

Drillthrough in Excel basato su un dataset Power BI

PUBBLICATO MAGGIO 1, 2022 DI FRANCESCO BERGAMASCHI

La funzionalità *drillthrough* in Excel è disponibile da molti anni, risalendo agli albori delle tabelle pivot classiche in Excel, quelle basate su una singola tabella, connesse ai dati del file Excel stesso o ad un database *Analysis Services Multidimensional*.

Questa funzionalità è sempre stata giudicata molto utile dagli utenti e, nello ambito di un file Excel connesso ad un dataset Power BI, offre funzionalità molto potenti, per quanto in molti casi non note, tra cui la possibilità di scegliere quali colonne rendere visibili quando viene lanciato il *drillthrough* stesso. La funzionalità *drillthrough*, infatti, è attiva anche quando Excel si collega a database *Analysis Services Tabular*, dunque per i dataset Power BI e Power Pivot. La *Detail Rows Expression* permetterà di customizzare le colonne mostrate e gli eventuali filtri di *Row-level security* verranno applicati.

Inoltre, benché la funzionalità *drillthrough* appaia inizialmente limitata all'estrazione di 1.000 righe, in questo articolo si vedrà che è semplice portare tale numero al limite fisico di Excel di 1.048.576 righe.

La generazione del set di righe sottostanti a un calcolo tramite *drillthrough* è oltremodo veloce e l'uso di Excel come tool di consumo può aiutare a superare l'iniziale diffidenza di alcuni utenti nei confronti di Power BI - o la generale resistenza al cambiamento -, permettendo di usare la potenza di Power BI senza che l'utente lo usi esplicitamente, bensì restando nel noto e confortevole ambiente Excel.

Si vedrà in questo articolo, infine, come creare misure customizzate che facciano percepire all'utente che i report sono stati creati sulla base delle sue esigenze, seguendo il trend della *self-service BI*, di cui Power BI è il rappresentante principale.

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.

La funzionalità *drillthrough*, in ambito di un file Excel collegato ad un dataset Power BI, può essere usata in due casi:

- 1 – Excel viene connesso ad un dataset Power BI pubblicato su Power BI Cloud, indipendentemente dal fatto che ciò avvenga attraverso la funzionalità *Analyze in Excel* di Power BI Cloud o connettendosi al dataset sul Cloud da un file Excel vuoto;
- 2 – Excel viene connesso ad un dataset Power BI locale, cioè ad un file Power BI Desktop (.pbix) che giace sul PC del progettista o su un server di sviluppo.

Il caso 1 rappresenta lo scenario di riferimento di questo articolo in quanto il caso 2 riguarda un file Excel *non persistente*, cioè utile soltanto al progettista per verifiche di *debug*. D'altro canto, per il caso 1 è necessaria la licenza pro di Power BI mentre non servono licenze di Power BI per il caso 2.

Il video allegato al presente articolo è l'ultimo di una serie di quattro, l'intera playlist è disponibile su canale YouTube di kubisco (<https://www.youtube.com/c/kubisco>).

In calce a questo articolo tre note che si raccomanda di leggere, in modo da comprendere come ottenere Tabular Editor, un tool necessario per fare quanto qui mostrato e per potere approfondire, se di interesse per la community, alcuni aspetti ulteriori di customizzazione.

Sviluppo

Si consideri di avere un report in Power BI Desktop (per semplicità non ci si addentererà nella separazione del *dataset* dal *report*, passaggio che si raccomanda tuttavia sempre di svolgere, e nella relativa attività di assegnazione di diritti di lettura dal *dataset* agli utenti) e di volerlo rendere disponibile al cliente sul suo *tenant* Power BI o ai colleghi sul proprio *tenant* aziendale. Per fare ciò è necessario pubblicare il report su un'area di lavoro condivisa (da qui, ma non solo, la necessità della licenza pro per chi condivide e per chi riceve la condivisione). La pubblicazione avviene attraverso il tasto *Pubblica* di Power BI Desktop, si veda la figura 1.

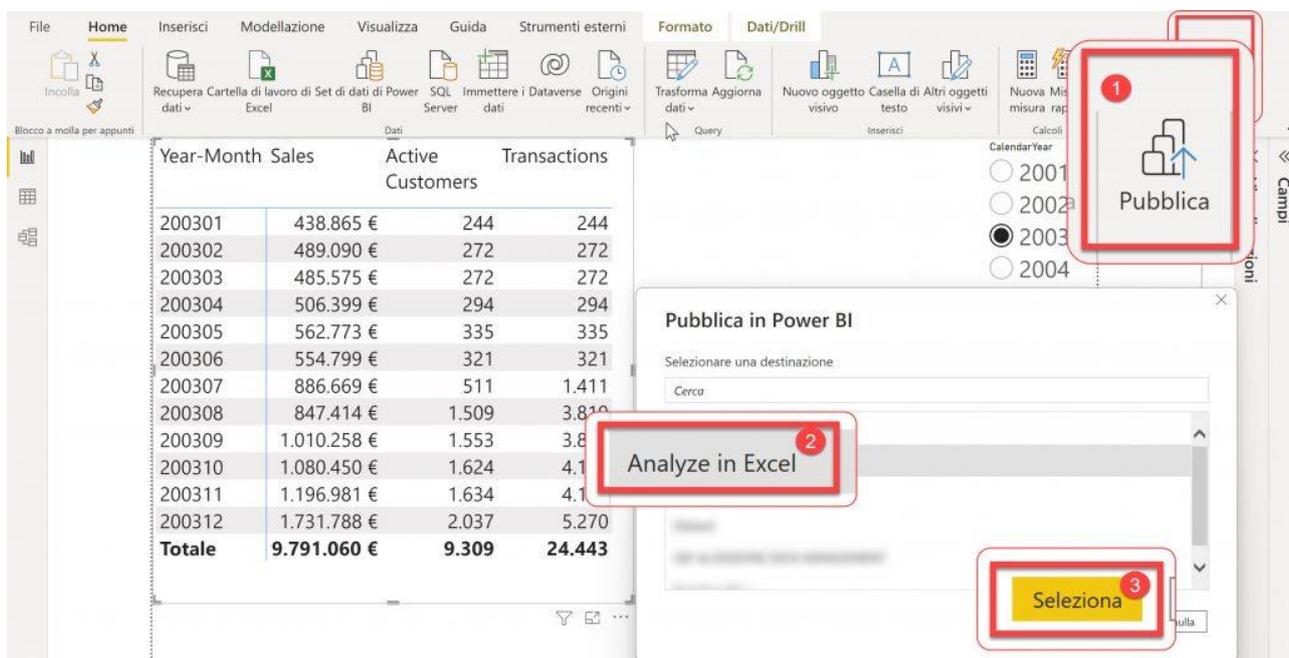


Figura 1

A questo punto, gli utenti con cui l'area di lavoro è stata condivisa potranno vedere il report su Power BI Cloud – attraverso qualunque browser –, secondo le regole della *Row-level security* eventualmente impostata.

