

Row-level security su parametri campo in Power BI

PUBBLICATO NOVEMBRE 7, 2022 DI FRANCESCO BERGAMASCHI E DAVID BIANCONI

In un recente aggiornamento di Power BI Desktop, è stata annunciata al *preview* dei parametri campo (*field parameter*). La funzionalità permette di selezionare, tramite comodi filtri visivi (*slicer*), le colonne e le misure da includere in una o più *visual* di Power BI.

Tra le caratteristiche più particolare dei parametri campo, spicca il fatto che essi nascono in tabelle calcolate create *ad hoc*. Dunque, visto che in queste tabelle sono listate, in diverse righe, le colonne e le misure coinvolte, è possibile applicare a queste tabelle calcolate la sicurezza di riga (*row-level security*, *RLS*). La conseguenza di ciò è che, grazie a questa nuova funzionalità, è possibile non solo rendere autonomi i consumatori dei report nel modificare le misure e le colonne coinvolte in una o più *visual* (senza dovere avere il diritto di modifica vero e proprio sul report – che non tutti i progettisti vogliono concedere – e senza dovere, avendo il diritto di modifica, effettivamente cercare tali misure e colonne nel pannello campi sull'interfaccia cloud del report, operazione non per tutti semplice), ma anche **filtrare la lista delle misure e delle colonne esposte negli slicer creati dai campi parametro sulla base dell'utente consumatore, tramite la creazione di ruoli sulle tabelle calcolate dei campi parametro.**

In parole semplici, un utente che appartiene alla produzione potrà, in modo automatizzato, trovare negli *slicer* soltanto le colonne e le misure che più gli interessano visto il suo dipartimento di appartenenza. Allo stesso modo, una persona che lavora nelle vendite vedrà disponibili altre colonne e misure.

Sviluppo

Due premesse:



1. come già accennato, la funzionalità dei parametri campo è, al momento della scrittura di questo articolo, in *preview*. Questo vuol dire che Microsoft non offre supporto in caso di difficoltà. Dunque, è bene esplorare le funzionalità ma non ancora metterla in produzione;
2. la tecnica che verrà a breve esposta non costituisce *security* bensì è una customizzazione. Infatti, se un utente ha il diritto di modifica su un report e procede in tal senso, le colonne e le misure saranno **tutte visibili**. Nulla di ciò che sarà mostrato in questo articolo crea vera e propria sicurezza. Se si vuole veramente nascondere misure, colonne e tabelle per alcuni utenti (questa sì che sarebbe sicurezza), bisogna usare la sicurezza di oggetto (*object-level security*). Per inciso, quest'ultima tecnica non permette di nascondere misure, bensì soltanto colonne e tabelle, ma, attraverso un *work-around*, è in effetti possibile nascondere misure. La *row-level security (RLS)* non è, invece, usata per nascondere oggetti bensì righe di una tabella (esempio: alcuni territori devono essere nascosti ad un utente che appartiene alle vendite per ragioni di assegnazione di aree di vendita ad altri agenti).

In questo articolo, dunque, si mostra un'applicazione della RLS che impedisce di vedere alcune righe di una o più tabelle la qual cosa, grazie alla funzionalità campi parametro, ha l'effetto visivo di nascondere dagli *slicer* alcune colonne e misure ma non, ripetiamo, **di renderle inaccessibili nel modello** (per ottenere quest'ultimo effetto servirebbe la *object-level security*). Non trattandosi di righe di dati ma di righe di tabelle tecniche create dai campi parametro, in questo caso la *row-level security* "non è" *security* nel senso che lo è soltanto in queste tabelle tecniche che sono scollegate dal modello come mostrato in figura 1.

