

OFFSET – una nuova (e benvenuta) funzione DAX

PUBBLICATO SETTEMBRE 27, 2022 DI FRANCESCO BERGAMASCHI

In un recentissimo meeting in cui ero chiamato a risolvere qualche problema di DAX, il mio carissimo amico Marco Raule, Direttore HR di Eridania Italia, mi ha messo al corrente dell'esistenza di una funzione DAX che, come la nuovissima *EVALUATEANDLOG* per il processo di *debugging*, è ben nascosta e non ancora supportata da *Intellisense* ma che funziona alla perfezione già adesso ed è molto utile: *OFFSET*.

Marco aveva saputo lui stesso di questa funzione da un articolo del blog *Data – Marc* (ecco il link all'articolo: <https://data-marc.com/2022/09/21/how-offset-in-dax-will-make-your-life-easier/>) e io ne sono stato subito interessatissimo. Prima di proseguire, dunque, ringrazio Marco e, indirettamente, il creatore del blog Marc Lelijveld.

La funzione *OFFSET*, non ancora documentata e dunque non ancora supportata, risolve un problema annoso del DAX: la necessità di scrivere codice complesso quando si voglia referenziare un elemento diverso dal corrente in un contesto di riga.

Useremo, in questo articolo, un modello molto semplice composto da soltanto due tabelle: una dimensione (*Product*) e una tabella dei fatti (*Sales*). Il modello è mostrato in figura 1.

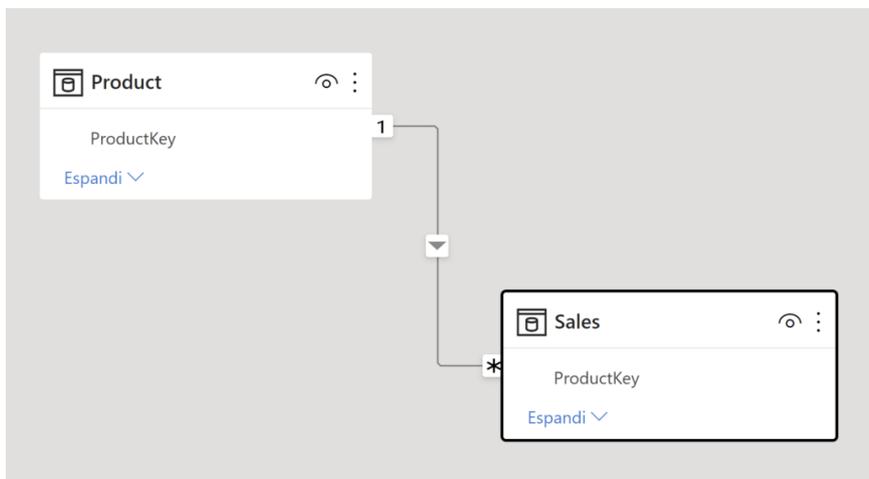


Figura 1

Sviluppo

Il caso d'uso di *OFFSET* è più semplice da mostrare in applicazione che da spiegare in astratto. Si consideri la figura 2 a seguire in cui è mostrata una matrice che mostra, per ogni colore di prodotto (colonna *'Product'[Color]*), le vendite (misura *Ricavi*). Si noti, in figura, che la colonna *'Product'[Color]* è ordinata in senso alfabetico crescente. Il codice di *Ricavi* è:

Ricavi =

SUMX (Sales, Sales[UnitPrice] * Sales[OrderQuantity])

Color Ricavi

| | |
|---------------|---------------------|
| Black | € 8.838.412 |
| Blue | € 2.279.096 |
| Multi | € 106.471 |
| NA | € 435.117 |
| Red | € 7.724.331 |
| Silver | € 5.113.389 |
| White | € 5.106 |
| Yellow | € 4.856.756 |
| Totale | € 29.358.677 |