Si supponga, tuttavia, che alcuni utenti non gradiscano usare Power BI Cloud per navigare i dati ma chiedano di restare in ambito Excel (scenario riferito al punto 1 dell'introduzione). Come fare?

Il primo modo è fare loro creare un nuovo file Excel e, da esso, farli connettere al *dataset* pubblicato su Power BI Cloud, previa autenticazione (figura 2).

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.

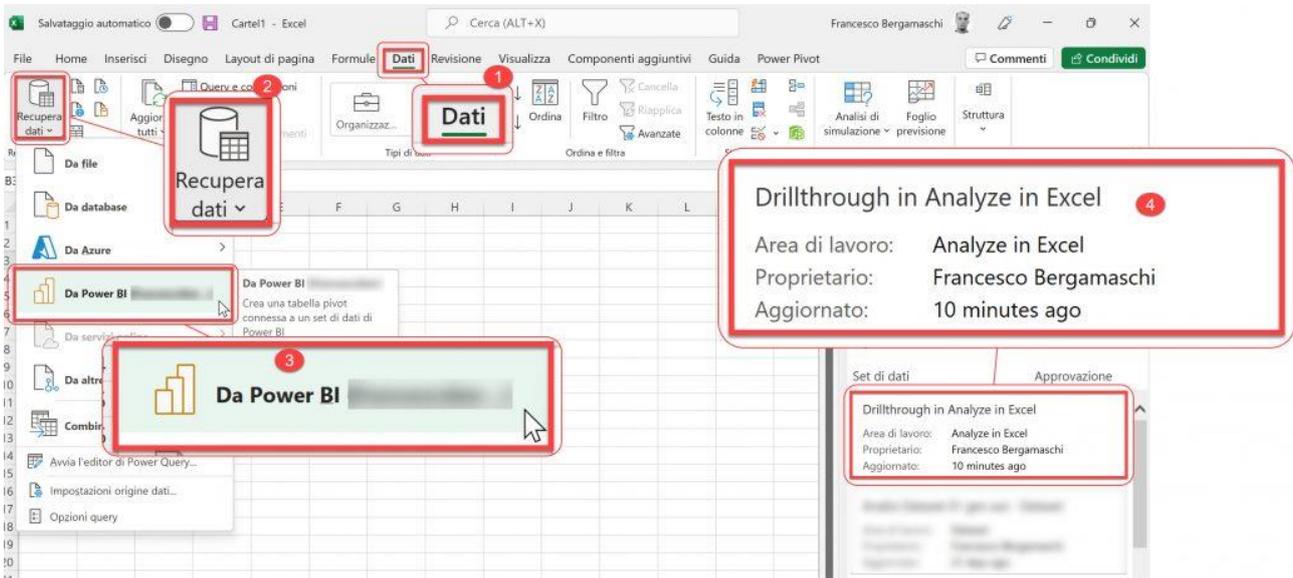


Figura 2

Alternativamente, è possibile scaricare da Power BI Cloud un file Excel già connesso al *dataset* Power BI pubblicato (figura 3) e spedirlo agli utenti via mail (il file Excel è leggero visto che contiene soltanto una stringa di connessione al *dataset*, i dati sono e restano sul cloud) o renderlo loro disponibile su una cartella condivisa. Il file Excel chiederà le credenziali una volta aperto.

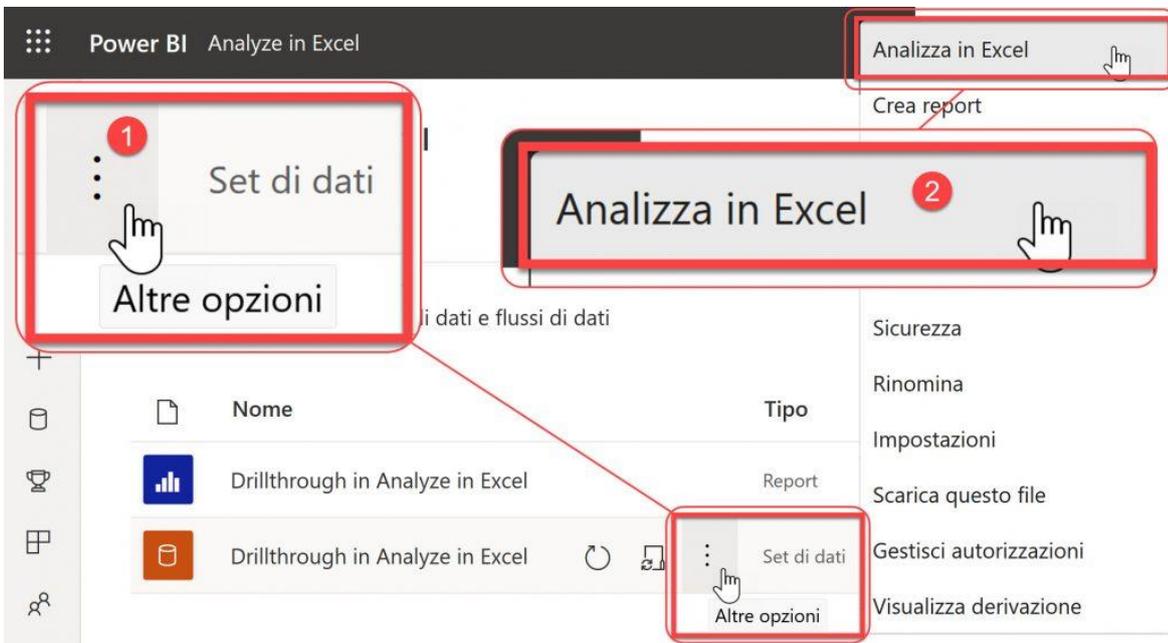


Figura 3

3

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 ogni caso, dunque, gli utenti che preferiscono interagire con il *dataset* Power BI usando Excel otterranno un file Excel con una tabella pivot pronta all'uso e replicabile in altri fogli di lavoro (figura 4).

