



عنوان مقاله: بهترین راه ایجاد جدول داینامیک تقویم شمسی و میلادی در نرم افزار Power BI

نویسنده مقاله: تیم فنی نیک‌آموز

تاریخ انتشار: ۱۹ بهمن ۱۴۰۲

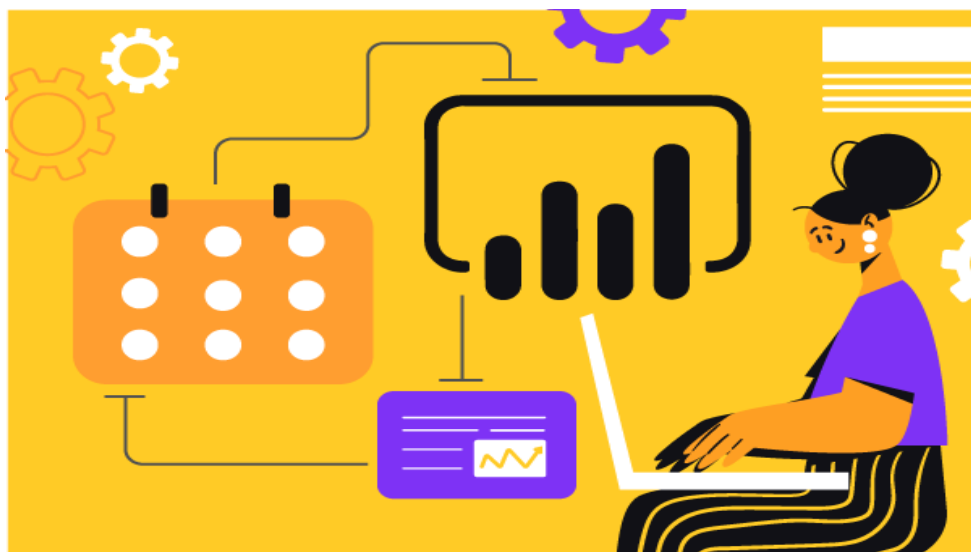
منبع: <https://nikamooz.com/create-a-solar-calendar-in-power-bi/>

تقویم شمسی در Power BI و استفاده از آن برای برخی از کاربران و سازمان‌ها حائز اهمیت است. ایجاد گزارش‌ها براساس سال، فصل و ماه بر مبنای تاریخ شمسی، به‌عنوان یکی از مهمترین چالش‌های کاربران ایرانی در کار با نرم افزار هوش تجاری Power BI تلقی می‌شود؛ زیرا تقویم پیش‌فرض نرم افزار Power BI، تقویم میلادی است. برای ایجاد این نوع تقویم‌ها، روش‌های مختلفی ارائه شده است؛ با این وجود، هر یک کاستی‌هایی نیز دارند. در این مقاله، به بررسی بهترین روش برای ایجاد جدول داینامیک تقویم شمسی و میلادی در Power BI می‌پردازیم.

روش های ایجاد جدول تقویم شمسی در Power BI

برای رفع مشکل تقویم شمسی در Power BI، لازم است یک جدول (DimDate) به‌عنوان جدول تاریخ شمسی اضافه شود که در آن، ستون‌های مختلف مرتبط با بازه زمانی وجود دارند. در جدول مذکور، یک ستون تاریخ میلادی، برای برقراری Relationship بین جدول DimDate و سایر جداول موردنیازتان وجود دارد. این جدول کاربردی، ابزاری مناسبی برای تجزیه و تحلیل داده‌ها در بازه‌های زمانی در زمان مدل‌سازی داده‌ها به شمار می‌رود.

پیش بررسی بیشتر مبحث استفاده از تاریخ شمسی در Power BI، توجه کنید که روش‌های زیر، از لحاظ عملکردی (Performance)، تفاوت چندانی باهم نخواهند داشت. تفاوت اصلی آن‌ها در سرعت، سادگی و داینامیک بودن‌شان در زمان استفاده است. در حال حاضر، جهت ایجاد جدول تقویم شمسی در Power BI یا همان جدول DimDate چندین راهکار زیر وجود دارند.



