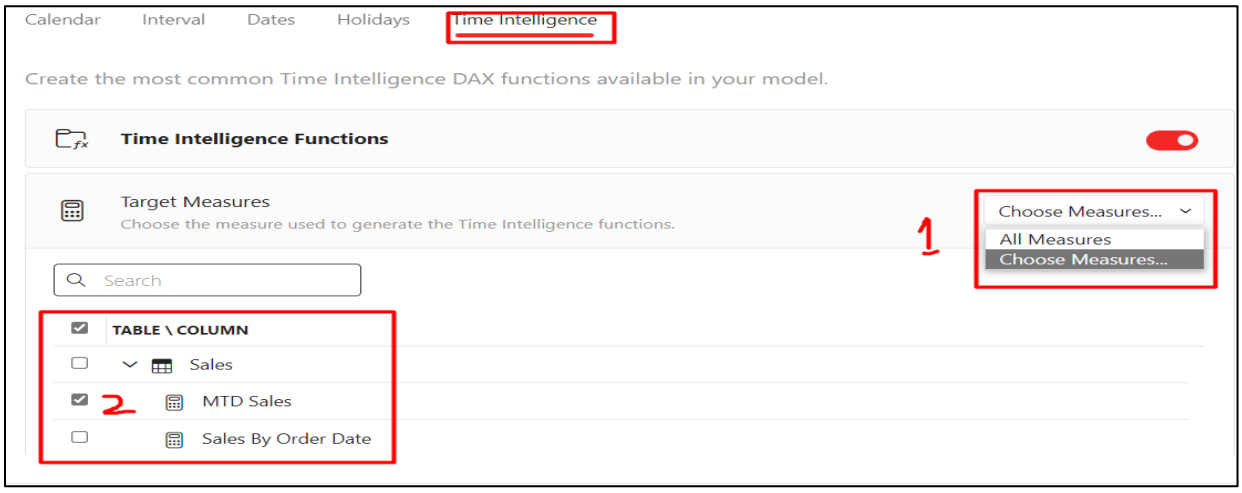
Step 4:

Holidays assist to include holidays in the calendar table, to activate this functionality On, set the Holidays on, Country wise holidays can be chosen from here. Keep the Holidays field on. Change the country in Holidays Country field if you want holidays for a specific country.



Step 5:

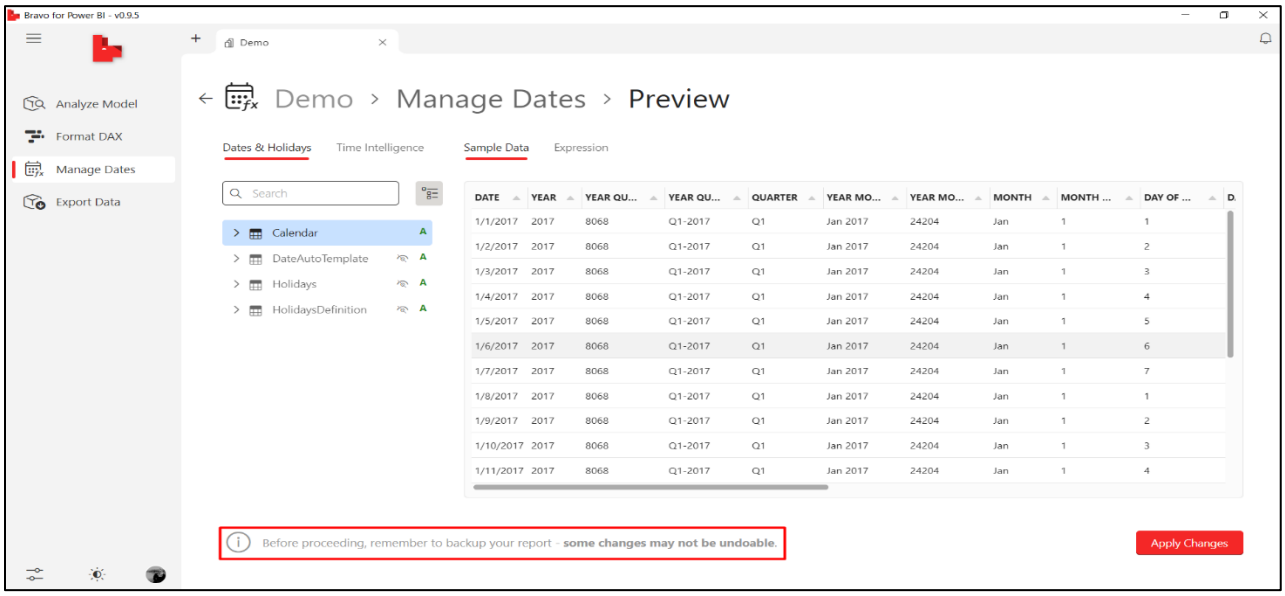
By enabling Time Intelligence functions on, we can utilize the time intelligence to our Calendar table. Another feature is available where we can initiate the time intelligence based on All Measures or Chosen Measures (Marked as 1). By Choosing a specific measure, BRAVO will create some measures (Marked as 2) to perform Time Intelligence.



