

Power BI

Giorno 4

Agenda

Cosa vedremo

- ❑ Contesto
- ❑ Override di contesto
- ❑ Funzioni Time-intelligence



Agenda

Obiettivi di apprendimento

- ☐ scrivere DAX avanzate
- ☐ implementare analisi nel tempo
- ☐ saper implementare filtri personalizzati in DAX

Advanced DAX

DAX Context



```
TotalSales =  
SUM(Sales[SalesAmount])
```

```
BikesTotalSales =  
CALCULATE(  
    SUM(Sales[SalesAmount])  
    , 'Product'[Category] = "Bikes")
```

La funzione ***CALCULATE*** è una delle funzioni DAX più importanti.

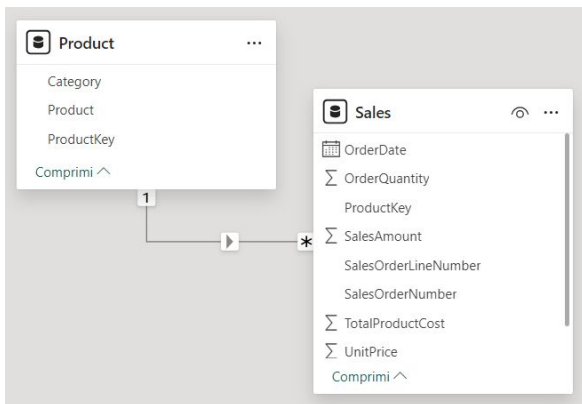
La funzione ***CALCULATE*** consente di fare 'override di contesto' applicando filtri 'custom' che rispondono a specifiche esigenze di analisi.

In altre parole, consente di modificare i filtri di contesto!

La sintassi generale è la seguente:

```
CALCULATE(<expression>, <filter1, ..>)
```

DAX Context



Category	TotalSales
Accessories	539.106,09 €
Bikes	64.069.033,16 €
Clothing	1.714.056,05 €
Components	11.226.374,90 €
Totale	77.548.570,20 €

Dato lo schema mostrato in figura, la relazione tra *Product* e *Sales* consente facilmente di ottenere la visualizzazione mostrata: la somma del campo *SalesAmount* per *Category*.

Supponiamo di dover calcolare l'incidenza percentuale, in termini di *SalesAmount*, della categoria *Bikes* sul totale.

In altre parole, è necessario rispondere alla domanda: quanto cubano i ricavi di *Bikes* sul totale?

DAX Context

"quanto cubano i ricavi di *Bikes* sul totale?"

Per ottenere il risultato richiesto è necessario eseguire questa divisione:

Somma di SalesAmount per la sola categoria Bikes
Somma di SalesAmount

La funzione **DIVIDE** richiede due argomenti obbligatori: il numeratore come primo e il denominatore come secondo!



```
1 SalesBikesPerc =  
2 DIVIDE(  
3     -- somma di SalesAmount solo quando Category è Bikes  
4     CALCULATE(  
5         SUM(Sales[SalesAmount])  
6         , 'Product'[Category] = "Bikes")  
7     -- somma del campo SalesAmount  
8     , SUM(Sales[SalesAmount])  
9 )
```

Category	TotalSales
Accessories	539.106,09 €
Bikes	64.069.033,16 €
Clothing	1.714.056,05 €
Components	11.226.374,90 €
Totale	77.548.570,20 €

82,62%
SalesBikesPerc

DAX Context Override, PARALLELPERIOD

Sales Analysis YoY



```
SalesPY =
CALCULATE(
    [TotalSales]
    , PARALLELPERIOD('Calendar'[Date], -12, MONTH))
```

La funzione **PARALLELPERIOD** è una funzione di tipo **Time-intelligence**.

Consente di valutare un'espressione in un periodo temporale parallelo rispetto a quello determinato dal contesto.

La funzione richiede 3 argomenti: il riferimento temporale (dimensione tempo), il numero di intervalli rispetto i quali andare a ritroso nel tempo (-) o in avanti (+), e la tipologia di intervallo.