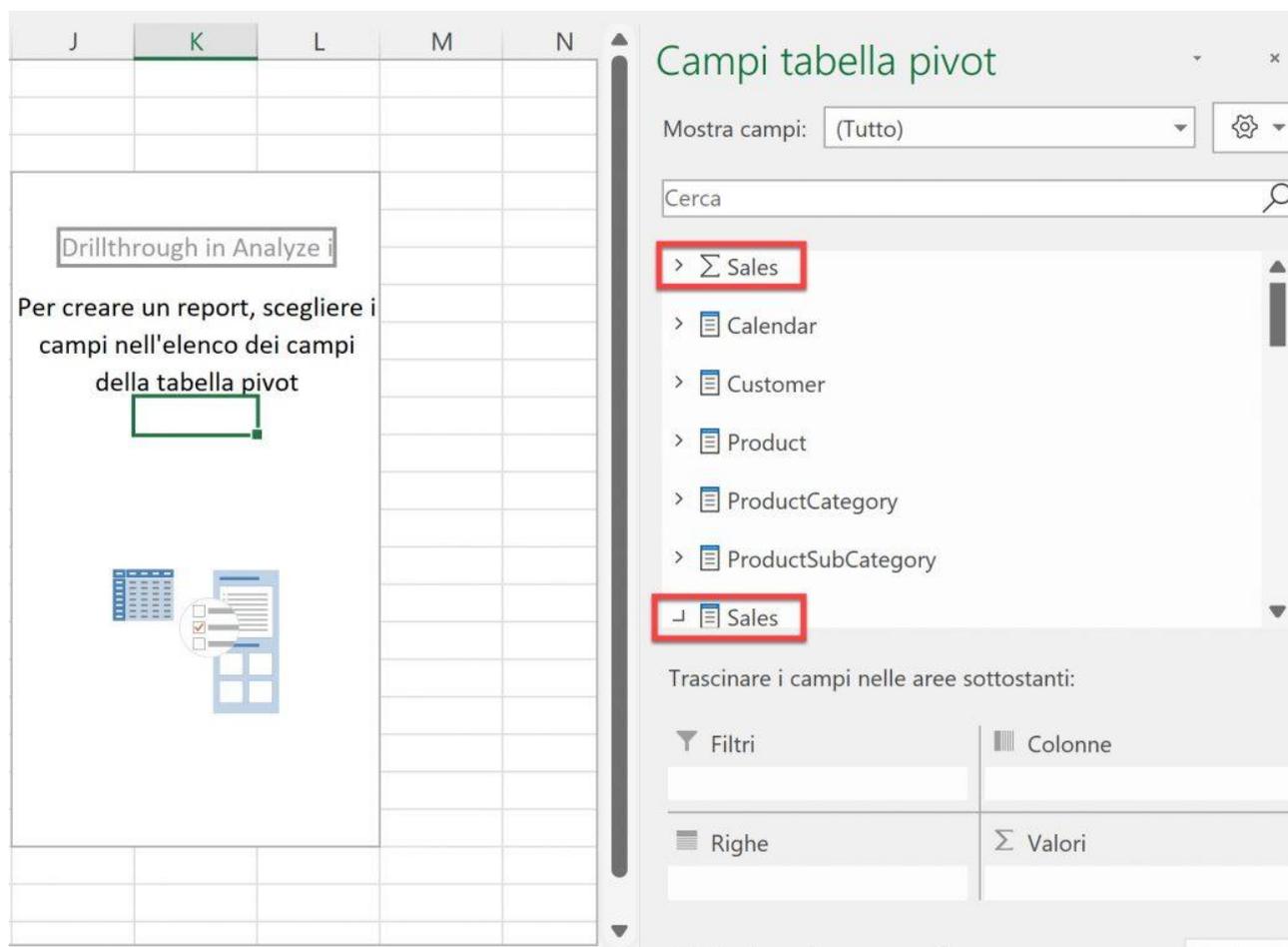


Figura 4

Si noti, in figura 4, che la tabella *Sales* è riportata due volte, in quanto essa contiene misure. Excel separa automaticamente le misure e le colonne, replicando di fatto la tabella. Ciò non avviene sulle tabelle che non contengono misure. Inoltre, nella zona *Valori* della pivot potranno essere mostrate soltanto misure *esplicite* (scritte in DAX nel progetto Power BI Desktop), anche se al momento della scrittura di questo articolo circola voce che Microsoft annuncerà a breve la possibilità di aggregare colonne come si fa nelle pivot classiche – tale modalità, tuttavia, è fortemente sconsigliata.

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.

Il file è persistente e, dunque, può essere usato indefinitamente, basta che il *dataset* esista sul cloud di Power BI e che si disponga sia della licenza pro che di una connessione ad Internet. Gli utenti potranno, dunque, creare il layout che preferiscono in termini di tabella pivot o pivotchart, filtri visivi e così via (figura 5).



Figura 5

Che succede, tuttavia, quando un utente effettua un doppio click su un numero della pivot? In figura 5 è evidenziato il valore della misura *Sales* per il mese di maggio 2003 (indicato in pivot come 200305), pari a 264.902 €. Il risultato è mostrato in figura 6.

Dati restituiti per Sales, 2003 - 200305, [Product].[Color].&[Black] (prime 1000 righe).

Sales[OrderDateKey]	Sales[DueDateKey]	Sales[ShipDateKey]	Sales[PromotionKey]	Sales[CurrencyKey]	Sales[SalesOrderNumber]	Sales
20030511	20030523	20030518	2	6	S050447	
20030522	20030603	20030529	2	6	S050544	
20030527	20030608	20030603	2	6	S050602	
20030515	20030527	20030522	2	6	S050489	
20030523	20030604	20030530	2	6	S050560	
20030506	20030518	20030513	2	6	S050396	
20030504	20030516	20030511	2	6	S050368	
20030528	20030609	20030604	2	6	S050611	
20030530	20030611	20030606	2	6	S050634	
20030516	20030528	20030523	2	100	S050493	
20030502	20030514	20030509	2	100	S050336	
20030527	20030608	20030603	2	100	S050596	
20030526	20030607	20030602	2	100	S050588	
20030514	20030526	20030521	2	100	S050474	
20030519	20030531	20030526	2	100	S050513	

Figura 6

Si notano, in figura 6, due aspetti da migliorare:

1 – il numero di righe estratto è limitato a 1.000, un valore piuttosto ristretto;

2 – vengono visualizzate tutte le colonne della tabella su cui la misura *Sales* è stata definita (tabella *Sales* in questo caso). Tale numero di colonne può essere elevato e alcune colonne possono contenere informazioni poco utili o incomprensibili per chi ha effettuato il *drillthrough*. Potrebbe essere utile, dunque, modificare l'elenco delle colonne e aggiungere, magari, qualche misura calcolata riga per riga.

Nel prosieguo dell'articolo, dunque, verranno migliorati entrambi gli aspetti sopra riportati.

Per prima cosa, sul primo aspetto, ecco come incrementare il numero di righe estratto, per quanto resta il limite di Excel di 1.048.576 righe. Per la funzione *drillthrough*, tuttavia, tale valore è, nella maggior parte dei casi, ampiamente sufficiente. Il numero di righe estratto è modificabile come segue (figure 7 e 8).