Note: All the measures will be based on the selected measure.

Step 6:

After everything's done, Click Preview. Check the preview if everything's ok, Click Apply Changes.

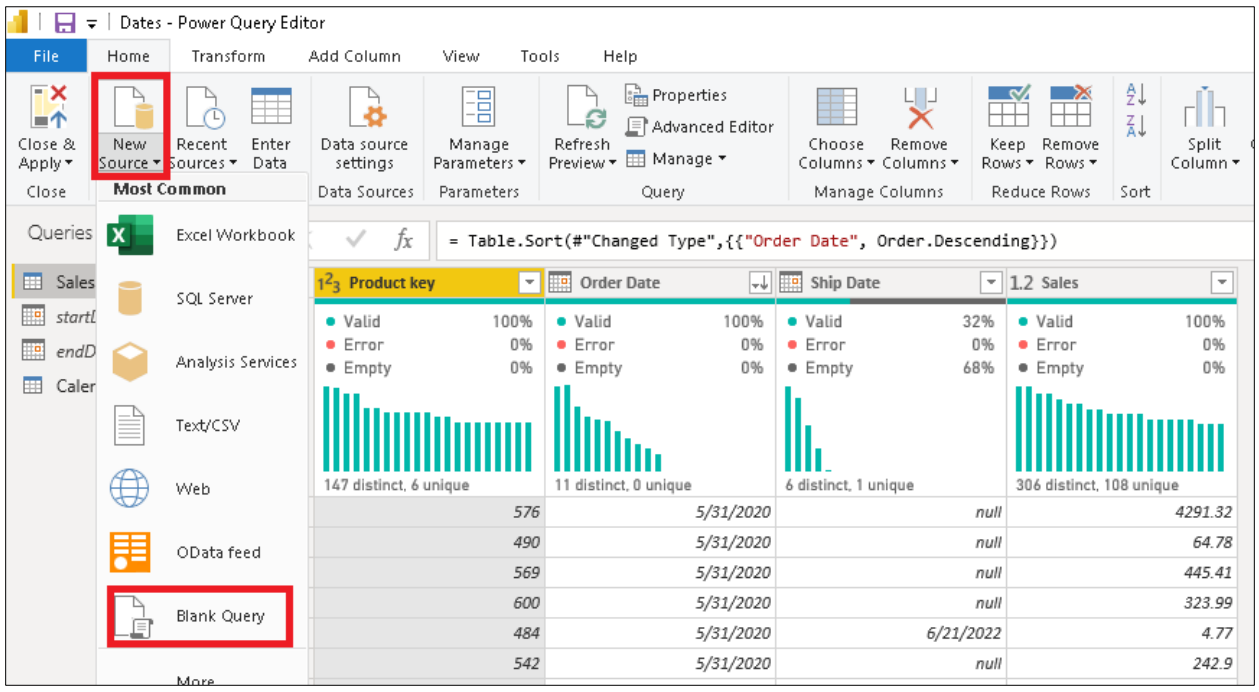


Now the Calendar table can be used to perform different operations in Power BI.

Create Calendar Table with Power Query

Step 1 – Load Data:
Open power Bi desktop application. Select a dataset from get data; and click transform data.

Step 2 – Pick Start Date with M Language:
From Home ribbon, navigate to New Source and Select blank Query.



Follow along the code below and put it in the formula bar of the new blank query, and name that query startDate.