DAX Context Override, PARALLELPERIOD

```
1 SalesPM =  
2 CALCULATE(  
3     [TotalSales]  
4     , PARALLELPERIOD('Calendar'[Date], -1, MONTH))
```

Sales Analysis MoM

Year	Month	TotalSales	SalesPM
2017	Jul	489.328,40 €	
2017	Aug	1.540.072,02 €	489.328,40 €
2017	Sep	1.166.332,45 €	1.540.072,02 €
2017	Oct	844.832,71 €	1.166.332,45 €
2017	Nov	2.325.754,91 €	844.832,71 €
2017	Dec	1.703.435,09 €	2.325.754,91 €
2018	Jan	713.229,54 €	1.703.435,09 €
2018	Feb	1.900.794,46 €	713.229,54 €
Totale		77.548.570,20 €	77.548.570,20 €

Nell'esempio precedente è stato analizzata una particolare metrica nello stesso periodo dell'anno precedente.

Allo stesso modo, è possibile utilizzare la funzione PARALLELPERIOD per valutare il mese precedente rispetto a quello determinato dal contesto.

Il secondo argomento specifica il numero di intervalli da sottrarre alla data determinata dal contesto (meno un mese).

Il terzo argomento specifica il tipo di intervallo rispetto al quale spostare le date.

DAX Context Override, PARALLELPERIOD

```
1 SalesMoM =  
2 DIVIDE( [TotalSales] - [SalesPM], [SalesPM])
```

Sales Analysis MoM

Year	Month	TotalSales	SalesPM	SalesMoM
2017	Jul	489.328,40 €		
2017	Aug	1.540.072,02 €	489.328,40 €	214,73%
2017	Sep	1.166.332,45 €	1.540.072,02 €	-24,27%
2017	Oct	844.832,71 €	1.166.332,45 €	-27,57%
2017	Nov	2.325.754,91 €	844.832,71 €	175,29%
2017	Dec	1.703.435,09 €	2.325.754,91 €	-26,76%

Questo approccio consente, ad esempio, di valutare gli scostamenti o variazioni percentuali rispetto al mese precedente.

"Abbiamo fatto meglio o peggio rispetto al mese precedente?"

"Rispetto al mese precedente di quanto (in termini percentuali) abbiamo fatto meglio o peggio?"

Queste sono due domande alle quali poter dare una risposta!

DAX Context Override, USERELATIONSHIP

Orders and Orders to ship

Year, Month

2020

☐ Jan

☒ Feb

☐ Mar

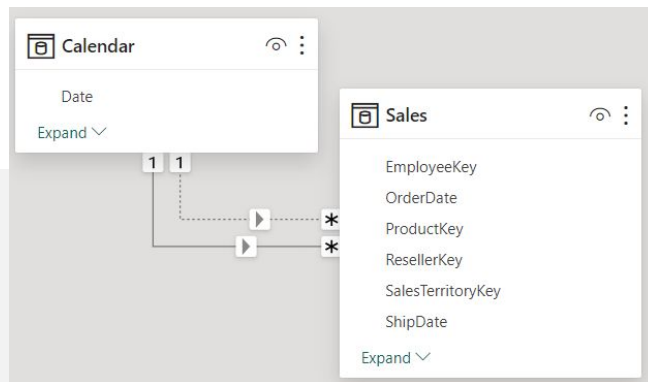
☐ Apr

2076

Orders(OrderDate)

1900

Orders(ShipDate)



```
Orders(OrderDate) =  
COUNTROWS(Sales)
```

```
Orders(ShipDate) =  
CALCULATE(  
    COUNTROWS(Sales)  
    ,USERELATIONSHIP(Sales[ShipDate], 'Calendar'[Date]))
```

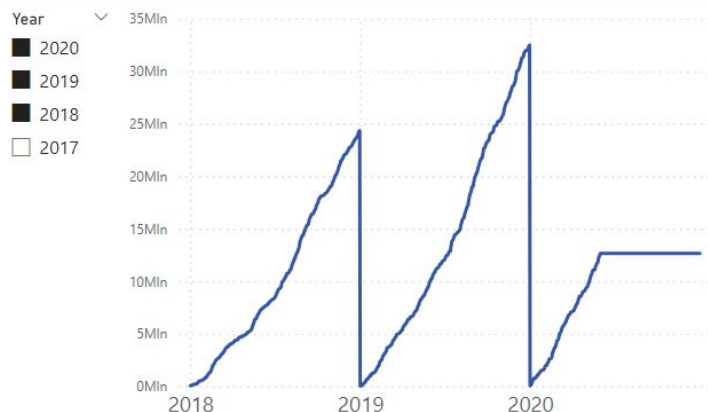
La funzione **USERELATIONSHIP** consente di utilizzare 'forzosamente' la relazione inattiva in un calcolo specifico.

È necessario indicare come argomenti le due colonne tra le quali insiste una relazione inattiva.

Non funziona se la relazione viene eliminata!

