

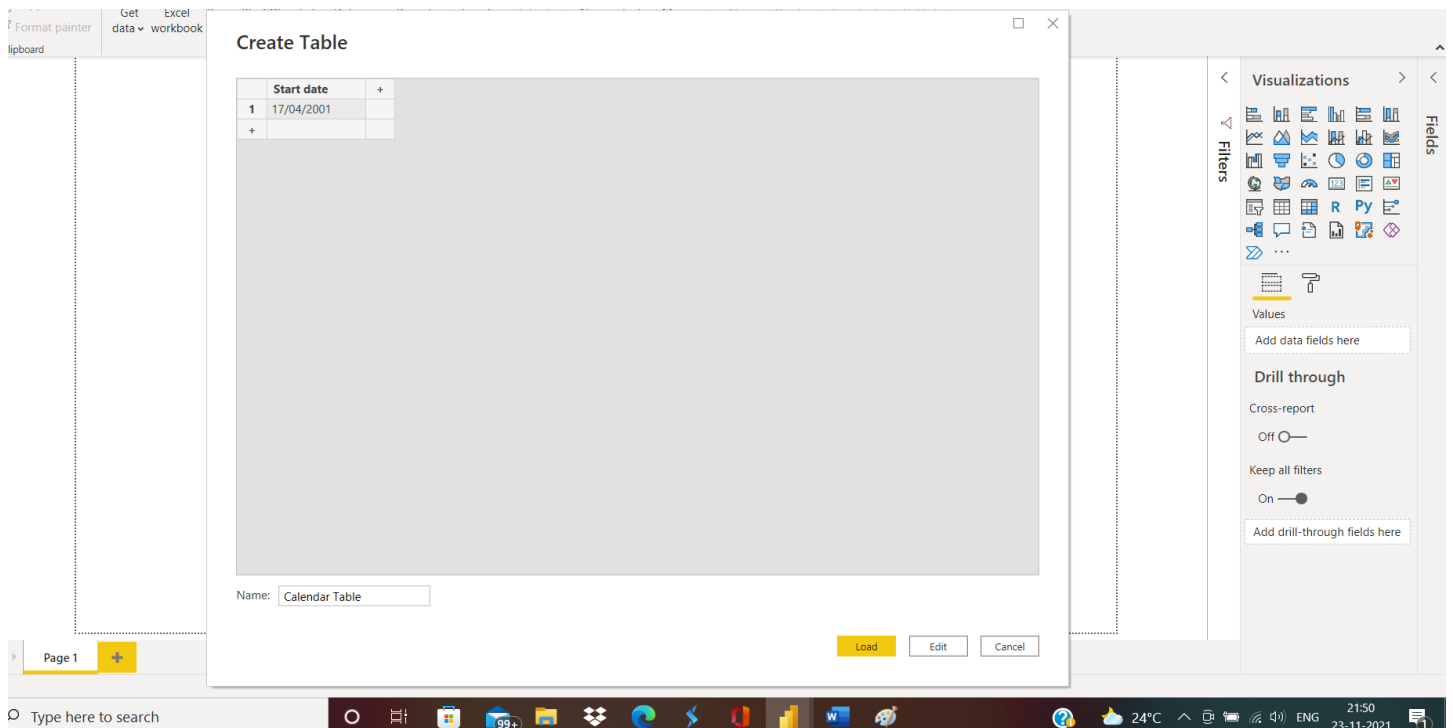
# Creating a Calendar Table Using the Power Query M-Language