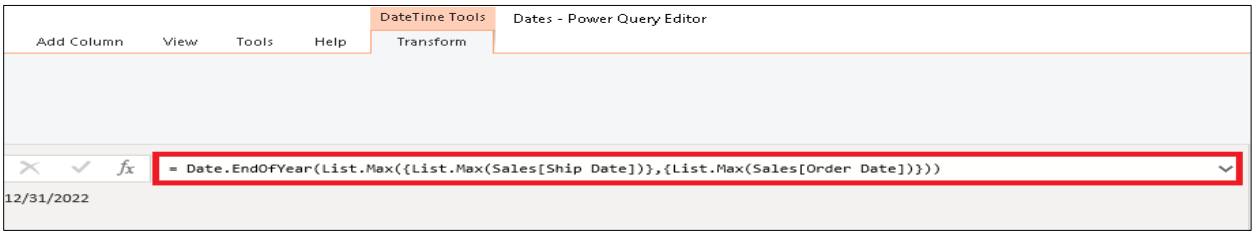
```
= Date.StartOfYear(List.Min({List.Min(Sales[Ship Date])},{List.Min(Sales[Order Date])}))
```

List.Min will select the minimum date from the Ship Date and Order Date columns. Then, Date.StartOfYear will pick the very first date of that specific year.

Here, Sales is the source table name.

Step 3- Pick End Date with M Language:

From Home ribbon drop down New Source and Select blank Query.



Write the code given below in the editor pane and name that query endDate.

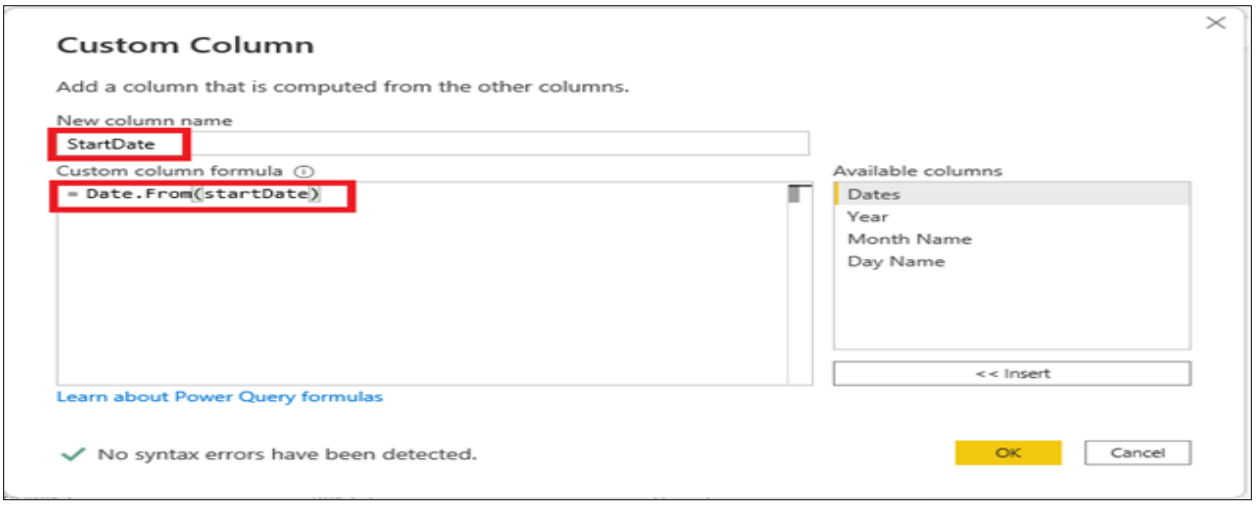
```
= Date.EndOfYear(List.Max({List.Max(Sales[Ship Date])},{List.Max(Sales[Order Date])}))
```

This code will select the maximum date from “Ship Date” and “Order Date” columns, and pick the last date of that specific year.

