Come mostrare i valori di una misura soltanto in assenza di valori BLANK( ) in tutti i periodi

PUBBLICATO MAGGIO 31, 2022 *DI* FRANCESCO BERGAMASCHI

Ancora una volta, in questo articolo riportiamo un caso pratico di applicazione del DAX, facendo riferimento ad un reale caso di business.

Si supponga di volere mostrare le quantità vendute (o i valori di qualunque altra misura), cliente per cliente (o prodotto per prodotto, territorio per territorio e così via), soltanto per i clienti che hanno quantità vendute non vuote in tutti gli anni considerati (o mesi, trimestri e così via), nascondendo cioè i dati di tutti i clienti che hanno almeno un anno in cui tale quantità è BLANK().

Useremo Power Pivot per Excel per risolvere questo quesito, sapendo, tuttavia, che per passare a Power BI Desktop bastano pochi click.

Si faccia riferimento alla figura 1 per il modello dati in analisi.

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.

Tutti i diritti riservati.
Senza apportare modifiche, questo documento può essere liberamente diffuso.
Il contenuto può essere riprodotto, in tutto o in parte, citando l'autore e il sito di kubisco (www.kubisco.com).

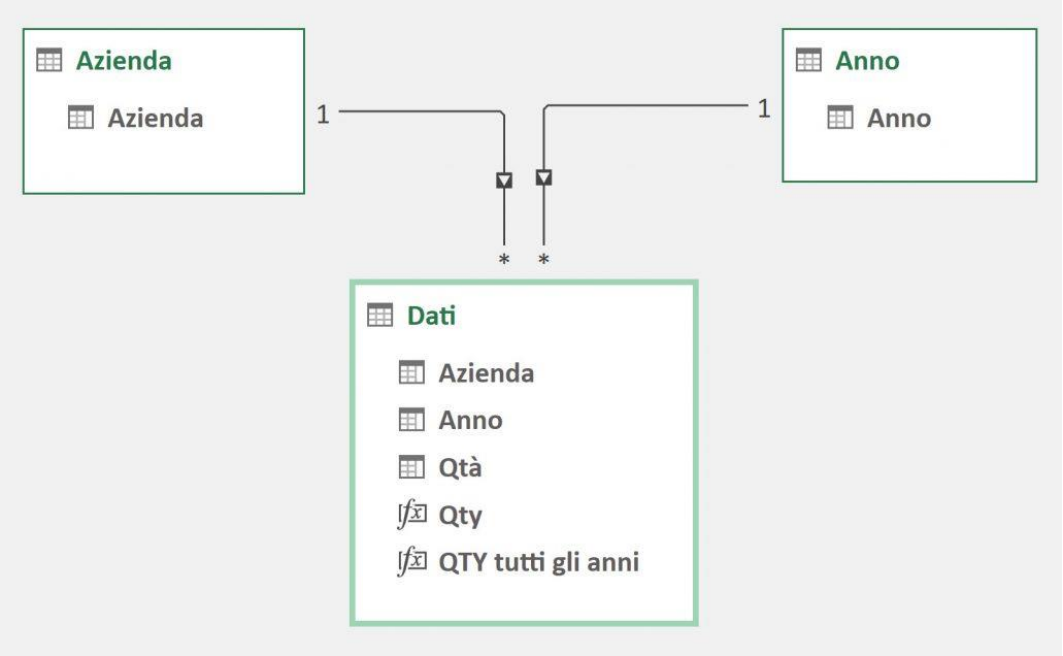

Figura 1

In figura 2, sono visibili le prime righe della tabella *Dati*. Nelle figure 3 e 4, rispettivamente, sono mostrate le tabelle *Anno* e *Azienda*.

Azienda	Anno	Qtà
A	2013	9.264
A	2014	9.034
A	2015	11.102
A	2016	11.711
A	2017	12.614
A	2018	13.705
A	2019	14.300
A	2020	9.118
A	2021	13.768
B	2015	4.378
B	2016	4.931
B	2017	6.280

Figura 2 – estratto della tabella *Dati*

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.

Tutti i diritti riservati.
Senza apportare modifiche, questo documento può essere liberamente diffuso.
Il contenuto può essere riprodotto, in tutto o in parte, citando l'autore e il sito di kubisco (www.kubisco.com).