## Introduction

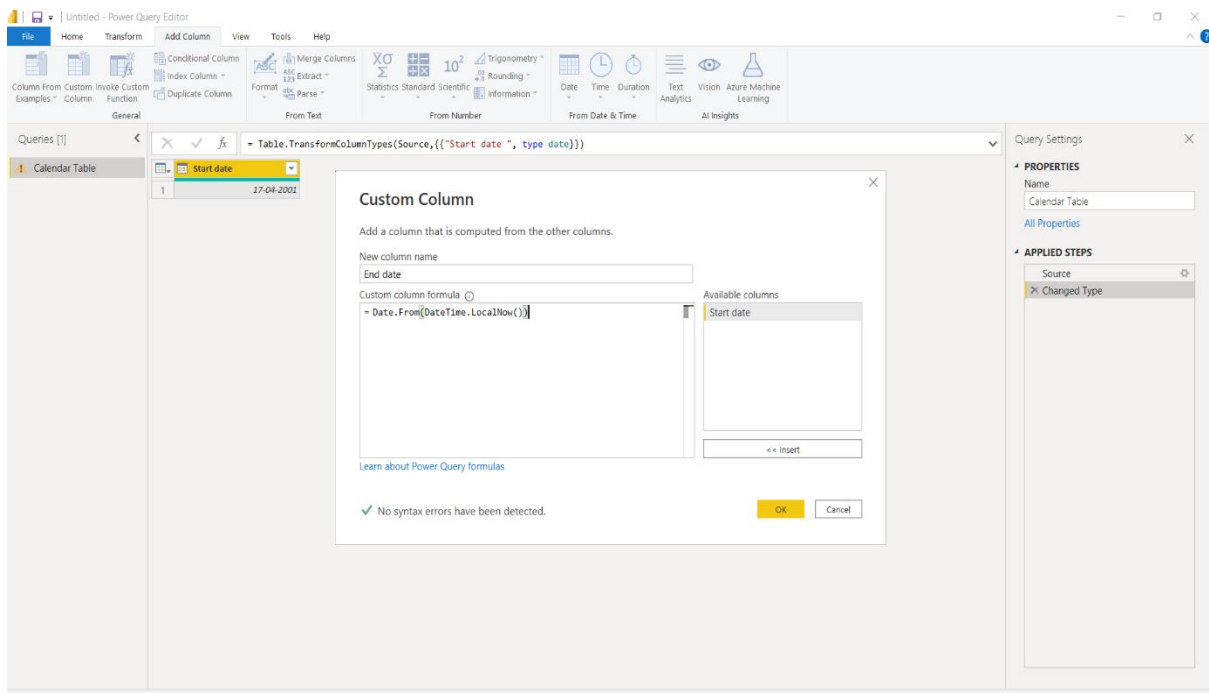
Calendar Date table can be created in Power BI using mostly DAX. However, there might be reasons one might want to do the same using Power Query M language which might be for Model Performance reasons or convenience reasons. In this i have created the calendar date table using a Insert Column and the M language in Power Query.

## Steps for creating the calendar table

**Step-1** Create a new table named "**Calendar table**" from the home tab using the enter data table.

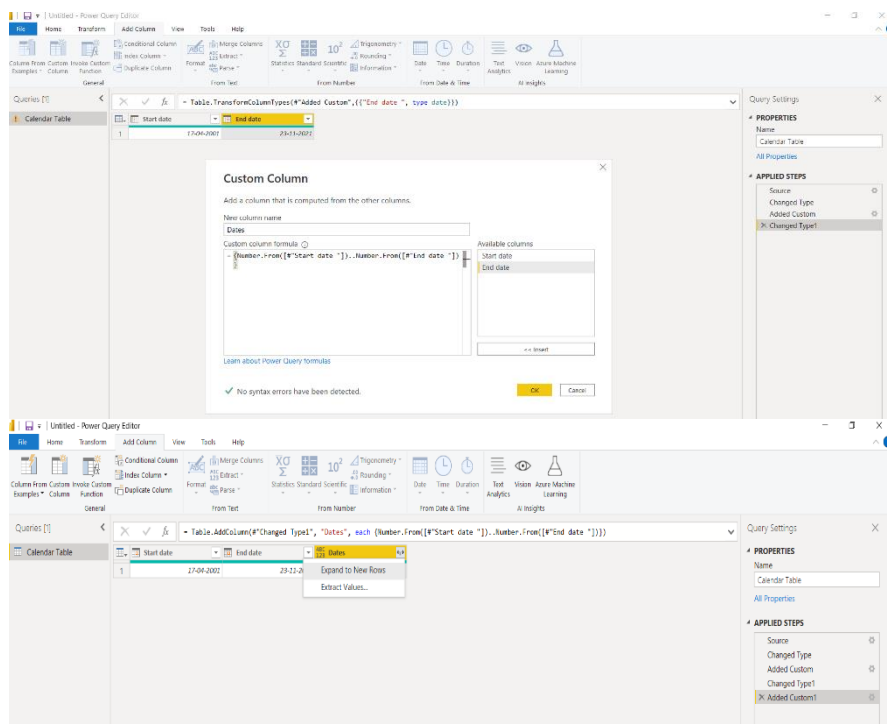


**Step-2** Select the start date query and from the add column ribbon choose custom column option and create a column named "\*\*\*End date\*\*", then in the custom column formula type - `**Date.From(DateTime.LocalNow())**` and enter ok .



