

A YTD-measure puzzle in DAX

PUBBLICATO MAGGIO 1, 2023 DI FRANCESCO BERGAMASCHI

When consulting in the BI field, it is common to find a Calendar table already present in the data warehouse or, in general, in the database to which you connect via Visual Studio, Power BI Desktop or Power Pivot for Excel.

As long as this calendar has the essential characteristics to use Time Intelligence in DAX (being complete with all days and having a primary key of type date), finding one ready to use is welcome.

However, one of the typical features of these calendars can cause a headache with the YTD DAX calculations which, in this article, will be shown and solved. Here I refer to my dear friend and customer Marcello Righi, Chief Marketing Officer of MOMA Group; it is during a consultancy to his company that this singular problem emerged.

The model in use is, as always, very simple and is shown in figure 1.

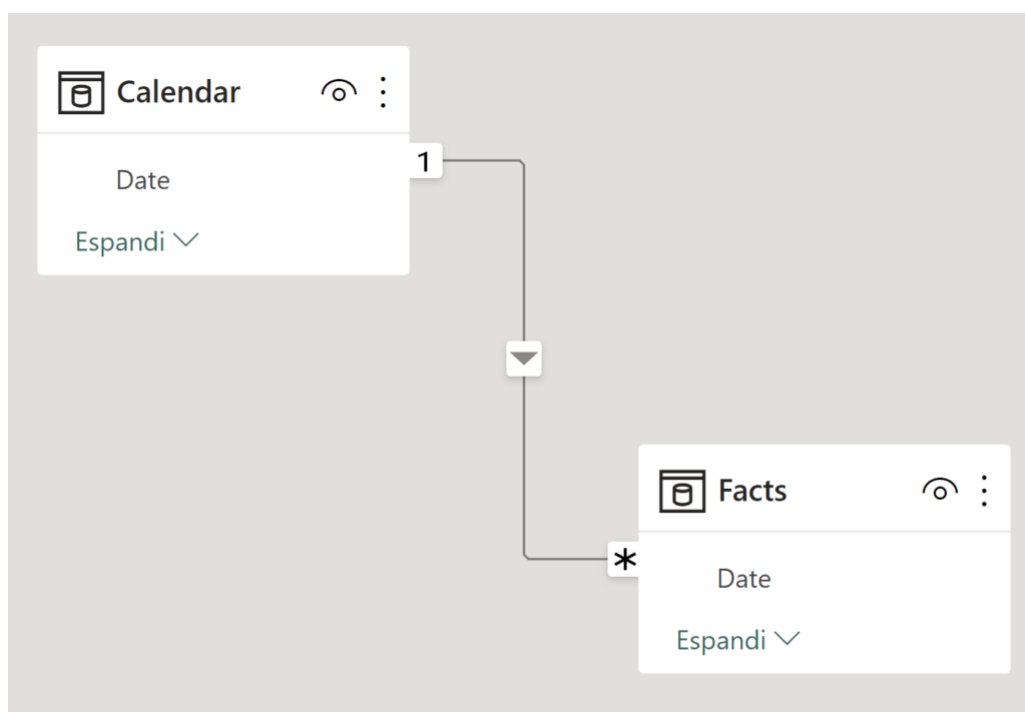


Figure 1

Il codice e i file contenuti in ogni singolo post sono rilasciati dagli autori così come sono e vengono proposti per scopi didattici. Ogni utilizzatore dei contenuti è tenuto a verificare autonomamente l'assenza di errori e la coerenza rispetto ai propri casi di applicazione.

In Figure 1, *Calendar* is the calendar table found in the database. Next, a calendar created in DAX will be added, which will be called *My Calendar*, and which will serve as debug.

Scenario

Suppose you want to make two simple calculations: the amount of sales and the annual cumulative sales, using the *Calendar* table. Figure 2 shows the result.

Year	Sales Amount	Sales Amount YTD Corp Cal
 2022	8.359.875	8.359.875
 2023	2.545.095	2.545.095
Totale	10.904.970	

Figure 2

As you can see, the value of the *Sales Amount YTD Corp Cal* measure is missing from the total. The size code is as follows:

```
Sales Amount YTD Corp Cal =
CALCULATE (
    [Sales Amount],
    DATESYTD ( 'Calendar'[Date] )
)
```

And here is the code of *Sales Amount*:

Sales Amount =

`SUM ( Facts[Amount] )`

What about the strange result in figure 2? The measure shows no defects. The prime suspect is the *Calendar* table. Yet, it has both characteristics listed above (completeness and provision of a primary key of data type).

At this point, it may be useful to create a calendar in DAX to see if it really is a calendar problem. The *My Calendar* table has the following rather classic code:

My Calendar =

VAR `TODAYCALC` =

`TODAY ()`

RETURN

`ADDCOLUMNS (`

`CALENDAR (`

`DATE ( YEAR ( MIN ( Facts[Date] ) ), 1, 1 ),`

`DATE ( YEAR ( MAX ( Facts[Date] ) ), 12, 31 )`

`),`

`"Year", YEAR ( [Date] ),`

`"YearMonth", YEAR ( [Date] ) & FORMAT ( MONTH ( [Date] ), "00" ),`

`"Present-Past-Future",`

`IF (`

`[Date] < TODAYCALC,`

`"Past",`

`IF ( [Date] = TODAYCALC, "Present", "Future" )`

`)`

`)`

Here is the effect of using *My Calendar*, once the relationship with the *Facts* table is created:

Year	Sales Amount	Sales Amount YTD My Cal
 2022	8.359.875	8.359.875
 2023	2.545.095	2.545.095
Totale	10.904.970	2.545.095

Figure 3

In figure 3, the total of the *Sales Amount YTD My Cal* measure is visible. Its code is identical to that of *Sales Amount YTD Corp Cal*, except for the use of a different calendar:

Sales Amount YTD My Cal =
`CALCULATE (`
 [Sales Amount],
 `DATESYTD ( 'My Calendar'[Date] )`
`)`

So, there is no doubt: the problem *is* the calendar.