.. Altre funzioni Time-Intelligence: TOTALYTD

Sales Total Year to Date



Date	TotalSales	TOTALYTD
23/12/2018	83.821,21 €	23.043.303,00
24/12/2018	14.551,66 €	23.657.856,66
25/12/2018	121.187,00 €	23.779.043,66
26/12/2018	1.409,37 €	23.780.453,03
27/12/2018	79.134,03 €	23.859.587,06
28/12/2018	217.489,43 €	24.077.076,49
29/12/2018	150.831,50 €	24.227.907,99
30/12/2018	98.282,60 €	24.326.190,59
31/12/2018	2.595,66 €	24.328.786,25
01/01/2019	35.989,59 €	35.989,59
02/01/2019	7.046,85 €	43.036,44
03/01/2019	1.292,76 €	44.329,20
04/01/2019		44.329,20
05/01/2019	103.087,78 €	147.416,98
Totale	69.478.814,62 €	12.650.022,79

La funzione **TOTALYTD** restituisce un'aggregazione cumulativa nell'anno.

Consente di visualizzare la somma totale (in questo esempio di *TotalSales*) mano a mano che cresce nel tempo.

La somma totale viene eseguita nell'anno dall'1/1 al 31/12 a meno che non vengano esplicitati filtri particolari!

TotalYTD =
TOTALYTD([TotalSales], 'Calendar'[Date])

.. Altre funzioni Time-Intelligence: TOTALYTD

Sales Total Year to Date



Date	TotalSales	TOTALYTD
24/03/2018	7.021,34 €	14.041.231,19
25/05/2018	190.369,15 €	14.831.606,34
26/05/2018	22.852,71 €	14.854.459,05
27/05/2018	111.118,86 €	14.965.577,91
28/05/2018	3.256,64 €	14.968.834,55
29/05/2018	113.400,45 €	15.082.235,00
30/05/2018	104.106,77 €	15.186.341,77
31/05/2018	105.448,93 €	15.291.790,70
01/06/2018	3.607,08 €	15.295.397,78
02/06/2018	82.280,42 €	15.377.678,20
03/06/2018	41.664,20 €	15.419.342,40
04/06/2018		15.419.342,40
05/06/2018	115.734,56 €	15.535.076,96
06/06/2018	776,36 €	15.535.853,32
Totale	77.548.570,20 €	

Il parametro **"6-30"** è una stringa letterale che consente di definire la data di fine anno cioè, in questo esempio, il 30 Giugno.

La somma totale viene eseguita nell'anno dal 1/06 al 31/07 (l'espressione è cumulata dal 1 di Giugno al 31 Luglio).

Il parametro è opzionale mentre il valore di default che viene applicato quando omissso è il 31/12.

```
TotalYTD =
TOTALYTD([TotalSales], 'Calendar'[Date], "6-30")
```

.. Altre funzioni Time-Intelligence: TOTALQTD

Sales Total Quarter to Date



Quarter	Date	TotalSales	TotalQTD
Q1	23/03/2019	94,37 €	4.990.615,65 €
	24/03/2019	35.800,79 €	5.026.416,44 €
	25/03/2019	28.130,03 €	5.054.546,47 €
	26/03/2019	1.409,37 €	5.055.955,84 €
	27/03/2019	40.119,67 €	5.096.075,51 €
	28/03/2019	44.537,95 €	5.140.613,46 €
	29/03/2019	118.136,28 €	5.258.749,74 €
	30/03/2019	209,26 €	5.258.959,00 €
	31/03/2019	10.001,12 €	5.268.960,12 €
	Totale	5.268.960,12 €	5.268.960,12 €
Q2	01/04/2019	114.676,39 €	114.676,39 €
	02/04/2019	33.604,42 €	148.280,81 €
	03/04/2019	203.422,64 €	351.703,45 €
	04/04/2019	48.462,18 €	400.165,63 €
	05/04/2019	75.877,34 €	476.042,97 €

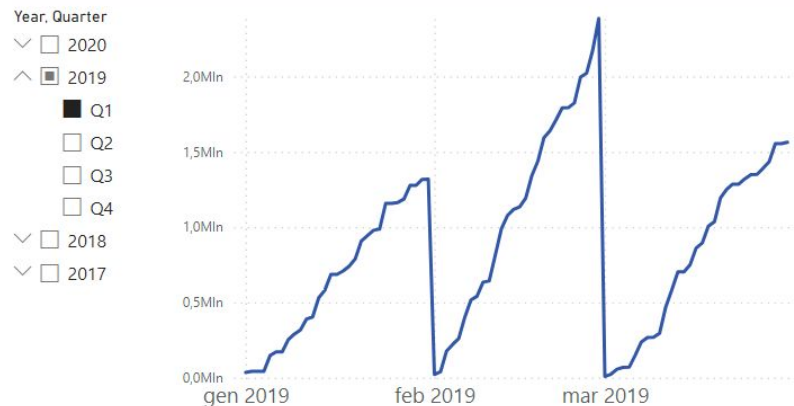
La funzione **TOTALQTD** restituisce un'aggregazione cumulativa nel trimestre.