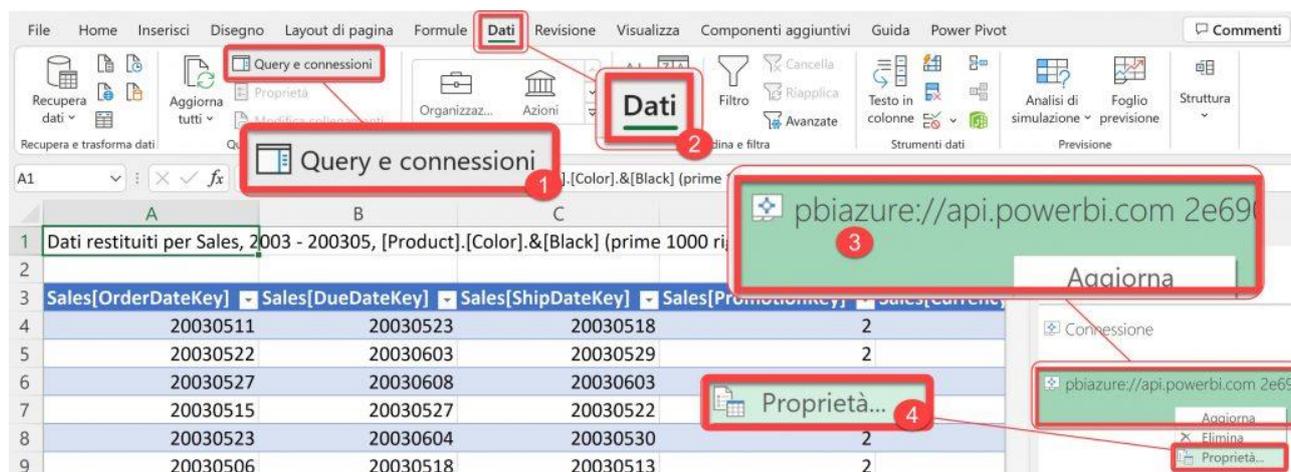


Figura 7

Si noti, in figura 7, esistono due connessioni su cui si potrebbe svolgere il passo 3, una denominata *pbiazure://api.powerbi.com...* e una denominata *Connessione*. Quest'ultima non permette la modifica del numero di righe, essendo la connessione relativa al *drillthrough* già svolto (mostrato in 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.

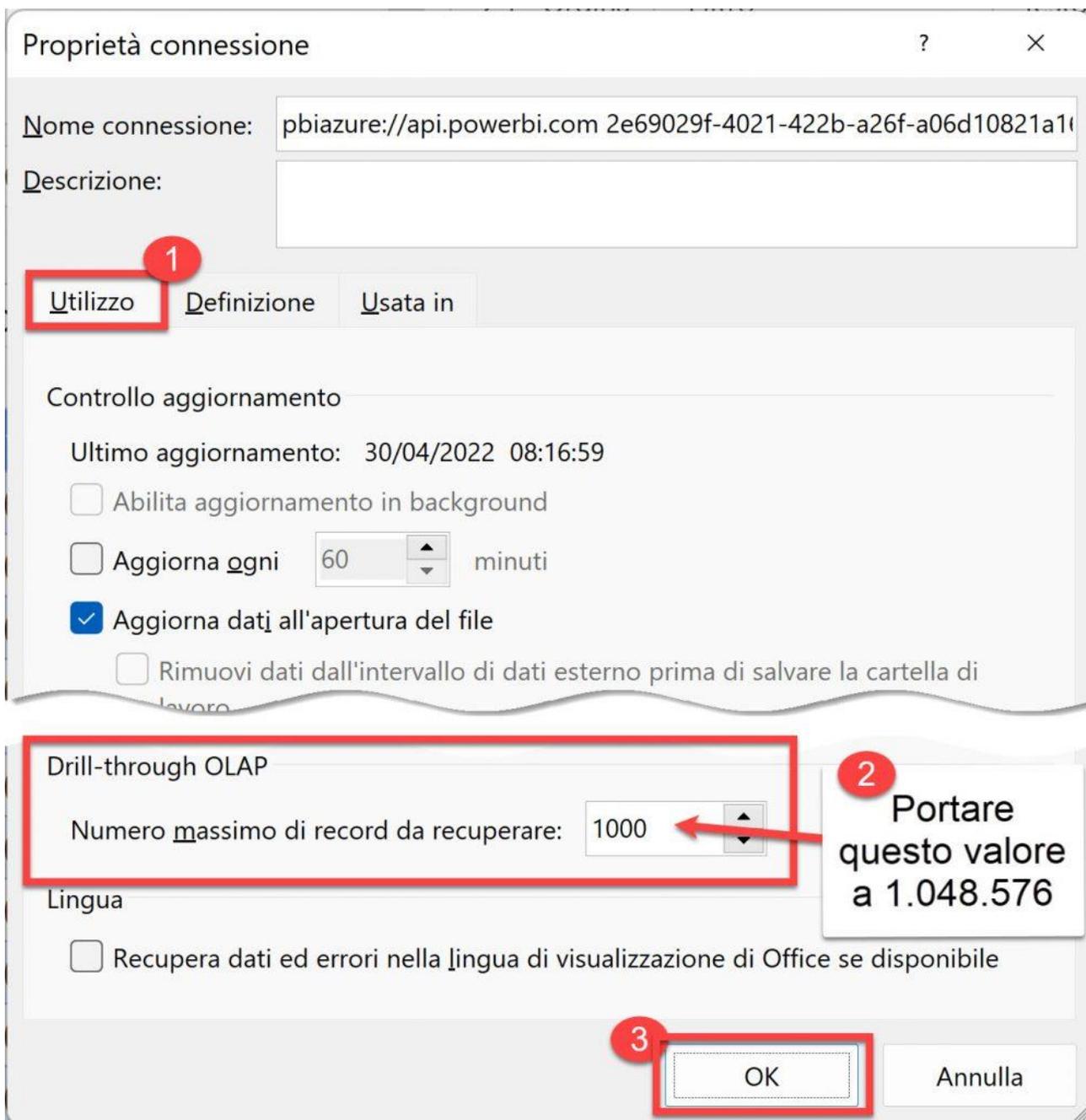


Figura 8

Riprovando il drillthrough sulla stessa cella, adesso vengono estratte fino a 1.048.576 righe (figura 9).

Dati restituiti per Sales, 2003 - 200305, Black (prime 1048576 righe).