Anno
2013
2014
2015
2016
2017
2018
2019
2020
2021

Figura 3

Azienda
A
B
C
D

Figura 4

Come visibile nelle figure sopra riportate, ci sono 4 aziende clienti (A, B, C e D) e nove anni, dal 2013 al 2021 estremi inclusi. Le transazioni incluse nella tabella dei fatti *Dati* descrivono quantità comprate dalle aziende clienti con un anno di riferimento per ognuna. Complessivamente, tutte le transazioni avvengono negli anni dal 2013 al 2021 estremi inclusi. Non tutti i clienti, tuttavia, hanno transazioni in ognuno degli anni, come è visibile in figura 5, in cui in una Power Pivot accompagnata da uno slicer mostriamo, azienda cliente per azienda cliente e anno per anno, le quantità vendute. Il codice della misura *QTY* è mostrato a seguire.

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.

Tutti i diritti riservati.
Senza apportare modifiche, questo documento può essere liberamente diffuso.
Il contenuto può essere riprodotto, in tutto o in parte, citando l'autore e il sito di kubisco (www.kubisco.com).

Etichette di riga	Qty
A	
2013	9.264
2014	9.034
2015	11.102
2016	11.711
2017	12.614
2018	13.705
2019	14.300
2020	9.118
2021	13.768
B	
2015	4.378
2016	4.931
2017	6.280
2018	9.294
2019	1.153
2020	1.400
2021	1.600
C	
2013	32.000
2014	8.326
2015	6.673
2016	5.135
2017	4.305
2018	5.519
2019	4.249
2020	4.826
2021	3.796
D	
2013	7.860
2014	8.630
2015	8.710
2016	8.860
2017	9.950
2018	9.900
2019	8.800
2020	7.550
Totale complessivo	278.741

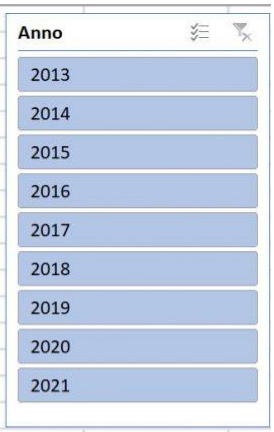

Figura 5

```
QTY = SUM ( Dati[Qtà] )
```

Sviluppo

In figura 5 si nota che soltanto le aziende clienti A e C hanno valori della misura *QTY* non nulli in ognuno dei nove anni.

La sfida è identificare queste aziende clienti e nascondere i valori della misura *QTY* per tutte le altre, in ognuno degli anni. Il risultato desiderato è mostrato in figura 6 in cui, alla già presente misura *QTY*, è stata aggiunta la misura *QTY tutti gli anni*.

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.

Tutti i diritti riservati.
Senza apportare modifiche, questo documento può essere liberamente diffuso.
Il contenuto può essere riprodotto, in tutto o in parte, citando l'autore e il sito di kubisco (www.kubisco.com).

Etichette di riga	Qty	QTY tutti gli anni
A		
2013	9.264	9.264
2014	9.034	9.034
2015	11.102	11.102
2016	11.711	11.711
2017	12.614	12.614
2018	13.705	13.705
2019	14.300	14.300
2020	9.118	9.118
2021	13.768	13.768
B		
2015	4.378	
2016	4.931	
2017	6.280	
2018	9.294	
2019	1.153	
2020	1.400	
2021	1.600	
C		
2013	32.000	32.000
2014	8.326	8.326
2015	6.673	6.673
2016	5.135	5.135
2017	4.305	4.305
2018	5.519	5.519
2019	4.249	4.249
2020	4.826	4.826
2021	3.796	3.796
D		
2013	7.860	
2014	8.630	
2015	8.710	
2016	8.860	
2017	9.950	
2018	9.900	
2019	8.800	
2020	7.550	
Totale complessivo	**278.741**	**179.445**

Figura 6

Come creare la misura *QTY tutti gli anni*? Come sempre, prima di scrivere del codice DAX, si ragioni su che tabelle o valori scalari è necessario avere per procedere.

Sarà necessario:

1 – disporre di un valore scalare che rappresenti il totale numero di anni (nove, nel nostro caso) indipendentemente dal *filter context* in essere;

2 – disporre, per ognuna delle aziende clienti, del numero di anni in cui ci sia almeno una transazione per la specifica azienda cliente considerata.

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.

Tutti i diritti riservati.
Senza apportare modifiche, questo documento può essere liberamente diffuso.
Il contenuto può essere riprodotto, in tutto o in parte, citando l'autore e il sito di kubisco (www.kubisco.com).

Avendo quanto listato nei punti 1 e 2, si potrà, azienda cliente per azienda cliente, aggregare con una somma il risultato della misura *QTY* solo quando il valore calcolato al punto 1 coincida con quello calcolato al punto 2 per l'azienda cliente in esame. In caso contrario, la misura *QTY* non sarà invocata.

