

Identificare gruppi di anni consecutivi di acquisto – tecnica statica

PUBBLICATO MAGGIO 29, 2023 DI FRANCESCO BERGAMASCHI

Per prendere decisioni di Marketing, è spesso necessario identificare i comportamenti dei clienti in fase di acquisto. In altri casi, si verifica la frequenza di acquisto, in altri gli acquisti consecutivi e così via.

In un recente piccolo progetto per un cliente, è emersa la richiesta di identificare e contare:

1. i gruppi di anni consecutivi in cui un cliente ha effettuato almeno un acquisto;
2. i clienti che hanno comprato consecutivamente per un certo numero di anni.

La richiesta può sembrare, in prima istanza, non complessa ma è una miopia: come vedremo, serve un DAX abbastanza avanzato.

In questo articolo ci occuperemo di risolvere il problema in modo statico, cioè usando una colonna calcolata – dove risiederà la maggior parte della complessità – e alcune misure. Se l'argomento sarà di interesse, pubblicheremo anche la versione dinamica, dove tutto è risolto con delle misure. La versione statica, infatti, presenta dei limiti che vengono superati dalla versione dinamica.

Il modello dati è visibile in figura 1, un classico schema a stella.

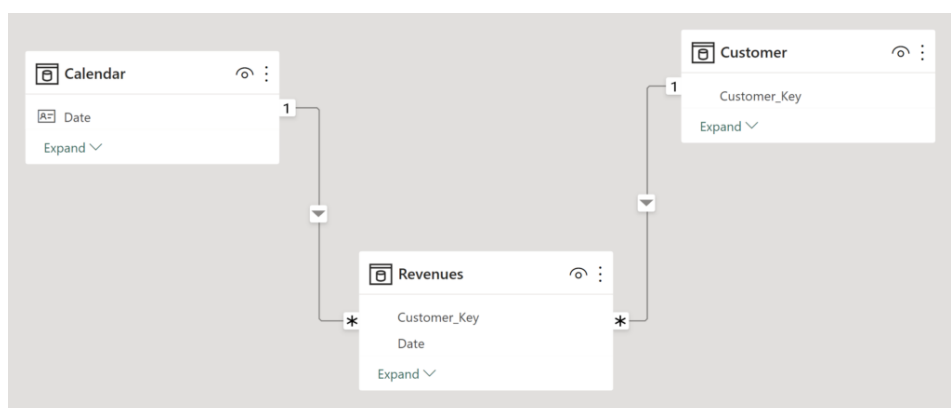


Figura 1

La tabella *Revenues* è una tabella di transazioni, mostrata in figura 2. Le altre due tabelle (*Customer* e *Calendar*) sono anagrafiche dotate di chiave primaria.

Date	Customer_Key	Total_Revenue
01/gen/2020	1	134
01/gen/2021	1	102
01/gen/2022	1	124
01/gen/2024	1	190
01/gen/2025	1	100
01/gen/2023	2	166
01/gen/2024	2	151
01/gen/2021	3	118
01/gen/2023	3	123
01/gen/2025	3	154
01/gen/2027	1	134
01/gen/2028	1	102
01/gen/2029	1	124

Figura 2

Il report che si vuole ottenere è mostrato in figura 3.

Customer_Key	Consecutive Years Col	Number of Events Static	Consecutive Years Col	Number of Customers Static	Number of Customers Static Pct
1	3	2	1	3	50,0%
1	2	1	2	2	33,3%
2	2	1	3	1	16,7%
3	1	3	Total	6	100,0%
Total	7				

Figura 3

Nella matrice a sinistra in figura 3 sono mostrati, cliente per cliente (*Customer_Key*), i gruppi di anni consecutivi di acquisto (*Consecutive Years Col*) e il numero di ognuno di questi gruppi (*Number of Events Static*). Per il cliente 1, per esempio, sono mostrate

due righe evidenziate in rosso: nella prima si nota che il cliente 1 ha due gruppi di 3 anni consecutivi (in figura 2 si può notare che essi sono 2020, 2021 e 2022 come primo gruppo e 2027, 2028 e 2029 come secondo), nella seconda si nota che il cliente 1 ha un gruppo di due anni consecutivi (sempre in figura 2 si può notare che essi sono il 2024 e il 2025).

Nella matrice a destra in figura 3 sono mostrati, invece, gruppo di anni consecutivi per gruppo di anni consecutivi (1, 2 o 3 visti i dati), quanti clienti, in assoluto e in percentuale sul totale, hanno avuto almeno un gruppo di anni consecutivi corrispondente (*Number of Customers Static* e *Number of Customers Static Pct*). Per esempio, nella riga evidenziata in rosso, si nota che due clienti (in figura 2 si può notare che si tratta del cliente 1 negli anni 2024 e 2025 e il cliente 2 negli anni 2023 e 2024) hanno avuto almeno un gruppo di due anni consecutivi di acquisto. Il valore pari a 6 al totale per la misura *Number of Customers Static* necessita di un chiarimento: la richiesta del cliente era esplicitamente di considerare come cliente ogni coppia *cliente - gruppo di anni consecutivi*. Il numero di clienti distinti è, in realtà, pari a 3.