روش های مرسوم استفاده از تقویم شمسی در Power BI

برای استفاده از تقویم شمسی در Power BI ، چند رویکرد رایج زیر پیش روی شما وجود دارند:

۱. Import کردن جدول تقویم شمسی آماده (در قالب فرمت‌های اکسل یا CSV در نرم افزار Power BI)

۲. ایجاد جدول تقویم شمسی از طریق ایجاد کوئری در SQL Server و [انواع پایگاه داده](#)

۳. ایجاد جدول تقویم شمسی از طریق کد زبان M در پاور کوئری

معایب روش های متداول

روش‌های فوق، معایب و کاستی‌های زیر را دارا هستند:

۱. تغییرات اعمال شده در جدول به صورت داینامیک و پویا انجام نمی‌شوند.

۲. تغییرات در تعریف بازه زمانی و همچنین، افزایش و کاهش تعداد ستون‌ها در جدول تقویم شمسی، انعطاف‌پذیری ندارند.

با توجه به کاستی‌های مذکور، بهتر است از بهینه‌ترین راهکار ایجاد جدول داینامیک تقویم شمسی و میلادی در نرم افزار Power BI استفاده شود. در ادامه، به بررسی بهترین روش استفاده از تقویم شمسی در Power BI می‌پردازیم.

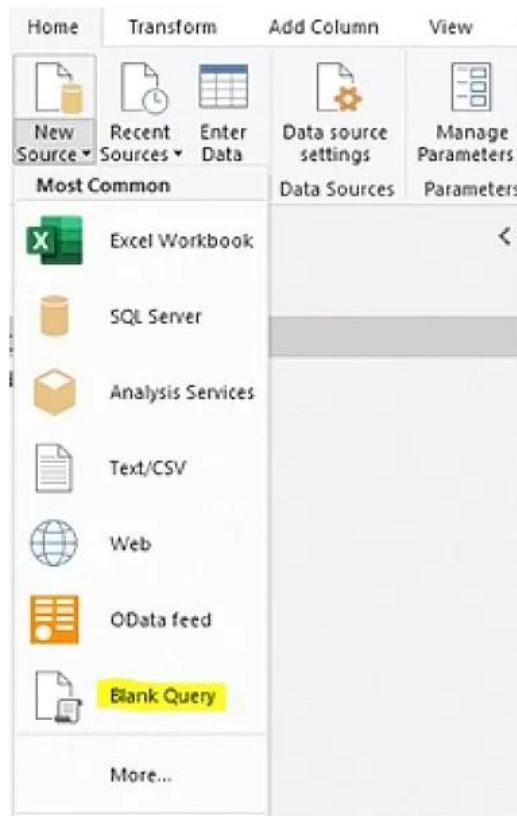
بهترین روش: ایجاد جدول تقویم شمسی (DimDate)

بهترین روش برای کار و استفاده از تقویم شمسی در Power BI ، ایجاد جدول تقویم شمسی داینامیک با استفاده از زبان M در پاور کوئری است. شما می‌توانید با انجام مراحل زیر و استفاده از کد نمایش داده شده، به راحتی جدول تاریخ شمسی را در Power Query ایجاد کنید. بدین طریق، امکان انتخاب بازه زمانی موردنظر و ستون‌های مدنظر برای شما ممکن خواهد بود.



۱- ورود به بخش Power Query

در وهله اول، لازم است نرم افزار Power BI را اجرا کرده و به بخش Power Query مراجعه کنید. سپس از منو بالا، گزینه New Source را انتخاب کرده و روی گزینه Blank Query کلیک کنید.