Sales[OrderDateKey]	Sales[DueDateKey]	Sales[OrderDateKey]	Sales[DueDateKey]	Sales[OrderDateKey]	Sales[DueDateKey]
20030511	20030603	20030529	20030603	20030529	20030603
20030522	20030603	20030529	20030603	20030529	20030603
20030527	20030608	20030603	20030603	20030603	20030603
20030515	20030527	20030522	20030522	20030522	20030522
20030523	20030604	20030530	20030530	20030530	20030530
20030506	20030518	20030513	20030513	20030513	20030513
20030504	20030516	20030511	20030511	20030511	20030511
20030528	20030609	20030604	20030604	20030604	20030604
20030530	20030611	20030606	20030606	20030606	20030606
20030516	20030528	20030523	20030523	20030523	20030523
20030502	20030514	20030509	20030509	20030509	20030509
20030527	20030608	20030603	20030603	20030603	20030603
20030526	20030607	20030602	20030602	20030602	20030602
20030514	20030526	20030521	20030521	20030521	20030521
20030519	20030531	20030526	20030526	20030526	20030526
20030506	20030518	20030513	20030513	20030513	20030513
20030522	20030603	20030529	20030529	20030529	20030529
20030508	20030520	20030515	20030515	20030515	20030515
20030524	20030605	20030531	20030531	20030531	20030531
20030504	20030516	20030511	20030511	20030511	20030511
20030531	20030612	20030607	20030607	20030607	20030607

Foglio1 Foglio2 Drillthrough in Analyze in Exce

Figura 9

Sul secondo aspetto da migliorare (la selezione delle colonne visualizzate e, possibilmente, l'aggiunta di qualche misura), si parte, invece, dal progetto in Power BI Desktop. Da esso, lanciando lo strumento esterno Tabular Editor, si può modificare la *Default Detail Rows Expression* di ogni tabella e, se necessario, la *Detail Rows Expression* di ogni misura sulla stessa tabella. In generale, la *Default Detail Rows Expression* di ogni tabella sarà applicata ad ogni misura che giace sulla tabella stessa, tuttavia, se è definita anche una *Detail Rows Expression* a livello di misura, questa sovrascrive quella della tabella su cui la misura è definita. Le misure della stessa tabella che non hanno una *Detail Rows Expression*, invece, seguiranno quella presente al livello di tabella (*Default*). Se la *Default Detail Rows*

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.

Expression non è definita, essa conterrà tutte le colonne della tabella su cui la misura è definita, come già accennato.

Una volta lanciato Tabular Editor da Power BI Desktop (dal menu *Strumenti esterni*) o autonomamente, connettendosi poi al *data model* del file Power BI Desktop in uso), in figura 10 è mostrato come procedere per modificare la *Default Detail Rows Expression* della tabella *Sales*, in quanto essa ospita le due misure del *report* mostrato in figura 5 (*Sales* e *Active Customers*). Si suppone, in figura 10, che il consumatore del *report* indichi di avere bisogno dell'ID del cliente, del relativo nome e cognome, della data di nascita e del genere (questi ultimi campi della tabella *Customer*, diversa dalla tabella che ospita la misura del *report*, *Sales*), nonché del valore della misura *Sales*, riga per riga. La *Default Detail Rows Expression* è una espressione tabellare, dunque il DAX ci aiuta ad ottenere quanto richiesto.

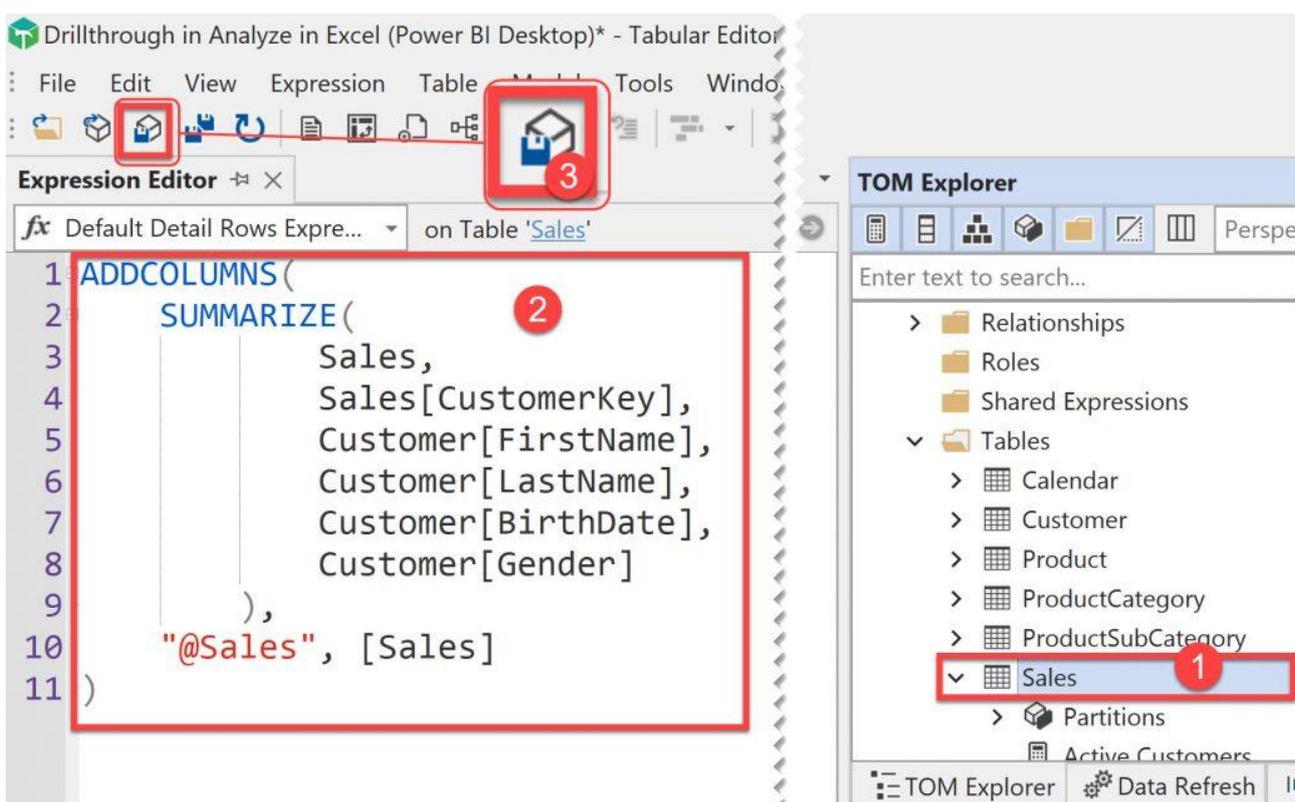


Figura 10

Tornati su Power BI Desktop, adesso sarà sufficiente ripubblicare il modello e fare *refresh* da Excel, per poi rilanciare il *drillthrough* ottenendo non solo fino a

1.048.576 righe, ma anche le colonne richieste e la misura indicata riga per riga (figura 11).