I limiti della tecnica statica che qui mostreremo risiedono nella non reattività del report in figura 3 ad eventuali filtri applicati.

Sviluppo

Volendo risolvere in maniera statica il problema, si ricorre ad una colonna calcolata della tabella *Revenues*: *Revenues[Consecutive Years Col]*. Per comodità, si è creata anche una colonna *Revenues[Year]*.

La colonna *Revenues[Consecutive Years Col]* deve indicare, al termine di un blocco consecutivo di anni di acquisto di un cliente, il numero di anni consecutivi (figura 4 in cui sono evidenziati i blocchi del cliente 1).

Date	Customer_Key	Total_Revenue	Year	Consecutive Years Col
01/gen/2020	1	134	2020	
01/gen/2021	1	102	2021	
01/gen/2022	1	124	2022	3
01/gen/2024	1	190	2024	
01/gen/2025	1	100	2025	2
01/gen/2023	2	166	2023	
01/gen/2024	2	151	2024	2
01/gen/2021	3	118	2021	1
01/gen/2023	3	123	2023	1
01/gen/2025	3	154	2025	1
01/gen/2027	1	134	2027	
01/gen/2028	1	102	2028	
01/gen/2029	1	124	2029	3

Figura 4

Ecco il codice della colonna *Revenues[Year]* e della colonna più importante per la soluzione statica, *Revenues[Consecutive Years Col]*:

Year =

`YEAR ( Revenues[Date] )`

Consecutive Years Col =

`VAR CurrentCust = Revenues[Customer_Key]`

`VAR CurrentDate = Revenues[Date]`

`VAR ConsYearTab =`

`ADDCOLUMNS (`

`ADDCOLUMNS (`

`FILTER ( Revenues, Revenues[Customer_Key] = CurrentCust ),`

`"@NearestLowerHole",`

`VAR RefYear = Revenues[Year]`

`VAR AllYears =`

`DISTINCT (`

`UNION (`

`ALL ( Revenues[Year] ),`

`GENERATESERIES ( MIN ( Revenues[Year] ) - 1, MAX ( Revenues[Year]`

`ar] ) + 1, 1 )`

`)`

`)`

`VAR YearsWithSales =`

`CALCULATETABLE (`

`VALUES ( Revenues[Year] ),`

```

        Revenues[Customer_Key] = CurrentCust,
        REMOVEFILTERS ( Revenues )
    )
    VAR YearsWithNoSales =
        EXCEPT ( AllYears, YearsWithSales )
    VAR NearestHole =
        MAXX ( FILTER ( YearsWithNoSales, Revenues[Year] <= RefYear ), Reven
ues[Year] )
    RETURN
        NearestHole,
        "@NearestHigherHole",
        VAR RefYear = Revenues[Year]
        VAR AllYears =
            DISTINCT (
                UNION (
                    ALL ( Revenues[Year] ),
                    GENERATESERIES ( MIN ( Revenues[Year] ) - 1, MAX ( Revenues[Ye
ar] ) + 1, 1 )
                )
            )
        VAR YearsWithSales =
            CALCULATETABLE (
                VALUES ( Revenues[Year] ),
                Revenues[Customer_Key] = CurrentCust,
                REMOVEFILTERS ( Revenues )
            )
        VAR YearsWithNoSales =
            EXCEPT ( AllYears, YearsWithSales )
        VAR NearestHole =
            MINX ( FILTER ( YearsWithNoSales, Revenues[Year] > RefYear ), Revenu
es[Year] )
        RETURN
            NearestHole
    ),
    "@ConsYears", Revenues[Year] - [@NearestLowerHole]
)
VAR ConsYearTabWithMax =
    ADDCOLUMNS (
        ConsYearTab,
        "@MaxConsYears",
        VAR CurrentLowerHole = [@NearestLowerHole]
        VAR CurrentHigherHole = [@NearestHigherHole]

```

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.

```

RETURN
  MAXX (
    FILTER (
      ConsYearTab,
      Revenues[Year] >= CurrentLowerHole
      && Revenues[Year] < CurrentHigherHole
    ),
    [@ConsYears]
  )
)
VAR CurrentCustDateConsYears =
  SELECTCOLUMNS (
    FILTER ( ConsYearTabWithMax, Revenues[Date] = CurrentDate ),
    "@@ConsYears", [@ConsYears]
  )
)
VAR MaxConsYearsCurrentCust =
  SELECTCOLUMNS (
    FILTER ( ConsYearTabWithMax, Revenues[Date] = CurrentDate ),
    "@@MaxConsYears", [@MaxConsYears]
  )
)
RETURN
  IF (
    CurrentCustDateConsYears = MaxConsYearsCurrentCust,
    CurrentCustDateConsYears
  )

```

Come si nota, il codice di *Revenues[Consecutive Years Col]* è tutt'altro che banale.