۲- افزودن کوئری مربوط به ایجاد جدول شمسی

پس از آن که به صفحه Power Query وارد شدید، روی گزینه Advanced Editor کلیک کرده تا صفحه پاپ آپ نمایش داده شود. اکنون کد کوئری نوشته شده زیر را کپی کنید و آن را در کادر Advanced Editor قرار دهید. این کوئری با استفاده از زبان M نوشته شده است. در نهایت، روی گزینه Done کلیک نمایید تا این کوئری اجرا شود.

```
let DateShamsiandmiladi = (
    Start.Date as date,
    optional End.Date as date,
    optional IncludeJalali2 as logical,
    optional IncludeJalali3 as logical,
    optional IncludeJDateKey as logical,
    optional IncludeJDateKey1 as logical,
    optional IncludeJYear as logical,
    optional IncludeJMonthN as logical,
    optional IncludeJDayN as logical,
    optional IncludeJDayN1 as logical,
    optional IncludeJMonthT as logical,
    optional IncludeJHalfYear as logical,
```

```

optional IncludeJSeasonT as logical,
optional IncludeJDayOfYear as logical,
optional IncludeJDayT as logical,
    optional IncludeYYYYMM as logical,
    optional IncludeQuarterN as logical,
    optional IncludeMyear as logical,
    optional IncludeMmonthN as logical,
    optional IncludeMmonthT as logical,
    optional IncludeMdayn as logical,
    optional IncludeMdayT as logical

) as table => let

    StartDate = Start.Date,
    EndDate = if End.Date = null then DateTime.Date(DateTime.LocalNow())
else End.Date,
    NumberOfDays = Duration.Days( EndDate - StartDate )+1,
    Dates = List.Dates(StartDate, NumberOfDays, #duration(1,0,0,0)),
    tablelist = Table.FromList(Dates, Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
    DateName = Table.RenameColumns(tablelist,{{"Column1", "Date"}}),
    DateFormat = Table.TransformColumnTypes(DateName,{{"Date", type date}}),