Dati restituiti per Sales, 2003 - 200305, Black (prime 1048576 righe)

Sales[CustomerKey]	Customer[FirstName]	Customer[LastName]	Customer[BirthDate]	Customer[Gender]	[@Sales]
13927	Gabriel	Gonzales	08/12/1963	M	2181,5625
15904	Janice	Bennett	25/01/1950	F	782,99
15908	David	Wilson	20/09/1951	M	782,99
15941	Bailey	Brooks			782,99
27260	Candice	Ma	05/03/1970	F	2049,0982
27268	Latasha	Serrano	15/02/1969	F	2049,0982
27276	Faith	Alexander	16/01/1969	F	2049,0982
27280	Naomi	Gutierrez	07/03/1968	F	2049,0982
27288	Destiny	Richardson	21/09/1970	F	2049,0982
27425	Steven	Morris	09/07/1949	M	2049,0982
27446	Tyler	Jones	08/06/1969	M	2049,0982
27263	Aaron	Kumar	28/07/1970	M	2049,0982
13912	Caroline	Simmons	20/03/1965	F	2181,5625
13939	Eduardo	Ramirez	20/03/1967	M	2181,5625
15890	James	Russell	27/02/1947	M	782,99
15948	Angela	Hughes	20/11/1964	F	782,99
27424	Luis	Mitchell	11/04/1949	M	2049,0982
15885	Garrett	Sanchez	18/05/1946	M	782,99

(prime 1048576 righe)

Figura 11

Gli ultimi aspetti da considerare sono:

1 – che fare se, su una misura che giace nella tabella *Sales*, si volesse avere un elenco di colonne e misure diverso da quello della *Default Detail Rows Expression* della tabella stessa?

2 – come generare misure che invitino l'utente al *drillthrough*, in modo da sfruttare questa funzionalità appieno e fare percepire la customizzazione?

Partiamo dal primo punto appena elencato. Consideriamo la misura *Active Customers* per la quale si voglia modificare la lista le colonne visualizzare nel *drillthrough* rispetto alla tabella *Sales*, per esempio includendo il nome del prodotto e rimuovendo il nome ed il cognome del cliente. Per prima cosa, una volta rilanciato Tabular Editor, si deve selezionare la misura e passare dall'espressione della misura stessa (non presente in una tabella importata come *Sales*) alla relativa *Detail Rows Expression* (figura 12), ancora una volta un'espressione tabellare, per poi immettere la *Detail Rows Expression* stessa (figura 13).

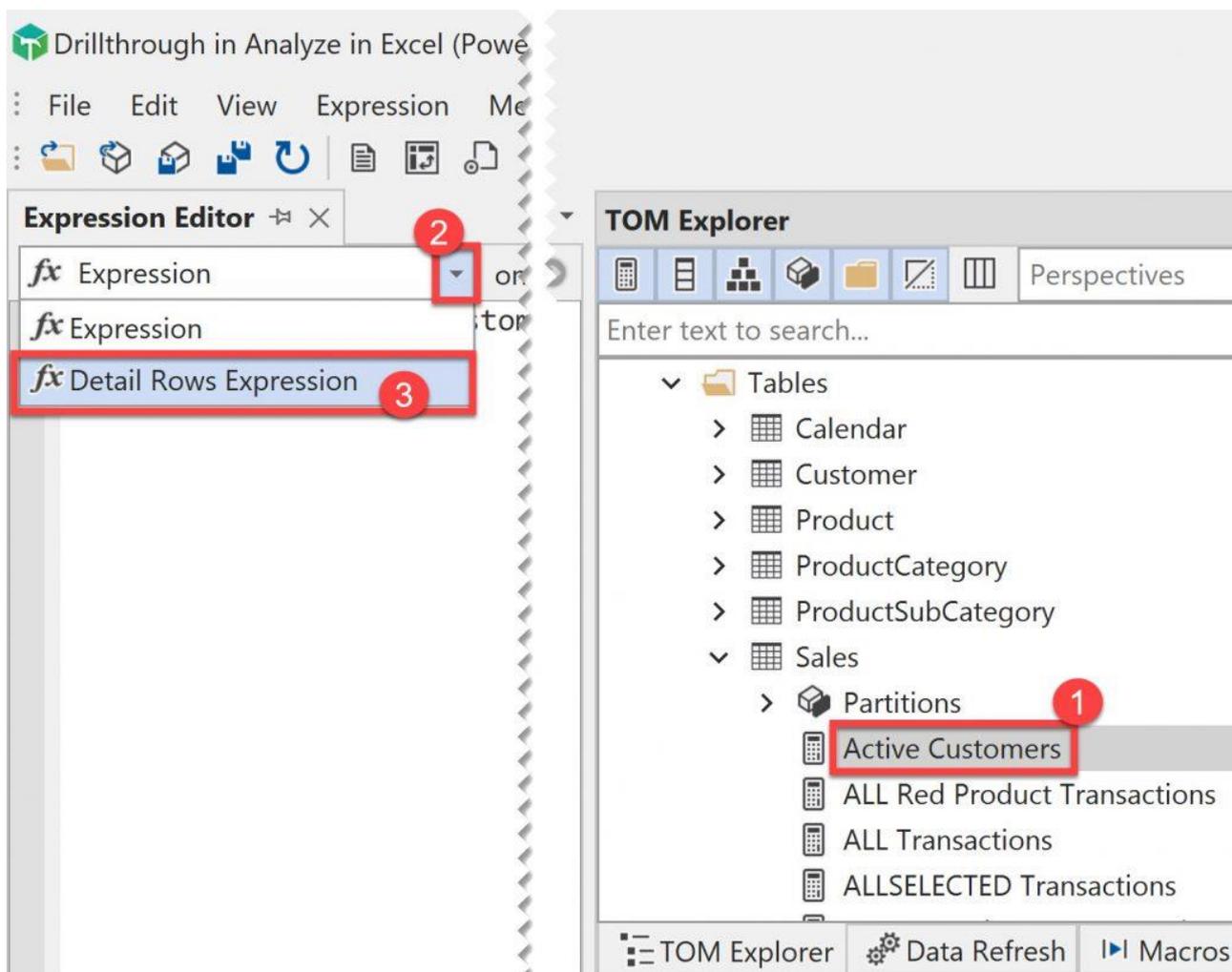


Figura 12

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.