Occupiamoci di una breve illustrazione del codice della colonna calcolata *Revenues[Consecutive Years Col]*:

- per prima cosa, il codice conserva in due variabili (*CurrentCust* e *CurrentDate*) la data e il cliente della riga corrente ;
- successivamente viene creata una tabella in memoria, *ConsYearTab*, in cui, riga per riga sulla tabella stessa, preselezionata per il cliente corrente tramite *FILTER*, si calcolano tre valori: l'anno senza acquisti più vicino, sia nel passato che nel futuro, rispetto all'anno della transazione considerata ([@NearestLowerHole] e [@NearestHigherHole]) e il numero di anni consecutivi ([@ConsYears]) come differenza *Revenues[Year]* e [@NearestLowerHole];

- successivamente viene creata un'altra tabella in memoria, *ConsYearTabWithMax*, che non è altro che la tabella *ConsYearTab* con l'aggiunta di una colonna calcolata (*[@MaxConsYears]*) che calcola il massimo dei valori della colonna *[@ConsYears]* con il vincolo di considerare soltanto i valori di *Revenues[Year]* che siano maggiori o uguali a *[@NearestLowerHole]* e minori di *[@NearestHigherHole]*;
- successivamente, viene selezionata la colonna *[@ConsYears]* di *ConsYearTabWithMax*, vincolando la data ad essere quella della riga correntemente iterata, per avere il valore di anni consecutivi alla riga corrente. Tale colonna ha una sola riga – si ricordi che il cliente è stato già selezionato in *ConsYearTab* – e si ottiene così un valore scalare, *CurrentCustDateConsYears*;
- successivamente viene selezionata la colonna *[@MaxConsYears]* di *ConsYearTabWithMax*, ancora vincolando la data ad essere quella della riga correntemente iterata, per avere il valore massimo degli anni consecutivi delle righe appartenenti al gruppo a cui appartiene la riga corrente – identificato dal cliente corrente e dai valori di *[@NearestLowerHole]* e *[@NearestHigherHole]*. Nuovamente, tale colonna ha una sola riga – si ricordi che il cliente è stato già selezionato in *ConsYearTab* – e si ottiene così un valore scalare, *MaxConsYearsCurrentCust*;
- infine, se *CurrentCustDateConsYears* è pari a *MaxConsYearsCurrentCust*, viene restituito *CurrentCustDateConsYears*, in modo da avere il valore solo al termine del blocco di anni consecutivi corrente per il cliente corrente. Avere i valori anche nelle altre righe avrebbe dato problemi di raggruppamento nel report.

Le misure che servono per creare il report sono, infine (come da figura 3):

Number of Events Static =
COUNTROWS (Revenues)

Number of Customers Static =
SUMX (
 ADDCOLUMNS (
 VALUES (Revenues[Consecutive Years Col]),
 "@Num",
 VAR ConsYears = Revenues[Consecutive Years Col]
 RETURN
 CALCULATE (

```

DISTINCTCOUNT ( Revenues[Customer_Key] ),
Revenues[Consecutive Years Col] >= ConsYears
),
[ @Num]
)

```

```

Number of Customers Static Pct =
DIVIDE (
    [Number of Customers Static],
    CALCULATE (
        [Number of Customers Static],
        ALLSELECTED ( Revenues[Consecutive Years Col] )
    )
)

```

Illustriamo adesso i limiti di questa soluzione, che è perfetta fino a quando non si introducono filtri sugli anni da considerare. Si osservi la figura 5 in cui l'anno 2020 di calendario è stato escluso. Ci si aspetterebbe di vedere calare il numero di gruppi da tre anni del cliente 1 da 2 (figura 3) a 1 perché il gruppo di anni 2020, 2021 e 2022 adesso dovrebbe essere formato da due anni (2021 e 2022) e crescere il numero di gruppi da due anni del cliente 1 da 1 a 2. Tuttavia, essendo la soluzione statica, nulla cambia nel report.

Year							
☒	Select all						
☐	2020						
☒	2021						
☒	2022						
☒	2023						
☒	2024						
☒	2025						
☒	2026						
☒	2027						
☒	2028						
☒	2029						
		Customer _Key	Consecutive Years Col	Number of Events Static	Consecutive Years Col	Number of Customers Static	Number of Customers Static Pct
		1	3	2	1	3	50,0%
		1	2	1	2	2	33,3%
		2	2	1	3	1	16,7%
		3	1	3	Total	6	100,0%
		Total		7			

Figura 5

Conclusioni

Il report è complesso da realizzare, nonostante la richiesta che lo ha generato possa sembrare semplice a prima vista. La colonna calcolata *Revenues[Consecutive Years Coll]*, che rappresenta la chiave del problema, permette di raggruppare per numero di anni consecutivi ma, essendo una colonna calcolata, è statica e i suoi valori non sono sensibili ai filtri del report. Se si vuole ottenere interazione con il report bisogna andare verso una soluzione dinamica, basata esclusivamente su misure. Se ne riceveremo richiesta, mostreremo anche questa soluzione in un futuro articolo.

[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.