    Jalali1 = Table.AddColumn(DateFormat, "jalali", each Date.ToText([Date],
"yyyy-MM-dd", "fa-IR")),
    Jalali2 = if IncludeJalali2 <> false then Table.AddColumn(Jalali1,
"jalali2", each Date.ToText([Date], "yyyy/MM/dd", "fa-IR")) else Jalali1,
    Jalali3 = if IncludeJalali3 <> false then Table.AddColumn(Jalali2,
"jalali3", each Date.ToText([Date], "yyyy/M/d", "fa-IR")) else Jalali2,
    jdatekey = if IncludeJDateKey <> false then
Table.AddColumn(Jalali3,"jdatekey", each Text.Remove([jalali], {"-"}))
else Jalali3,
    jdatekey1 = if IncludeJDateKey1 <> false then
Table.TransformColumnTypes(jdatekey,{{"jdatekey", Int64.Type}}) else
jdatekey,
    jyear = if IncludeJYear <> false then Table.AddColumn(jdatekey1,
"jyear", each Number.FromText(Text.Start([jalali],4)), Int64.Type) else
jdatekey1,
    jmonthN = if IncludeJMonthN <> false then Table.AddColumn(jyear,
"jmonthN", each
Number.FromText(Text.End(Text.Start([jalali],7),2)),Int64.Type) else
jyear,
    jdayn = if IncludeJDayN <> false then Table.AddColumn(jmonthN, "jdayn",
each Text.End([jalali],2)) else jmonthN,
    jdayn1 = if IncludeJDayN1 <> false then
Table.TransformColumnTypes(jdayn,{{"jdayn", Int64.Type}}) else jdayn,
    jmonthT = if IncludeJMonthT <> false then Table.AddColumn(jdayn1,
"jmonthT", each Date.ToText([Date],"MMMM","fa-IR"), type text) else
jdayn1,
    jHalfYear = if IncludeJHalfYear <> false then Table.AddColumn(jmonthT,
"jHalfYear", each if [jmonthN]<=6 then "نیم سال اول" else "نیم سال دوم",
type text) else jmonthT,

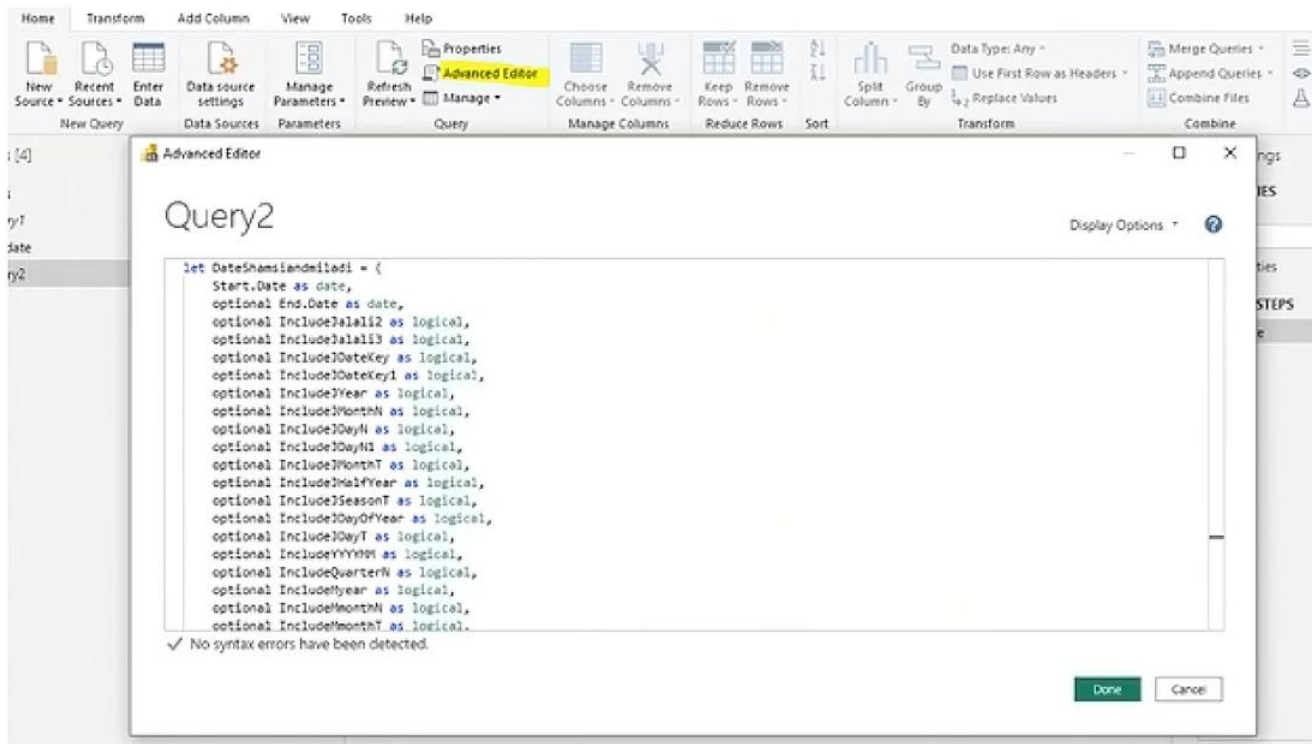
```

```

jseasonT = if IncludeJSeasonT <> false then Table.AddColumn(jHalfYear,
"jseasonT", each if List.Contains({1, 2, 3}, [jmonthN]) then "بهار" else
if List.Contains({4, 5, 6}, [jmonthN]) then "تابستان" else if