What is the difference between the two calendars? Looking at the visuals it is not noticeable, but the calendar available in the database (*Calendar*) extends beyond the years of the facts (it reaches 12/31/2025 while the facts stop in April 2023). This is a typical case, given that the IT department understandably likes to do things so they don't have to go back to them for a while, plus the fact that the calendar could be used by other reports where the dates are, perhaps, forecast- or budget-related.

This should not be a problem, however, and in fact it is not for measurements that do not involve YTD calculations. In the YTD case, on the other hand, the problem exists: *DATESYTD*, in the presence of several years in the filter context, calculates the value of the measure first argument of *CALCULATE* in a range of dates that

goes from the first date to the maximum visible date, but *both refer to the maximum year in context*. If the measure is *BLANK* in that year, the total will be *BLANK*. Therefore, it is necessary to eliminate from the database calendar the years following the maximum year of the facts. In the case of the calendar written in DAX, it is natural to avoid this happening. Different is the case with a calendar created by the IT department.

Deleting the years following the maximum year of the facts manually from the database calendar, however, is simple but to be done every year. Here's how to do it automatically using Power Query: just add a custom column that calculates the year of today's date – M code: `Date.Year(DateTime.LocalNow())` – and then another column, this time conditional, which has the value, for example, *Y* if the calendar year is less than or equal to the year of today's date and *N* otherwise. After that, just filter that column by the *Y* value.

Once this is done, the *Sales Amount YTD Corp Cal* measure correctly displays the total value even when using the *Calendar* table:

Year	Sales Amount	Sales Amount YTD Corp Cal
 2022	8.359.875	8.359.875
 2023	2.545.095	2.545.095
Totale	10.904.970	2.545.095

Figure 4

To avoid that the YTD measures show values beyond today's date – or the maximum date of sale with a small modification – which would be correct as a calculation but inconvenient for users, the two measures can thus be modified (ref. this excellent [article](#) by SQLBI):

Sales Amount YTD Corp Cal =

```
CALCULATE (
  [Sales Amount],
  CALCULATETABLE (
    DATESYTD ( 'Calendar'[Date] ),
    'Calendar'[Present-Past-Future] <> "Future"
  )
)
```

Sales Amount YTD My Cal =

```
CALCULATE (
  [Sales Amount],
  CALCULATETABLE (
    DATESYTD ( 'My Calendar'[Date] ),
    'My Calendar'[Present-Past-Future] <> "Future"
  )
)
```

Year	Sales Amount	Sales Amount YTD Corp Cal
 2022	8.359.875	8.359.875
 2023	2.545.095	2.545.095
202301	582.739	582.739
202302	684.076	1.266.815
202303	668.945	1.935.760
202304	609.335	2.545.095
Totale	10.904.970	2.545.095

Figure 5

Il codice e i file contenuti in ogni singolo post sono rilasciati dagli autori così come sono e vengono proposti per scopi didattici. Ogni utilizzatore dei contenuti è tenuto a verificare autonomamente l'assenza di errori e la coerenza rispetto ai propri casi di applicazione.

Finally, how to obtain the total value of sales for all years with regard to the YTD measures? Here are two examples, one for each calendar considered:

Sales Amount YTD Corp Cal With Modified Total =

```
SUMX (
  VALUES ( 'Calendar'[Year] ),
  CALCULATE (
    [Sales Amount YTD Corp Cal]
  )
)
```

Year	Sales Amount	Sales Amount YTD Corp Cal	Sales Amount YTD Corp Cal With Modified Total
 2022	8.359.875	8.359.875	8.359.875
 2023	2.545.095	2.545.095	2.545.095
202301	582.739	582.739	582.739
202302	684.076	1.266.815	1.266.815
202303	668.945	1.935.760	1.935.760
202304	609.335	2.545.095	2.545.095
Totale	10.904.970	2.545.095	10.904.970

Figure 6

Sales Amount YTD My Cal With Modified Total =

```
SUMX (
  VALUES ( 'My Calendar'[Year] ),
  CALCULATE (
    [Sales Amount YTD My Cal]
  )
)
```

Year	Sales Amount	Sales Amount YTD My Cal	Sales Amount YTD My Cal With Modified Total
 2022	8.359.875	8.359.875	8.359.875
 2023	2.545.095	2.545.095	2.545.095
202301	582.739	582.739	582.739
202302	684.076	1.266.815	1.266.815
202303	668.945	1.935.760	1.935.760
202304	609.335	2.545.095	2.545.095
Totale	10.904.970	2.545.095	10.904.970

Figure 7

Conclusions

The calendars available in and in corporate databases in general are convenient and useful, sometimes indispensable in the case of specific, complex needs that can hardly be replicated in DAX. However, pay attention to the YTD calculations: this requires a calendar that does not go beyond 12/31 of the year of the maximum date of the facts. With a little effort in Power Query it is possible to automate the selection of corporate calendar lines in order to be in line with this need. This is less problematic in calendars written in DAX which are naturally extended no later than the year of the maximum date of the facts.

[.pbix file](#) [Download](#)

Il codice e i file contenuti in ogni singolo post sono rilasciati dagli autori così come sono e vengono proposti per scopi didattici. Ogni utilizzatore dei contenuti è tenuto a verificare autonomamente l'assenza di errori e la coerenza rispetto ai propri casi di applicazione.