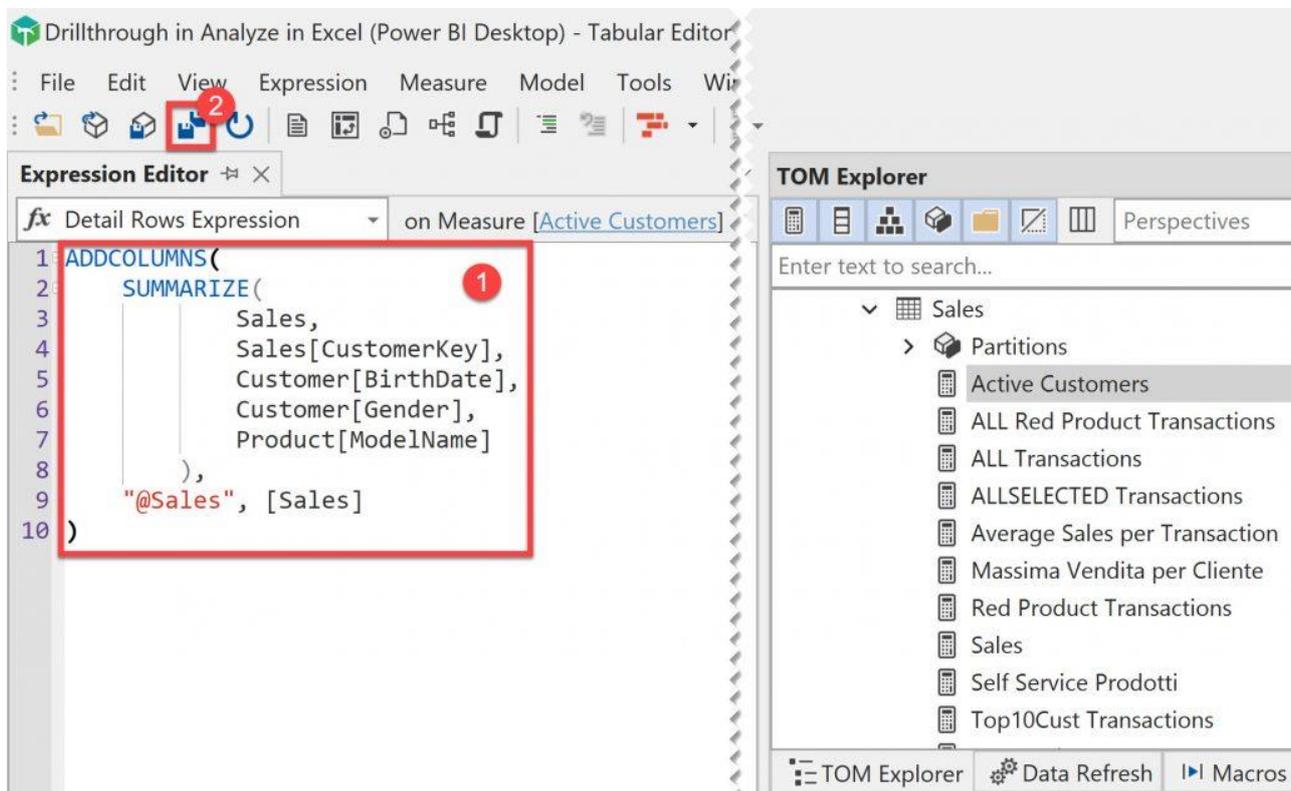


Figura 13

Tornati su Power BI Desktop, adesso sarà sufficiente pubblicare nuovamente il modello e fare *refresh* da Excel, per poi rilanciare il *drillthrough* ottenendo quanto richiesto (figura 14, dove si fa riferimento al doppio click sul dato di maggio 2003, per la misura *Active Customers*, valore 149 – si faccia riferimento alla 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.

Dati restituiti per Active Customers, 2003 - 200305, Black (prime 1048576 righe).

Sales[CustomerKey]	Customer[BirthDate]	Customer[Gender]	Product[ModelName]	[@Sales]
13927	08/12/1963	M	Road-250	2181,5625
15904	25/01/1950	F	Road-650	782,99
15908	20/09/1951	M	Road-650	782,99
15941	26/02/1965	F	Road-650	782,99
27260	05/03/1970	F	Mountain-200	2049,0982
27268	15/02/1969	F	Mountain-200	2049,0982
27276	16/01/1969	F	Mountain-200	2049,0982
27280	07/03/1968	F	Mountain-200	2049,0982
27288	21/09/1970	F	Mountain-200	2049,0982
27425	09/07/1949	M	Mountain-200	2049,0982
27446	08/06/1969	M	Mountain-200	2049,0982
27263	28/07/1970	M	Mountain-200	2049,0982
13912	20/03/1965	F	Road-250	2181,5625
13939	20/03/1967	M	Road-250	2181,5625
15890	27/02/1947	M	Road-650	782,99
15948	20/11/1964	F	Road-650	782,99
27424	11/04/1949	M	Mountain-200	2049,0982
15885	18/05/1946	M	Road-650	782,99
15888	05/05/1946	F	Road-650	782,99
15893	15/11/1948	F	Road-650	782,99

Figura 14

Infine, sul secondo punto sopra listato: la flessibilità del drillthrough crea l'opportunità di creare misure che invitino al drillthrough e il cui solo scopo sia principalmente, se non unicamente, il *drillthrough* stesso.

Per mostrare nella pratica questo aspetto, si può immaginare di creare una misura *Self-service Products* (diversa dall'eventuale misura *Products* già presente nel modello in modo da averle entrambe, si ricorda che il *drillthrough* funziona soltanto in Excel), con il seguente codice:

Self Service Products =

```
VAR Prodotti =
    DISTINCTCOUNT ( Sales[ProductKey] )
RETURN
    IF (
```

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.

```
[Sales] > 0,
IF (
  Prodotti > 1,
  "Clicca per dettaglio dei " & DISTINCTCOUNT ( Sales[ProductKey] ) & " prodotti"
),
IF (
  Prodotti = 1,
  "Clicca per dettaglio di " & DISTINCTCOUNT ( Sales[ProductKey] ) & " prodotto"
)
)
```

Una volta ripubblicato il modello (se la misura non era già presente in esso), la misura può essere inclusa nella pivot in Excel (figura 15) dando una percezione molto forte di customizzazione.



Figura 15

Come già mostrato, per questa misura, che giace sulla tabella *Sales*, è possibile customizzare la *Detail Rows Expression* (figura 16) per ottenere, per esempio, l'ID del prodotto, il relativo colore, peso e nome della sottocategoria e categoria.

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.