List.Contains({7, 8, 9}, [jmonthN]) then "پاییز" else "زمستان" , type
text) else jHalfYear,
JDayofYear = if IncludeJDayOfYear <> false then
Table.AddColumn(jseasonT, "JDayofYear", each if Date.DayOfYear([Date])>=80
then Date.DayOfYear([Date])-79 else Date.DayOfYear([Date])+286, Int64.Type)
else jseasonT,
jdayT = if IncludeJDayT <> false then Table.AddColumn(JDayofYear,
"jdayT", each Date.ToText([Date], "dddd", "fa-IR"), type text) else
JDayofYear,
YYYYMM = if IncludeYYYYMM <> false then Table.AddColumn(jdayT,
"YYYY-MM", each Text.BeforeDelimiter([jalali], "-", 1), type text) else
jdayT,
QuarterN = if IncludeQuarterN <> false then Table.AddColumn(YYYYMM,
"QuarterN", each if List.Contains({1, 2, 3}, [jmonthN]) then 1 else if
List.Contains({4, 5, 6}, [jmonthN]) then 2 else if List.Contains({7, 8,
9}, [jmonthN]) then 3 else 4, Int64.Type ) else YYYYMM,
Myear = if IncludeMyear <> false then Table.AddColumn(QuarterN, "Myear",
each Date.Year([Date]), Int64.Type) else QuarterN,
MmonthN = if IncludeMmonthN <> false then Table.AddColumn(Myear,
"MmonthN", each Date.Month([Date]), Int64.Type) else Myear,
MmonthT = if IncludeMmonthT <> false then Table.AddColumn(MmonthN,
"MmonthT", each Date.ToText([Date], "MMMM"), type text) else MmonthN,
Mdayn = if IncludeMdayn <> false then Table.AddColumn(MmonthT, "Mdayn",
each Date.DayOfWeek([Date]), Int64.Type) else MmonthT,
MdayT = if IncludeMdayT <> false then Table.AddColumn(Mdayn, "MdayT",
each Date.ToText([Date], "dddd"), type text) else Mdayn
in
MdayT,

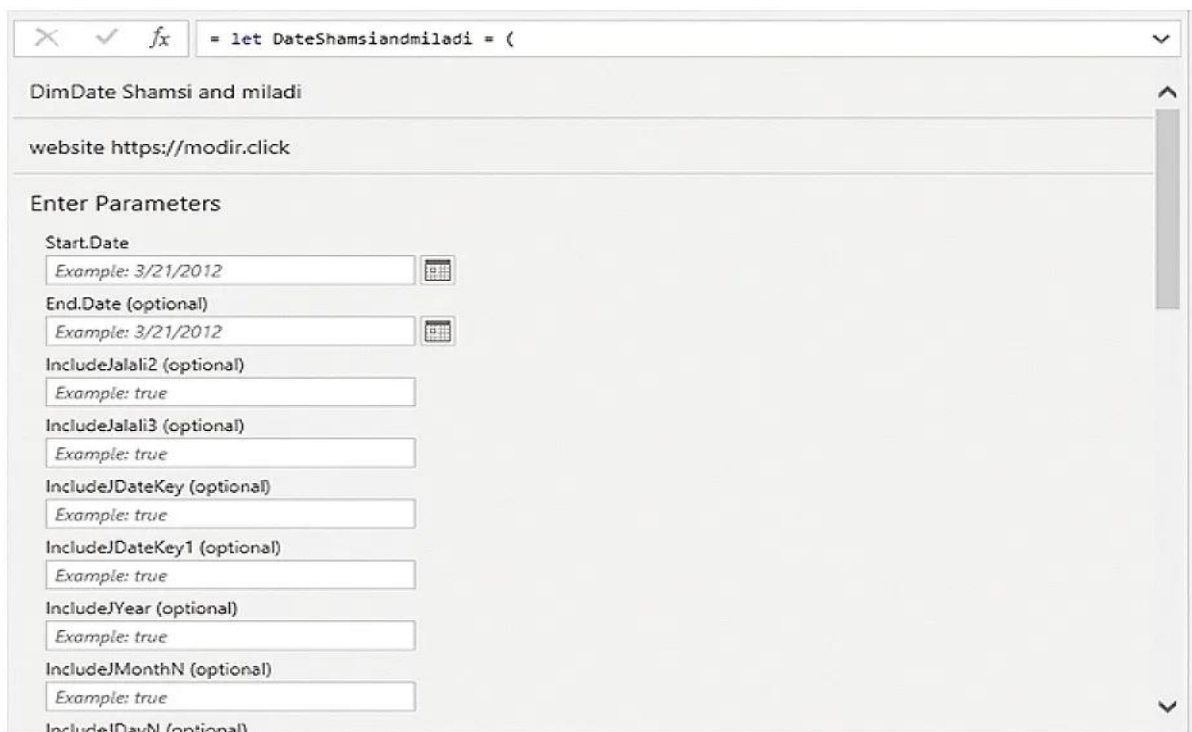
Documentation = [
Documentation.Name = "DimDate Shamsi and miladi",
]
in
Value.ReplaceType(DateShamsiandmiladi,
Value.ReplaceMetadata(Value.Type(DateShamsiandmiladi), Documentation))