Step 4- Create StartDate and EndDate Columns:

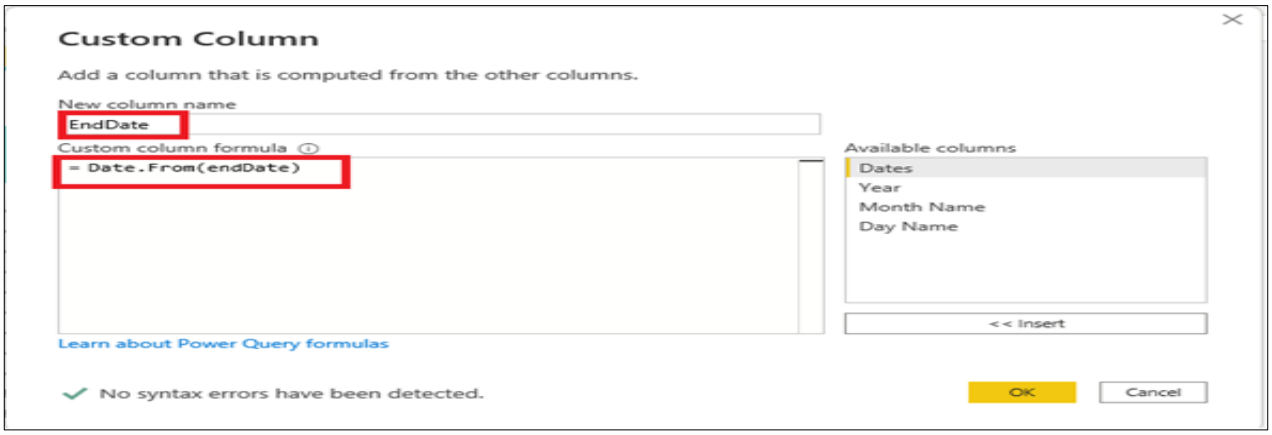
First, duplicate the Sales table.

Select custom column, name that column StartDate. In the formula pane write =Date.From(startDate)

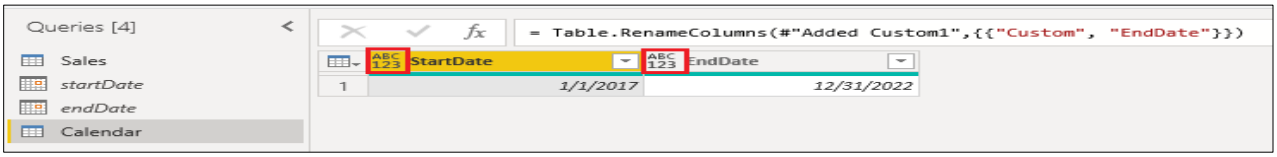


Then click ok. Now, keep only StartDate column, and delete rest of the columns. We can do it from Choose column option from home tab.

Right click on the column header and select remove duplicate. Again, Select custom column, name that column EndDate. In the formula pane write =Date.From(endDate) and then click ok.

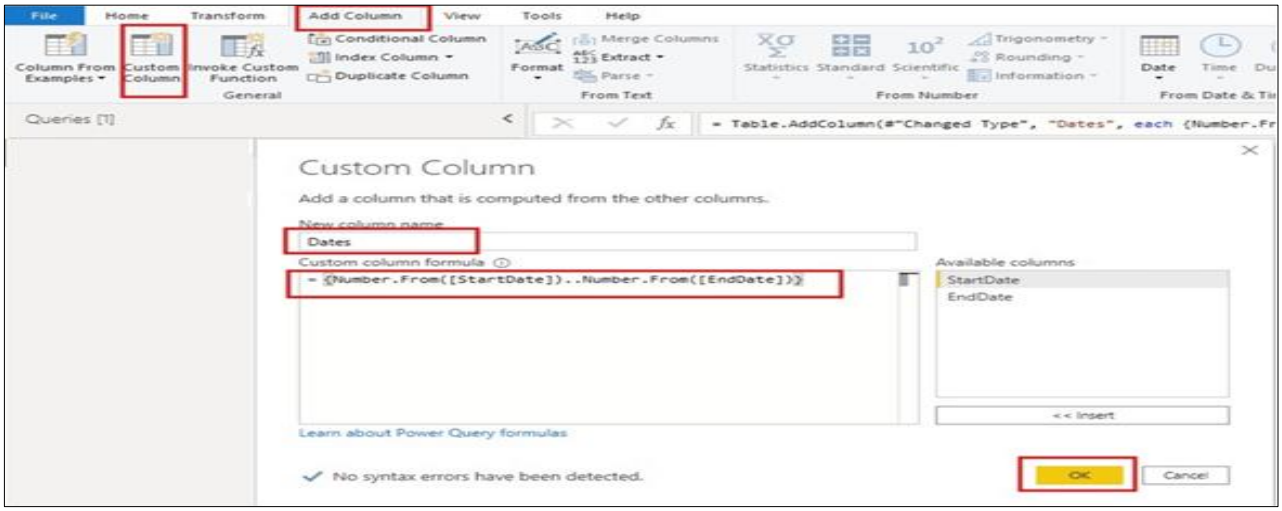


Once the M code above is entered correctly you should now be able to see the two columns "**StartDate**" and "**EndDate**". Please remember to change the datatypes of both columns to **Date** datatypes as seen in the screenshot below.