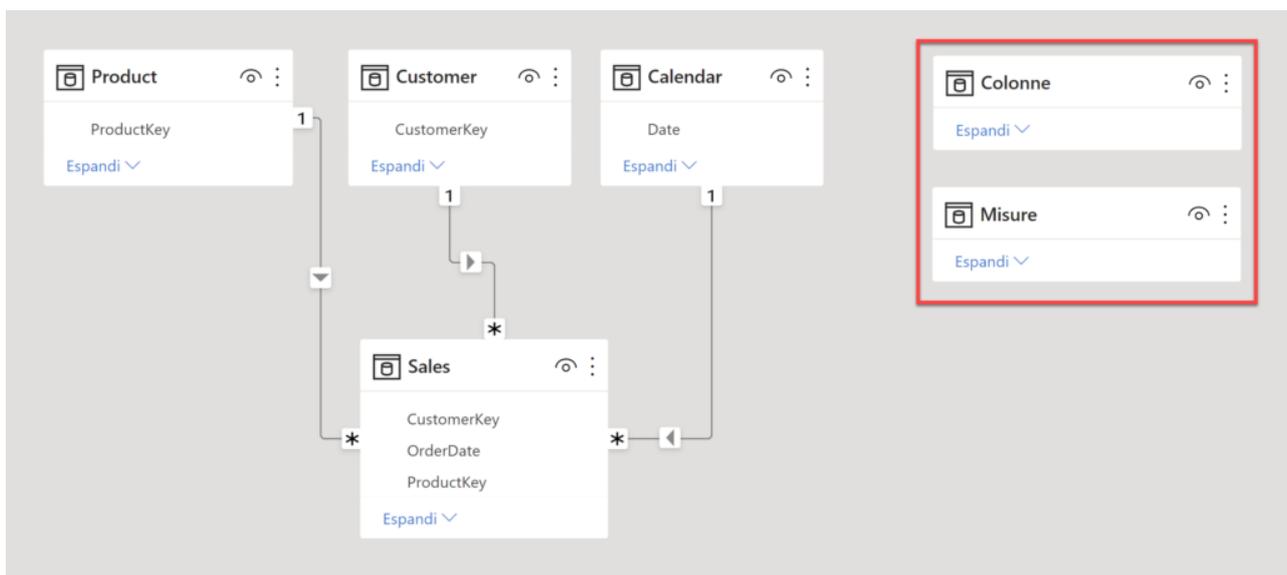


Figura 1

Come ultimo dettaglio introduttivo, sarà mostrata la tecnica applicata ad un parametro campo costituito da una serie di misure. La stessa tecnica può essere applicata ad un parametro campo costituito da una serie di colonne. In linea di principio, è possibile creare un parametro campo che annoveri al suo interno sia colonne che misure, ma lo sconsigliamo vivamente.

Sviluppo

Si consideri il seguente report, in cui sono stati creati due campi parametro: *Misure* e *Colonne*. Si tratta di tabelle, che sono state modificate per permettere di avere la stessa misura, nel caso in esame, listata più di una volta in modo da poterla attribuire a più dipartimenti.

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.

✕
✓

```

1 Colonne = {
2   ("CalendarYear", NAMEOF('Calendar'[CalendarYear]), 0),
3   ("Date", NAMEOF('Calendar'[Date]), 1),
4   ("Gender", NAMEOF('Customer'[Gender]), 2),
5   ("EnglishEducation", NAMEOF('Customer'[EnglishEducation]), 3),
6   ("Color", NAMEOF('Product'[Color]), 4),
7   ("SizeRange", NAMEOF('Product'[SizeRange]), 5)
8 }
```

Colonne	Campi di Colonne	Ordine di Colonne
CalendarYear	'Calendar'[CalendarYear]	0
Date	'Calendar'[Date]	1
Gender	'Customer'[Gender]	2
EnglishEducation	'Customer'[EnglishEducation]	3
Color	'Product'[Color]	4
SizeRange	'Product'[SizeRange]	5

Figura 4

Non si approfondirà la funzione *NAMEOF*, tuttavia si noti che la tabella è una lista di stringhe e di interi – questi ultimi servono a decidere l'ordine di apparizione del campo in un report. Le tabelle sono modificabili perché sono tabella calcolate. In effetti, si è intervenuti soltanto sulla tabella calcolata -ancora generata a fronte del parametro campo – *Misure*, eccone la versione finale dove si noterà che:

1. la stessa misura è listata più volte a seguito della modifica, come già accennato, per poterla attribuire a diversi dipartimenti;
2. è stata aggiunta una colonna con il dipartimento autorizzato ad avere quella particolare misura in selezione (figura 5). Non è possibile, in fase di creazione del campo parametro, al momento della scrittura di questo articolo, aggiungere più volte lo stesso oggetto.

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.

✕ ✓

```

1 Misure = {
2   ("Costi", NAMEOF('Sales'[Costi]), 3, "Produzione"),
3   ("Costi", NAMEOF('Sales'[Costi]), 3, "Marketing"),
4   ("Margini", NAMEOF('Sales'[Margini]), 4, "Marketing"),
5   ("Margini pct", NAMEOF('Sales'[Margini pct]), 5, "Marketing"),
6   ("Vendite", NAMEOF('Sales'[Vendite]), 0, "Vendite"),
7   ("Volumi Produzione", NAMEOF('Sales'[Volumi Produzione]), 1, "Produzione"),
8   ("Volumi Produzione", NAMEOF('Sales'[Volumi Produzione]), 1, "Marketing"),
9   ("Volumi Produzione", NAMEOF('Sales'[Volumi Produzione]), 1, "Vendite")
10 }
```