```



۳- انتخاب بازه زمانی جدول دینامیک تقویم شمسی

بعد از تأیید کد جایگذاری شده، صفحه به شکل زیر نمایش داده می‌شود. در این صفحه، شما می‌توانید با انتخاب بازه زمانی (تاریخ شروع و پایان) به ایجاد جدول دینامیک تقویم شمسی اقدام کنید.



برای تعیین تاریخ شروع و پایان بازه زمانی در جدول، باید به صورت تاریخ میلادی، روز موردنظر را در فیلد مربوطه در بخش Enter Parameters وارد کنید. به عنوان مثال، اگر بخواهید داده های جدول شمسی از ابتدای سال ۱۴۰۱ تا انتهای سال ۱۴۰۲ ایجاد شود، لازم است در فیلد Start.Date تاریخ معادل میلادی آن یعنی ۲۰۲۲/۳/۲۱ و در فیلد End.Date تاریخ ۲۰۲۴/۳/۱۹ انتخاب شود.

شایان ذکر است اگر فیلد تاریخ پایان (End.Date) را خالی بگذارید، اسکرپت به صورت خودکار تاریخ روز جاری میلادی را درج می کند؛ به طوری که با هر بار Refresh شدن گزارش، به صورت خودکار تاریخ روز جاری نیز به جدول اضافه و آپدیت می شود.

در این قسمت، به صورت پیش فرض تمامی ستون های درج شده به جدول اضافه می شوند. لذا در صورت عدم نیاز به ایجاد برخی ستون ها و حذف ستون هایی، کافی است در فیلد ستون موردنظر کلمه False را تایپ کنید. توجه کنید که بعد از ایجاد جدول، امکان ویرایش مجدد ستون ها (حذف و اضافه) وجود دارد.

۴- نهایی سازی ایجاد جدول تقویم شمسی در Power BI

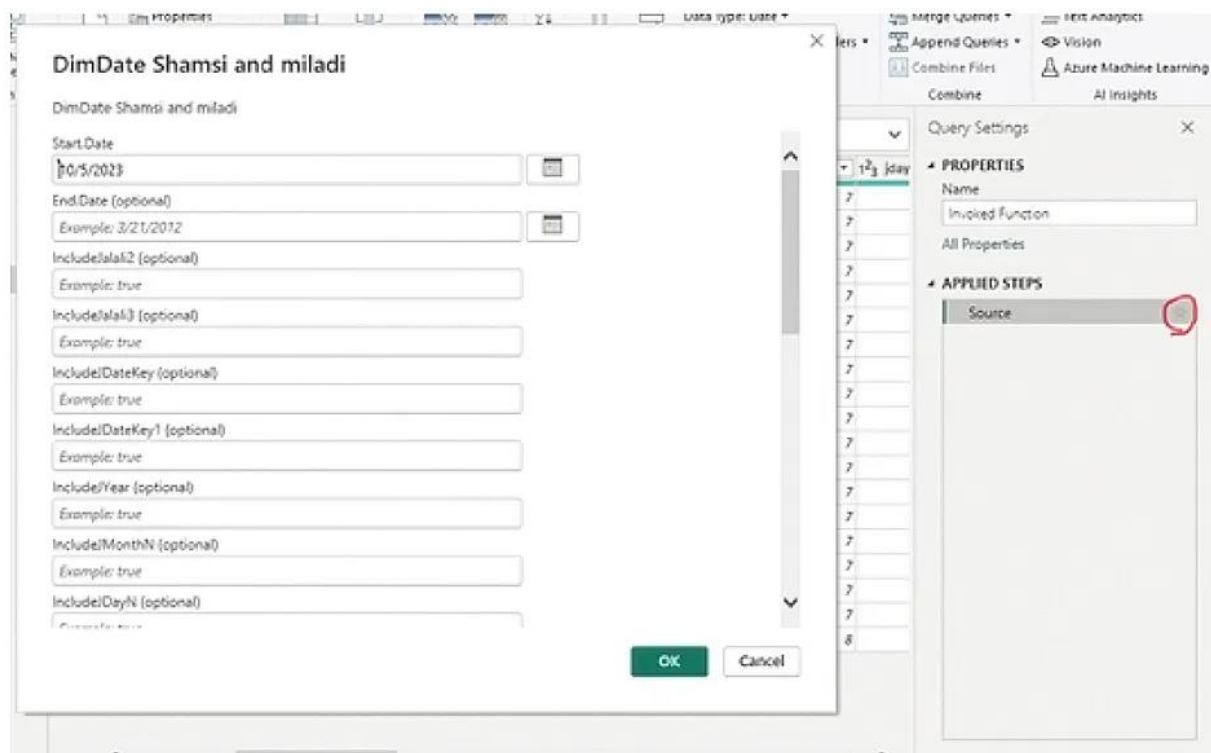
بعد از تنظیم پارامترهای مدنظر خود براساس مرحله قبل، به منظور ایجاد جدول تقویم شمسی، روی گزینه Invoke کلیک کنید.