Ecco il codice che implementa in DAX quanto progettato mentalmente:

```
QTY tutti gli anni =
VAR TuttiGliAnni =
   COUNTROWS (
      ALL ( Anno[Anno] )
   )
RETURN
   SUMX (
      VALUES ( Azienda[Azienda] );
      VAR AnniAziendaCorrente =
         COUNTROWS (
            FILTER (
               ALL ( Anno[Anno] );
               NOT (
                  ISEMPTY (
                     CALCULATETABLE ( Dati )
                  )
               )
            )
         )
      RETURN
         IF (
            AnniAziendaCorrente = TuttiGliAnni;
            [Qty]
         )
   )
```

Si notino i seguenti punti al riguardo della misura *QTY tutti gli anni*:

3 – l'uso di *COUNTROWS (ALL (Anno[Anno]))* può risultare criptico. Perché non usare *COUNTROWS (ALL (Anno))* ? In effetti, la tabella *Anno* è una dimensione con una sola colonna che è anche chiave primaria. Il codice usato, tuttavia, vuole ricordare che è sempre bene selezionare il numero minimo di colonne necessarie per il calcolo in esame. Dunque, si è preferito lasciare questo codice, in modo da

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.

Tutti i diritti riservati.
Senza apportare modifiche, questo documento può essere liberamente diffuso.
Il contenuto può essere riprodotto, in tutto o in parte, citando l'autore e il sito di kubisco (www.kubisco.com).

inquadrare come procedere per listare gli anni nel modo più efficiente possibile nel caso di uso di una vera e propria tabella *Calendario* o comunque di una tabella temporale con granularità più fine di *Anno*;

4 – l'uso di *CALCULATETABLE* è necessario per fare scattare la *context transition* e filtrare, così, la tabella *Dati* per il valore correntemente iterato di *VALUES (Azienda [Azienda])* aggiungendo tale filtro a quelli già presenti nel *filter context*

5 – la misura ha un funzionamento non necessariamente utile se si selezionano soltanto alcuni anni attraverso lo slicer, si osservi la figura 7. Pur avendo tutte le imprese dati nei tre anni selezionati, soltanto quelle che hanno dati in tutti i nove anni (ancora una volta, A e C) mostrano dati;

Etichette di riga	Qty	QTY tutti gli anni
A		
2015	11.102	11.102
2016	11.711	11.711
2017	12.614	12.614
B		
2015	4.378	
2016	4.931	
2017	6.280	
C		
2015	6.673	6.673
2016	5.135	5.135
2017	4.305	4.305
D		
2015	8.710	
2016	8.860	
2017	9.950	
Totale complessivo	94.649	51.540

Figura 7

6 – la misura *QTY tutti gli anni* funziona bene al totale soltanto se il calcolo che viene aggregato azienda per azienda e anno per anno è additivo come nel caso in esame (*QTY* è una misura additiva che aggrega tramite *SUM*). Infatti, la misura *QTY tutti gli anni* usa una *SUMX* – e non funzionerebbe bene, per fare un esempio, nel caso di un calcolo aggregato, azienda per azienda e anno per anno, facente uso di *DISTINCTCOUNT* che non è additivo per definizione e

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.

Tutti i diritti riservati.
Senza apportare modifiche, questo documento può essere liberamente diffuso.
Il contenuto può essere riprodotto, in tutto o in parte, citando l'autore e il sito di kubisco (www.kubisco.com).

risulterebbe in un valore errato. Un progetto ben fatto, è bene ribadirlo, deve considerare l'eventuale non additività.

Conclusioni

Il codice che risolve lo scenario proposto non è particolarmente complesso. Tuttavia, per produrlo, bisogna avere chiaro cosa si vuole fare e conoscere i fondamenti del DAX come i contesti di valutazione, la *context transition* e gli iteratori (di cui *SUMX* è una autorevole rappresentante). Ancora una volta, ribadiamo che per lavorare bene col DAX è necessario prima disegnare a mano o con la mente le tabelle da iterare – e i valori da calcolare in eventuali colonne aggiuntive – ed elencare i valori scalari necessari. Una volta chiaro il piano, si tratta di trovare le funzioni DAX che risolvono lo specifico passaggio tecnico che traduce il progetto in realtà. No, non è finita: bisogna anche avere dimestichezza con le funzioni ed il loro uso pratico ed è per questo che suggeriamo di crescere col DAX mediante una continua oscillazione tra teoria e pratica.

file.xlsxDownload

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.

Tutti i diritti riservati.
Senza apportare modifiche, questo documento può essere liberamente diffuso.
Il contenuto può essere riprodotto, in tutto o in parte, citando l'autore e il sito di kubisco (www.kubisco.com).