Misure	Campi di Misure	Ordine di Misure	Dipartimento
Costi	'Sales'[Costi]	3	Produzione
Costi	'Sales'[Costi]	3	Marketing
Margini	'Sales'[Margini]	4	Marketing
Margini pct	'Sales'[Margini pct]	5	Marketing
Vendite	'Sales'[Vendite]	0	Vendite
Volumi Produzione	'Sales'[Volumi Produzione]	1	Produzione
Volumi Produzione	'Sales'[Volumi Produzione]	1	Marketing
Volumi Produzione	'Sales'[Volumi Produzione]	1	Vendite

Figura 5

In figura 5, si noti che soltanto le misure *Vendite* e *Volumi Produzione* sono pertinenti al dipartimento vendite. Dunque, selezionando il ruolo *Vendite*, ecco come si mostra il report (figura 6).

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.

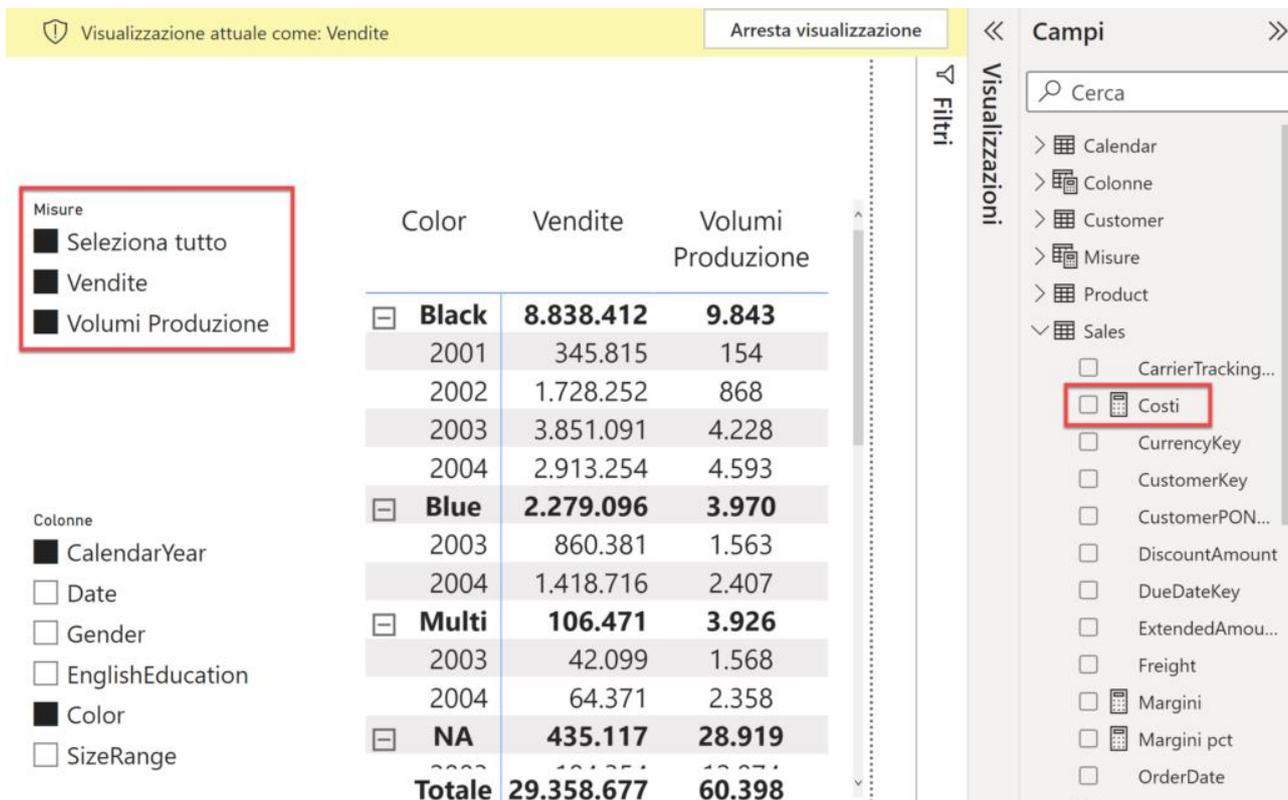


Figura 6

Si noti, in figura 6, che la misura *Costi* è ancora visibile tra i campi di *Sales* – e potrebbe essere inclusa nella matrice mostrano i suoi valori –, ecco cosa si intende per “non è security” in questo caso: sono state soltanto nascoste alcune misure dallo *slicer* ma non dal modello e solo in quest'ultimo caso si potrebbe parlare veramente di *security*. L'effetto è comunque molto interessante e può aiutare a rendere immediato e flessibile l'uso di report anche in presenza di decine e decine di misure.