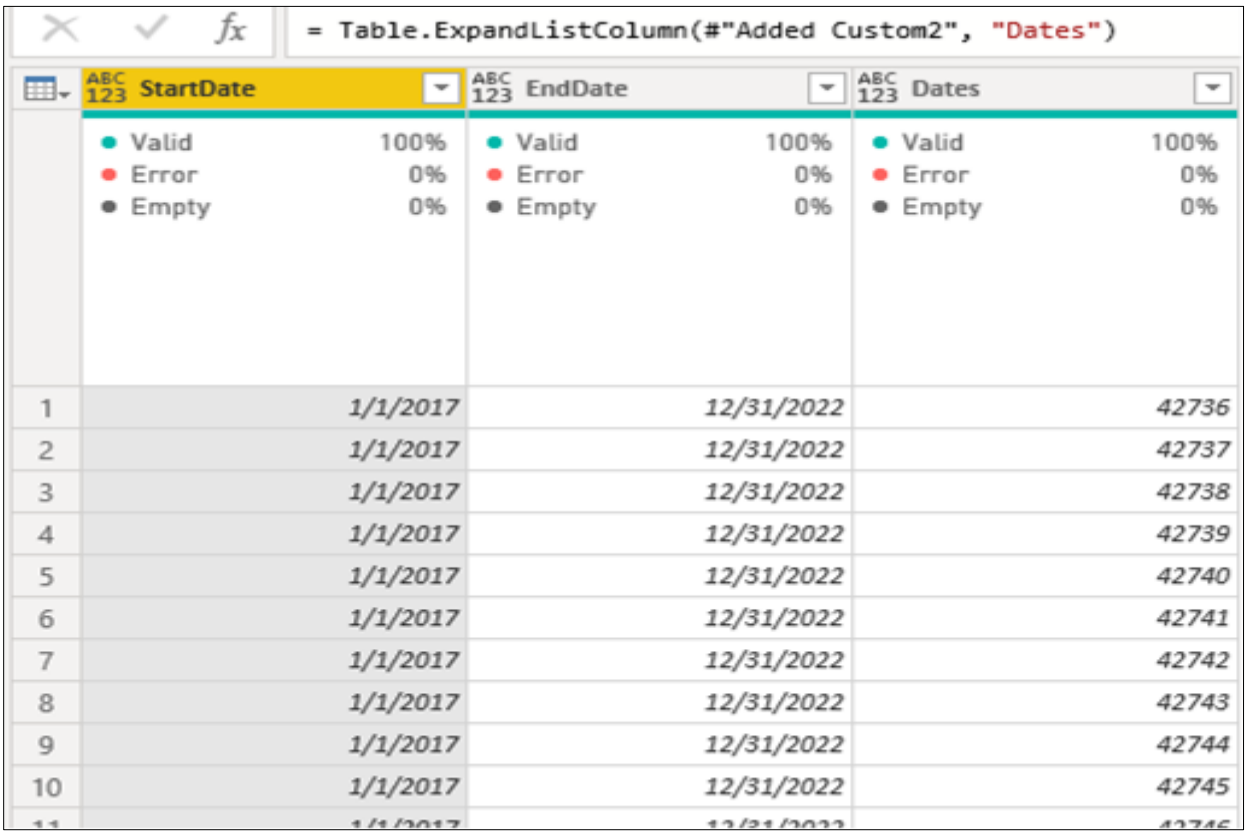


Step 5- Creating Date Column:

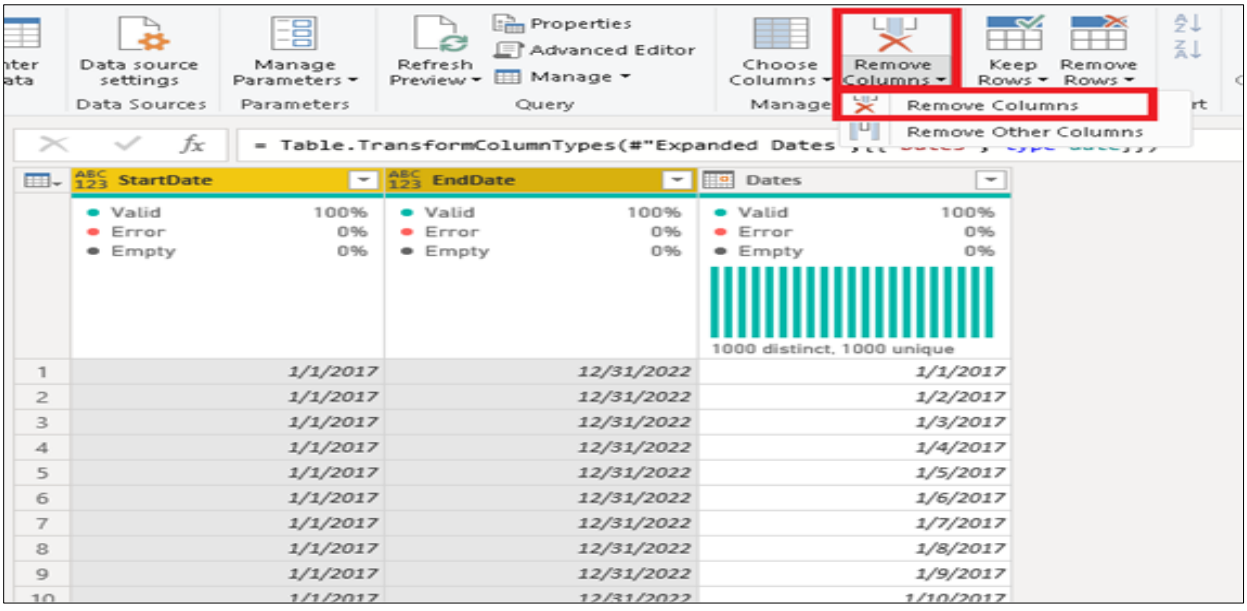
Select custom column, name that column Dates. In the formula pane write ={Number.From([startDate])..Number.From([endDate])}.



Once the Dates column has been expanded, you should be able to see the date range lists, but in numbers, since we converted the dates to numbers using "**Number.From**" function earlier.



Now, change the data type of the "**Dates**" column to Date data type. Finally, remove the startDate and endDate column.



Creating Fiscal Year

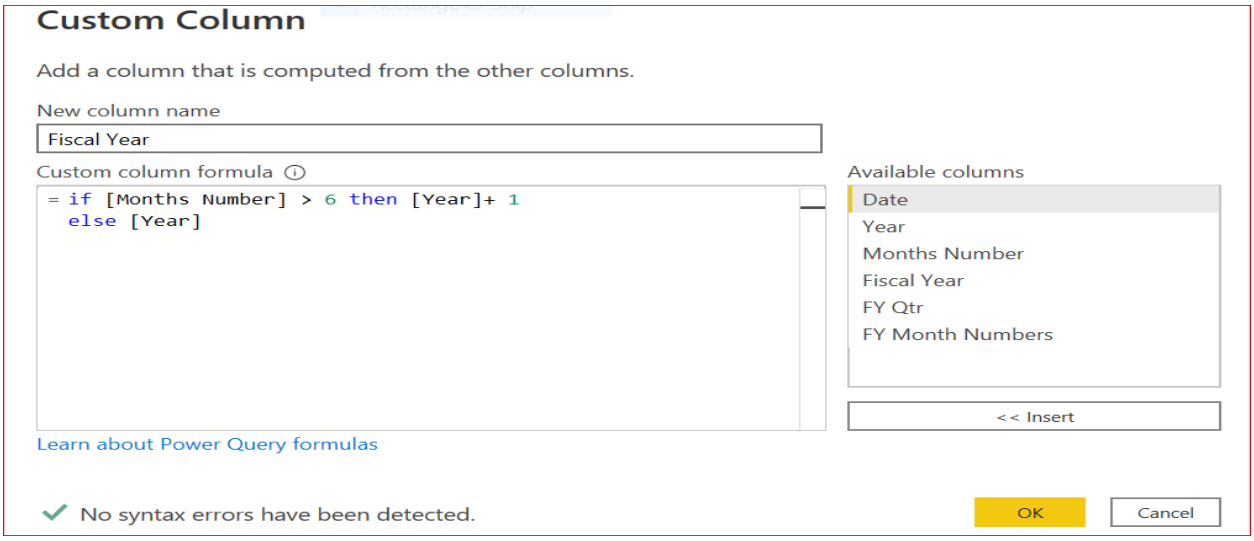
Step 1: Click the right button on the Dates column and create a duplicate column. Rename it "Year". Transform the column into "Year – Year" by clicking right button.