Figura 2

Si supponga, adesso, di volere calcolare, per ogni colore, il valore di *Ricavi* in corrispondenza del colore subito precedente in senso alfabetico, in modo da potere, poi, calcolarne la differenza (figura 3):

| Color | Ricavi | Ricavi OFFSET -1 Color |
|---------------|---------------------|------------------------------|
| Black | € 8.838.412 | |
| Blue | € 2.279.096 | € 8.838.412 |
| Grey | | € 2.279.096 |
| Multi | € 106.471 | |
| NA | € 435.117 | € 106.471 |
| Red | € 7.724.331 | € 435.117 |
| Silver | € 5.113.389 | € 7.724.331 |
| Silver/Black | | € 5.113.389 |
| White | € 5.106 | |
| Yellow | € 4.856.756 | € 5.106 |
| Totale | € 29.358.677 | |

Figura 3

Ottenere il risultato in figura 3, fino ad ora, è stato piuttosto complesso. Una persona relativamente poco esperta di DAX non sarebbe mai stata in grado di generare la misura *Ricavi OFFSET -1 Color*, pur essendo un calcolo, in apparenza, molto semplice (soprattutto se si fa – erroneamente – riferimento alla facilità di realizzare un calcolo del genere in Excel).

Ecco il codice della misura *Ricavi OFFSET -1 Color* che usa la funzione *OFFSET* (in un'immagine visto che il codice non è ancora formattabile con DAX formatter per ottenerne la versione HTML), in figura 4:

```

1 Ricavi OFFSET -1 Color =
2 IF (
3     HASONEVALUE('Product'[Color]),
4     CALCULATE(
5         [Ricavi],
6         OFFSET(
7             -1,
8             ALLSELECTED('Product'[Color]),
9             ORDERBY('Product'[Color],ASC)
10        )
11    )
12 )

```

Figura 4

Ricordate che *Intellisense* mostrerà un errore, dunque non potrete formattare neanche con Bravo, fino a quando *OFFSET* non sarà rilasciata ufficialmente (stessa cosa per *EVALUATEANDLOG*).

Osservando la figura 4, è abbastanza semplice dedurre la sintassi di *OFFSET*, che consta di tre input:

1. l'entità dello sfasamento, cioè il numero di righe da saltare e la relativa direzione (segno + o -);
2. la tabella su cui creare il contesto a cui applicare lo sfasamento;
3. l'ordinamento dei valori in tabella, in modo da chiarire cosa si intenda per precedente/successivo.

Nel caso mostrato, lo sfasamento è stato posto pari a -1 (valore immediatamente precedente al quello attuale – è possibile cambiarlo o renderlo selezionabile dall'utente in modo dinamico), la tabella di riferimento è stata *ALLSELECTED('Product'[Color])* – in generale sarebbe sufficiente, in una matrice del genere, *ALL('Product'[Color])* per accedere anche ai colori diversi da quello del contesto attuale, l'uso di *ALLSELECTED* è in ogni caso molto utile in caso ci siano filtraggi esterni alla visual sulla colonna *'Product'[Color]* – e, infine, l'ordinamento è stato posto in senso alfabetico sulla

colonna 'Product'[Color] tramite la funzione *ORDERBY*, fino ad ora utilizzabile soltanto in DAX Studio.

Non è (ancora) possibile ottenere un meccanismo analogo nel caso in cui si ordini la matrice in senso, per esempio, decrescente sui valori della misura, un caso molto diffuso. Si osservi la figura 5 in cui la matrice mostra i valori della misura *Ricavi* ordinati in senso decrescente e i colori non hanno, di conseguenza, un ordinamento alfabetico.

| Color | Ricavi | Ricavi OFFSET -1 Color |
|---------------|---------------------|------------------------------|
| Black | € 8.838.412 | |
| Red | € 7.724.331 | € 435.117 |
| Silver | € 5.113.389 | € 7.724.331 |
| Yellow | € 4.856.756 | € 5.106 |
| Blue | € 2.279.096 | € 8.838.412 |
| NA | € 435.117 | € 106.471 |
| Multi | € 106.471 | |
| White | € 5.106 | |
| Grey | | € 2.279.096 |
| Silver/Black | | € 5.113.389 |
| Totale | € 29.358.677 | |