**Step-3** Change the datatype of both the columns to date type.

**Step-4** Again from the add column ribbon choose custom column name as **\*\*\*Dates\*\*\*** and enter the formula as **\*\*{Number.From([#"start date"])..Number.From([#"End date"])}\*\*** click ok , expand the Dates column to new rows , change the datatype . Delete the **\*\*start date\*\*** and **\*\*\*End date\*\*\*** columns .



**Step-5** Load Calendar dates table to Power BI data model and add start of year , month, day, week, quarter etc from the Date options .

The screenshot shows the Power Query Editor with a table named 'Calendar Table'. The table has 10 columns: Date, Year, Month Name, Month, Quarter, Week of Year, Week of Month, and three additional columns. The data is organized by date, starting from 27-01-2001 and ending on 23-02-2001. The 'Month Name' column shows 'January' for the first 10 rows and 'February' for the remaining 18 rows. The 'Month' column shows '1' for January and '2' for February. The 'Quarter' column shows '1' for January and '2' for February. The 'Week of Year' column shows values from 4 to 8. The 'Week of Month' column shows values from 1 to 8.

**Step- 6** Open the "Advanced Editor" window to see the full M code behind this Calendar table.

The screenshot shows the 'Advanced Editor' window in Power Query Editor. The title bar says 'Calendar Table'. The code is as follows:

```
let fnDateTable = (StartDate as date, EndDate as date) as table =>
let
    Source = Table.FromRows(Json.Document(Binary.Decompress(Binary.FromText("I45WQJLNUZDUZLWFSKJQUA", BinaryEncoding.Base64), Compression.Default))),
    #Changed Type1 = Table.TransformColumnTypes(Source,{{"Start date", type date}}),
    #Added Custom1 = Table.AddColumn(#Changed Type1, "End date", each Date.From(DateTime.LocalNow())),
    #Changed Type2 = Table.TransformColumnTypes(#Added Custom1,{{"End date", type date}}),
    #Added Custom2 = Table.AddColumn(#Changed Type2, "Dates", each (Number.From([#"Start date"])..Number.From([#"End date"]))),
    #Expanded Dates = Table.ExpandListColumn(#Added Custom2, "Dates"),
    #Changed Type3 = Table.TransformColumnTypes(#Expanded Dates,{{"Dates", type date}}),
    #Removed Columns1 = Table.RemoveColumns(#Changed Type3,{"Start date", "End date"}),
    #Inserted Year = Table.AddColumn(#Removed Columns1, "Year", each Date.Year([#"Dates"]), Int64.Type),
    #Inserted Month Name = Table.AddColumn(#Inserted Year, "Month Name", each Date.MonthName([#"Dates"]), type text),
    #Inserted Month = Table.AddColumn(#Inserted Month Name, "Month", each Date.Month([#"Dates"]), Int64.Type),
    #Inserted Quarter = Table.AddColumn(#Inserted Month, "Quarter", each Date.QuarterOfYear([#"Dates"]), Int64.Type),
    #Inserted Week of Year = Table.AddColumn(#Inserted Quarter, "Week of Year", each Date.WeekOfYear([#"Dates"]), Int64.Type),
    #Inserted Week of Month = Table.AddColumn(#Inserted Week of Year, "Week of Month", each Date.WeekOfMonth([#"Dates"]), Int64.Type),
    #Inserted Day Name = Table.AddColumn(#Inserted Week of Month, "Day Name", each Date.DayOfWeekName([#"Dates"]), type text),
    #Inserted Day = Table.AddColumn(#Inserted Day Name, "Day", each Date.Day([#"Dates"]), Int64.Type),
    #Inserted Day of Year = Table.AddColumn(#Inserted Day, "Day of Year", each Date.DayOfYear([#"Dates"]), Int64.Type)
in
    #Inserted Day of Year
fnDateTable
```

The code defines a function 'fnDateTable' that takes 'StartDate' and 'EndDate' as arguments. It starts with a 'Source' table, changes the type of 'Start date', adds an 'End date' column, changes its type, adds a 'Dates' column, expands it, changes its type, removes the original date columns, and then adds various date-related columns: Year, Month Name, Month, Quarter, Week of Year, Week of Month, Day Name, Day, and Day of Year. Finally, it returns the 'Inserted Day of Year' column.