Step 2: Repeat the above process, however, this time rename it "Months Number" and choose "Month – Month" from the transform option.

Step 3: Now let's go to the "Add Column" tab at the top of the window, and press "Custom Column"

We'll see a new custom column window. Write "Fiscal Year" in the "New column name" box. In the formula box, let's write the formula given below.

= if [Months Number] > 6 then [Year]+ 1 else [Year]



If a fiscal year starts from July 2017, we should take it as fiscal year 2018 that will end in the month of June 2018. Hence, we add 1 with Year column.

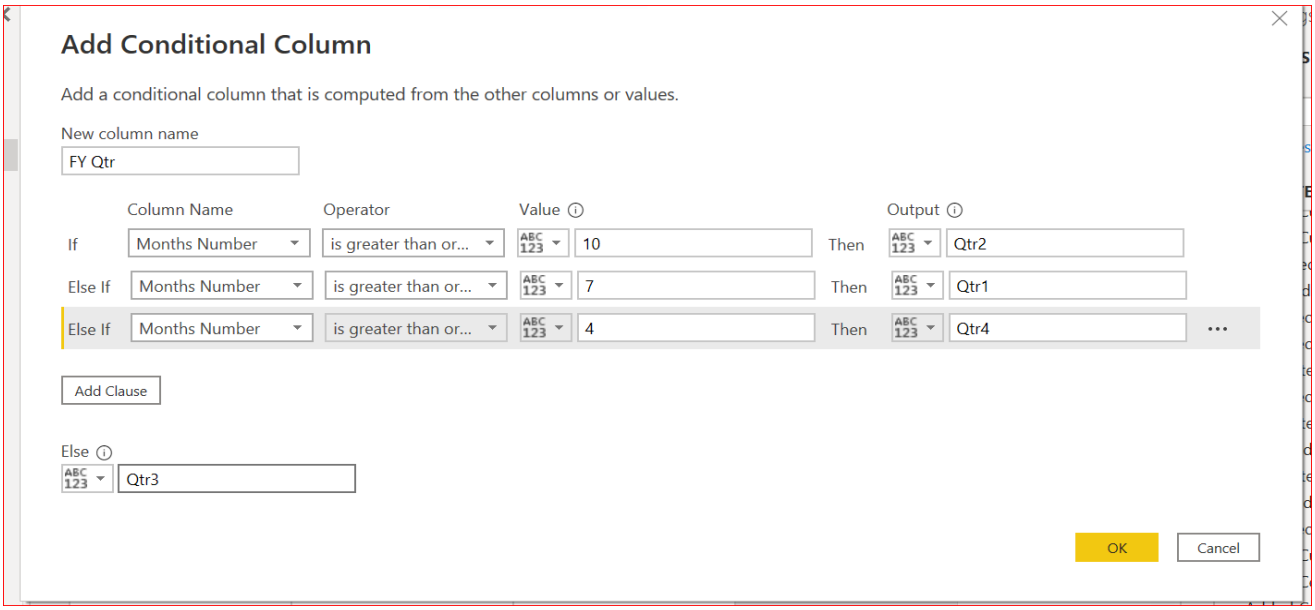
Press ok, and it will create the Fiscal year column.

Note: M language is case sensitive. It's a good idea to select the column name by double clicking them on the "Available Columns" box.

Creating Fiscal Quarters

Step 1: Choose “Conditional Column” from the “Add Colum” tab. Set the name “FY Qtr”.

Follow along exactly with the screenshot below assuming that July is the starting of the fiscal year.



Once you’re done, press ok and it will create a new column with FY Quarters.