Figura 5

È evidente, in figura 5, che non c'è corrispondenza con i valori precedenti del colore questa volta, perché la misura *Ricavi OFFSET -1 Color* usa un ordinamento che non è quello attualmente usato nella matrice. Lo scopo qui è sempre lo stesso: calcolare il valore di *Ricavi* in corrispondenza del colore immediatamente

precedente (o sfasato in modo più esteso, basterà cambiare il valore dello sfasamento in valore assoluto e segno) ma questa volta seguendo l'ordinamento dei valori della misura *Ricavi*.

Il tentativo mostrato in figura 6, che arricchisce la tabella secondo parametro di *OFFSET* con *ADDCOLUMNS*, non porta, purtroppo, che ad un errore (stessa cosa se si generasse una variabile prima con lo stesso codice):

```

1 Ricavi OFFSET -1 Color MISURA =
2 IF (
3     HASONVALUE('Product'[Color]),
4     CALCULATE(
5         [Ricavi],
6         OFFSET(
7             -1,
8             ADDCOLUMNS(
9                 ALLSELECTED('Product'[Color]),
10                "@RicaviColore",
11                [Ricavi]
12            ),
13            ORDERBY([@RicaviColore],DESC)
14        )
15    )
16 )

```

! Impossibile utilizzare la funzione 'ORDERBY' aggiungendo colonne dalle funzioni *ADDCOLUMNS* o *SUMMARIZE*. Figura 6

E', dunque, necessario in questo caso, usare la tecnica già esistente, un po' articolata. Una possibile soluzione è la seguente misura:

Ricavi OFFSET -1 Color MISURA =

```

VAR RankRicaviColoreCorrente =
    RANKX ( ALLSELECTED ( 'Product'[Color] ), [Ricavi] )
VAR TabellaRankRicaviColore =
    ADDCOLUMNS (
        ALLSELECTED ( 'Product'[Color] ),
        "@RankRicaviColore", RANKX ( ALLSELECTED ( 'Product'[Color] ), [Ricavi] )
    )
RETURN
    IF (
        HASONVALUE ( 'Product'[Color] ),
        CALCULATE (

```

```
[Ricavi],
FILTER (
  TabellaRankRicaviColore,
  [@RankRicaviColore] = RankRicaviColoreCorrente - 1
)
)
```

In figura 7 è possibile notare come i valori siano adesso corretti secondo l'ordinamento decrescente della misura, anche in presenza di un filtro sui colori, grazie all'uso fatto di *ALLSELECTED*:

| Color | Ricavi | Ricavi OFFSET -1 Color | Ricavi OFFSET -1 Color MISURA |
|---------------|---------------------|------------------------|-------------------------------|
| Black | € 8.838.412 | | |
| Yellow | € 4.856.756 | € 5.106 | € 8.838.412 |
| Blue | € 2.279.096 | € 8.838.412 | € 4.856.756 |
| NA | € 435.117 | € 106.471 | € 2.279.096 |
| Multi | € 106.471 | € 2.279.096 | € 435.117 |
| White | € 5.106 | | € 106.471 |
| Silver/Black | | € 435.117 | € 5.106 |
| Totale | € 16.520.958 | | |

Color

- Seleziona tutto
- Black
- Blue
- Grey
- Multi
- NA
- Red
- Silver
- Silver/Black
- White
- Yellow

Figura 7

Conclusioni

OFFSET è una funzione molto potente e pratica. Semplificherà molto la scrittura del DAX e speriamo che arrivino altre funzioni di questo tipo per aiutare a diffondere il DAX e Tabular. Non è (ancora) possibile usare *OFFSET* quando si voglia cambiare elemento del contesto ma basandosi su un ordinamento che consti dei valori di una misura, nel qual caso bisogna usare le tecniche già esistenti, piuttosto articolate in quanto, spesso, usano *RANKX*.

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.