The screenshot displays a Power BI interface. On the left, a table is shown with columns: 'date', 'year', 'month', and 'day'. The 'date' column contains dates from 1402/7/13 to 1402/7/31. The 'year' column contains the value 1402, and the 'month' column contains the value 7. The 'day' column contains values from 7 to 8. On the right, the 'Query Settings' panel is visible, showing the 'PROPERTIES' section with 'Name' set to 'Invoiced Function' and the 'APPLIED STEPS' section with 'Source' selected.

	date	year	month	day
1	1402/7/13	1402	7	7
2	1402/7/14	1402	7	7
3	1402/7/15	1402	7	7
4	1402/7/16	1402	7	7
5	1402/7/17	1402	7	7
6	1402/7/18	1402	7	7
7	1402/7/19	1402	7	7
8	1402/7/20	1402	7	7
9	1402/7/21	1402	7	7
10	1402/7/22	1402	7	7
11	1402/7/23	1402	7	7
12	1402/7/24	1402	7	7
13	1402/7/25	1402	7	7
14	1402/7/26	1402	7	7
15	1402/7/27	1402	7	7
16	1402/7/28	1402	7	7
17	1402/7/29	1402	7	7
18	1402/7/30	1402	7	7
19	1402/7/31	1402	7	8

۵- امکان ویرایش داینامیک بودن جدول

ویرایش و تغییر ویژگی داینامیک بودن جدول امکان پذیر است. برای ویرایش بازه زمانی و درج / حذف ستون های درج، می توان در بخش Query Settings روی گزینه تنظیمات Source کلیک کنید. حال در صفحه پاپ آپ باز شده، تنظیمات مدنظر خود را اعمال و سپس تأیید کنید.



۶- ایجاد ارتباط با سایر جداول

پس از ایجاد و ذخیره جدول تقویم شمسی، می توان به واسطه ایجاد روابط (Relationships) بین جداول از ستون های تاریخ میلادی در نرم افزار Power BI استفاده کرد.



جمع بندی: ساخت جدول داینامیک تقویم شمسی

برای استفاده از تقویم شمسی در Power BI روش‌هایی وجود دارند که در این مقاله به بررسی بهترین آن اشاره شد. شما می‌توانید با دنبال کردن گام‌های مذکور، به راحتی به این مزیت دسترسی داشته باشید. برای آشنایی بیشتر با این نرم افزار قدرتمند، پیشنهاد می‌کنیم [مقاله جامع Power BI](#) را نیز مطالعه کنید.