Creating Fiscal Month Numbers

We need to create this column to sort the months column properly into Power BI visualizations.

Step 1: Choose “Custom Column” first. Set the name “FY Month Numbers”

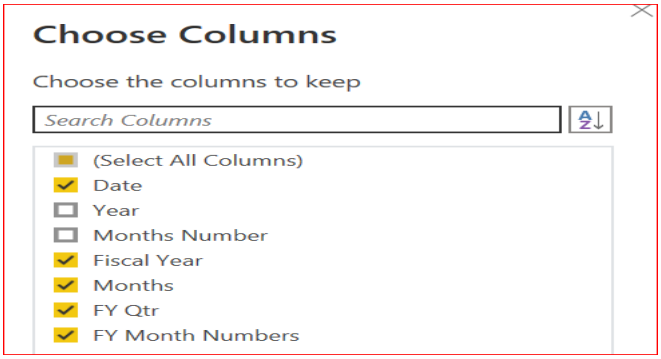
Put the syntax given below into the formula box. Press ok.

```
= if[Months Number] > 6 then [Months Number] - 6 else [Months Number] + 6
```

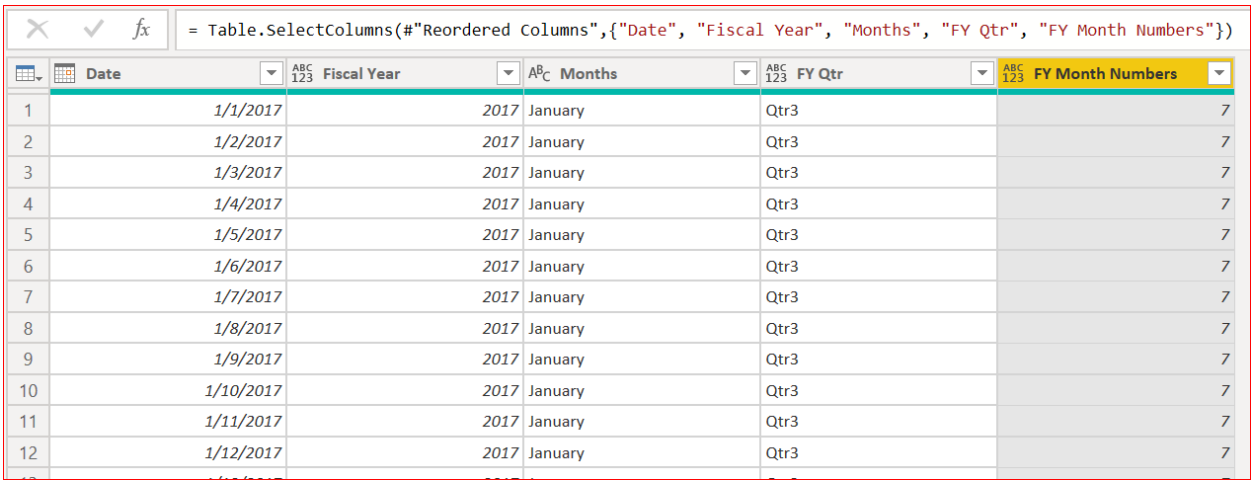
Here, we’re commanding power query to take July as the first month of the Fiscal Year.

We can create month names/week names columns also by duplicating the date column. Right click on the duplicated columns and choose transform. This will allow us to transform into months or weekdays.

Now we can delete unnecessary columns. To do so, go to the Choose Columns option from the home tab, and select Choose Column. Next, deselect Year & Months Number columns.



Now we can see our calendar table below.



	Date	Fiscal Year	Months	FY Qtr	FY Month Numbers
1	1/1/2017	2017	January	Qtr3	7
2	1/2/2017	2017	January	Qtr3	7
3	1/3/2017	2017	January	Qtr3	7
4	1/4/2017	2017	January	Qtr3	7
5	1/5/2017	2017	January	Qtr3	7
6	1/6/2017	2017	January	Qtr3	7
7	1/7/2017	2017	January	Qtr3	7
8	1/8/2017	2017	January	Qtr3	7
9	1/9/2017	2017	January	Qtr3	7
10	1/10/2017	2017	January	Qtr3	7
11	1/11/2017	2017	January	Qtr3	7
12	1/12/2017	2017	January	Qtr3	7