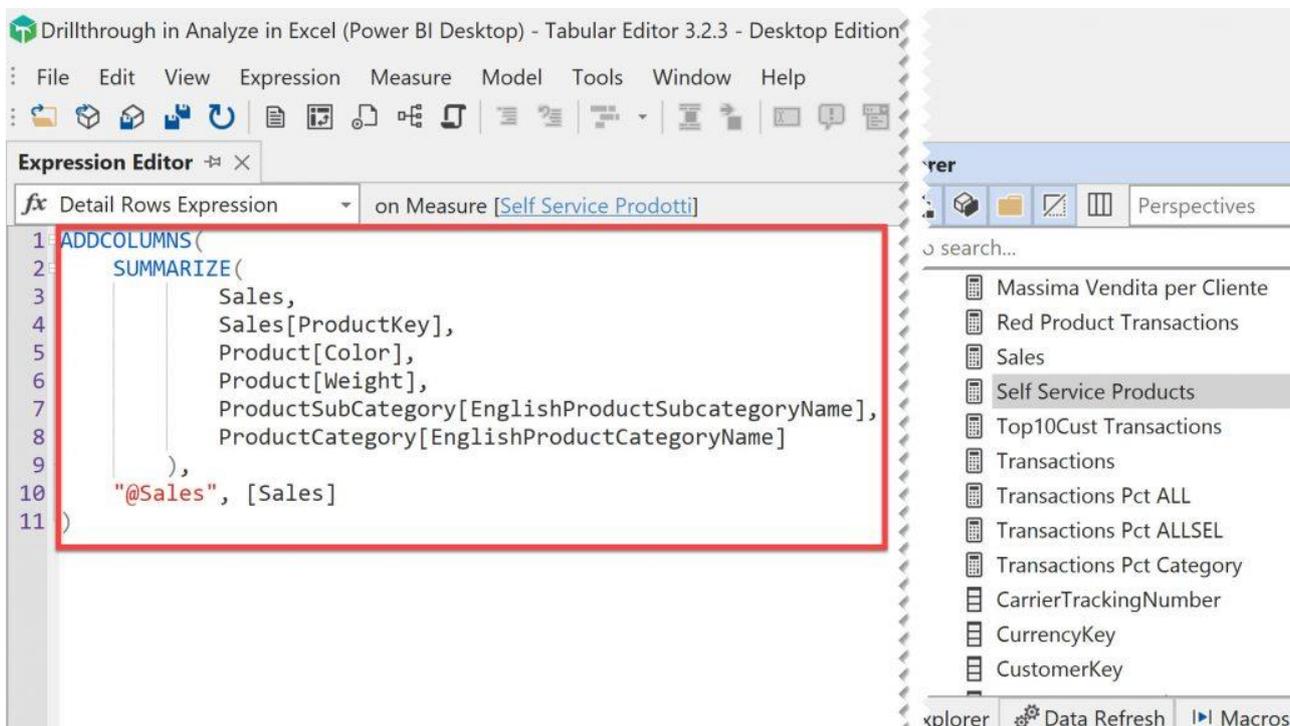


Figura 16

Ecco, infine, l'esperienza di *drillthrough* per il dato di maggio 2003 della misura *Self-service Products* (figura 17).

Sales[ProductKey]	Product[Color]	Product[Weight]	ProductSubCategory[EnglishProductSubcategoryName]	ProductCategory[EnglishProductCategoryName]	@Sales
377	Black	15	Road Bikes	Bikes	28360,3125
341	Black	19	Road Bikes	Bikes	4697,94
333	Black	20	Road Bikes	Bikes	6263,92
339	Black	19	Road Bikes	Bikes	3131,96
358	Black	23	Mountain Bikes	Bikes	49178,3568
360	Black	24	Mountain Bikes	Bikes	53276,5532
375	Black	15	Road Bikes	Bikes	8726,25
343	Black	19	Road Bikes	Bikes	4697,94
335	Black	20	Road Bikes	Bikes	5480,93
337	Black	20	Road Bikes	Bikes	3914,95
373	Black	15	Road Bikes	Bikes	26178,75
379	Black	16	Road Bikes	Bikes	21815,625
362	Black	24	Mountain Bikes	Bikes	49178,3568

Figura 17

17

Conclusioni e Note

Power BI non sostituisce Excel ed Excel (base o Power Pivot) non è un'alternativa a Power BI. Excel e Power BI possono, invece, essere strumenti ottimamente integrati tra loro, ognuno per il proprio scopo. Power BI può essere sia il tool di progetto che di consumo della BI, Excel può solo invece essere il tool di consumo

alternativo di un modello Power BI. La possibilità di costruire la BI con Excel (Power Pivot) non porta ad una soluzione completa di BI – che, per esserlo, deve prevedere la condivisione integrata che solo Power BI offre. Power Pivot resta, cioè, un buon strumento di esplorazione dati ma solo a scopo personale. Tuttavia, Excel resta il miglior strumento di inserimento dati al mondo, è molto semplice e immediato e viene percepito dagli utenti come un ambiente sicuro, noto ed ospitale. Questo aspetto psicologico va sfruttato per aiutare la transizione alla BI e in questo articolo è stato mostrato come fare fruire un modello Power BI attraverso Excel.

Note:

1 – Tabular Editor è disponibile in versione gratuita (versioni 2.xx, prive di *Intellisense*) e/o a pagamento (versioni 3.xx). Quanto mostrato in questo articolo è fattibile con entrambe le versioni, tuttavia nell'articolo sono mostrate le schermate di Tabular Editor 3.2.3. Una volta installato Tabular Editor, e chiuso Power BI Desktop, alla riapertura di Power BI Desktop verrà mostrato il menu *Strumenti esterni* da cui si potrà comodamente lanciare Tabular Editor direttamente connesso al modello Power BI Desktop in uso, ottenendo quanto mostrato nelle figure. Ecco i link per Tabular Editor 2.xx (<https://www.sqlbi.com/tools/tabular-editor/>) e Tabular Editor 3.xx (<https://tabulareditor.com/>);

2 – un'ulteriore applicazione di questo articolo è la generazione di query personalizzate in DAX che creino una tabella Excel eseguendo il comando inserito e che possono, poi, essere importate nel data model (il nome con cui questa funzionalità è nota è *linked-back tables* e disponibile in ambito Excel versione 2013 o successive). Le *linked-back tables* sono molto utili visto che in Excel Power Pivot, al contrario di Power BI, le tabelle calcolate non sono creabili. Le *linked-back tables* possono essere usate, per esempio, per generare dimensioni (anagrafiche) sulla base dei fatti – per quanto ciò abbia qualche limite e controindicazione – o tabelle *bridge*. Se la cosa è di interesse, invitiamo i lettori a lasciare commenti in modo da generare video e articoli al riguardo;

3 – nel caso si usi una tabella *Misure*, solitamente una tabella vuota creata ove vengano ospitate tutte le misure, il *drillthrough* sulle misure in essa ospitate restituirà una tabella vuota (si ricorda che il *drillthrough* lavora sulla tabella su cui le misure sono definite, ed essa è vuota in questo caso, e non sulle tabelle su cui le misure lavorano per fare i calcoli). Tuttavia, la *Detail Rows Expression* di una tabella può essere richiamata in un'altra tabella, ancora una volta invitiamo la community a manifestare interesse, nel caso ci sia, per questo tecnicismo, in modo da ovviare a questo aspetto.

[file .pbix](#) [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.