Per concludere, ecco la sintassi usata per i filtri dei ruoli, che sono stati creati in modo statico (figura 7), gli altri filtri sono del tutto simili ma riportano un diverso nome del dipartimento (*Produzione* o *Vendite*).

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.

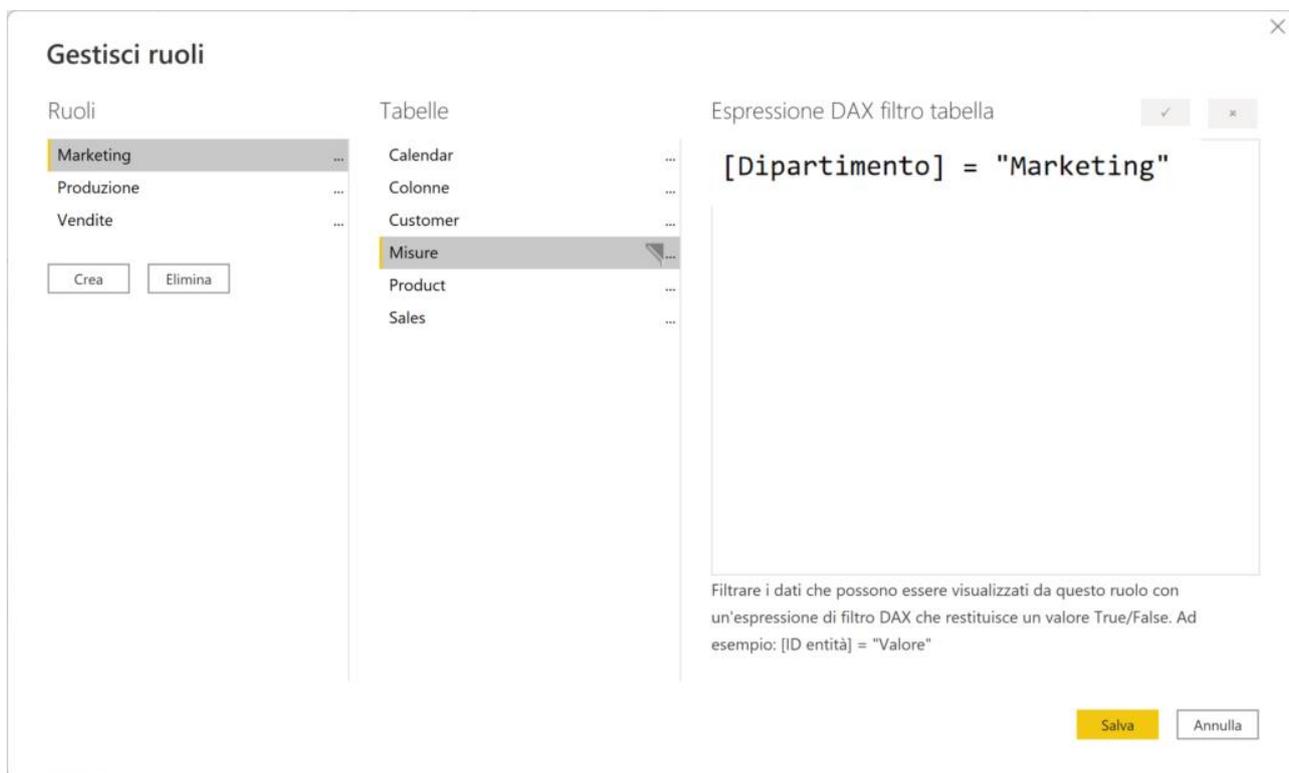


Figura 7

Si noti i parametri campo sono una caratteristica del client che manda le query e non di Tabular. Per esempio, Excel non è un client dotato di questa funzionalità, dunque non è possibile consumare un report Power BI come quello mostrato in questo articolo tramite Excel.

Conclusioni

Power BI offre tantissime funzionalità che possono essere usate in modo creativo per migliorare la flessibilità e la customizzazione dei report. Pensate out of the box e provate, provate, provate. Tutto ciò che è utile a fare sentire i consumatori del report al centro del processo di design, aiuterà a fare circolare i report e a distribuire Intelligence. Si raccomanda di notare che quanto mostrato in questo articolo non è, a tutti gli effetti, *security*, se non applicata ad una tabella scollegata dal modello che, in effetti, non nasconde alcuna informazione dal modello stesso.

[file.pbixDownload](#)