Valgono le considerazioni fatte in precedenza!

TotalQTD =
TOTALQTD([TotalSales], 'Calendar'[Date])

.. Altre funzioni Time-Intelligence: TOTALMTD

Sales Total Month to Date



Date	TotalSales	TOTALMTD
23/01/2019	8.076,02 €	987.730,75
24/01/2019	170.688,73 €	1.158.419,48
25/01/2019	469,79 €	1.158.889,27
26/01/2019	5.735,01 €	1.164.624,28
27/01/2019	22.843,13 €	1.187.467,41
28/01/2019	90.147,76 €	1.277.615,17
29/01/2019		1.277.615,17
30/01/2019	40.297,79 €	1.317.912,96
31/01/2019	679,05 €	1.318.592,01
01/02/2019	22.633,44 €	22.633,44
02/02/2019	17.283,82 €	39.917,26
03/02/2019	138.826,12 €	178.743,38
04/02/2019	42.938,07 €	221.681,45
Totale	5.268.960,12 €	1.564.294,92

La funzione **TOTALMTD** restituisce un'aggregazione cumulativa nel mese.

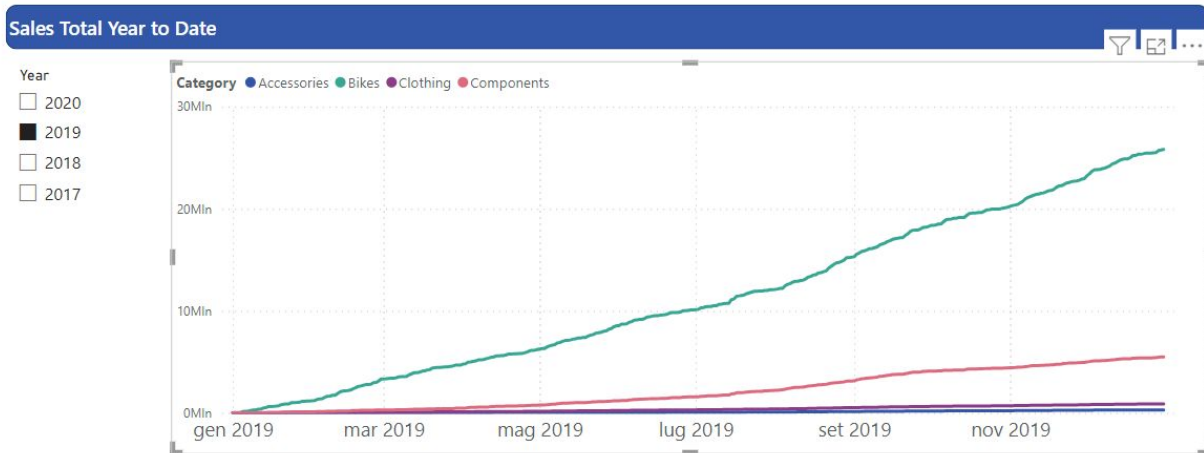
Valgono le considerazioni fatte in precedenza!

TotalMTD =
 TOTALMTD([TotalSales], 'Calendar'[Date])

.. Altre funzioni Time-Intelligence: TOTALMTD

Quali informazioni è possibile dedurre dalla visualizzazione?

Cosa è accaduto?/Cosa sta accadendo?



Visualizzazioni

Compila oggetto visivo

Filtri

Asse X

Date

Asse Y

TOTALYTD

Asse Y secondario

Aggiungere qui i campi dati

Legenda

Category

Check Point!

Bisogna calcolare il fatturato al mese precedente. Come completeresti l'espressione indicata:

SalesPY =

```
CALCULATE(  
    [TotalSales]  
    , PARALLELPERIOD('Calendar'[Date], __, __))
```

Perché è così importante la funzione CALCULATE?

**Let's take a
look!**



GRAZIE
Epicode