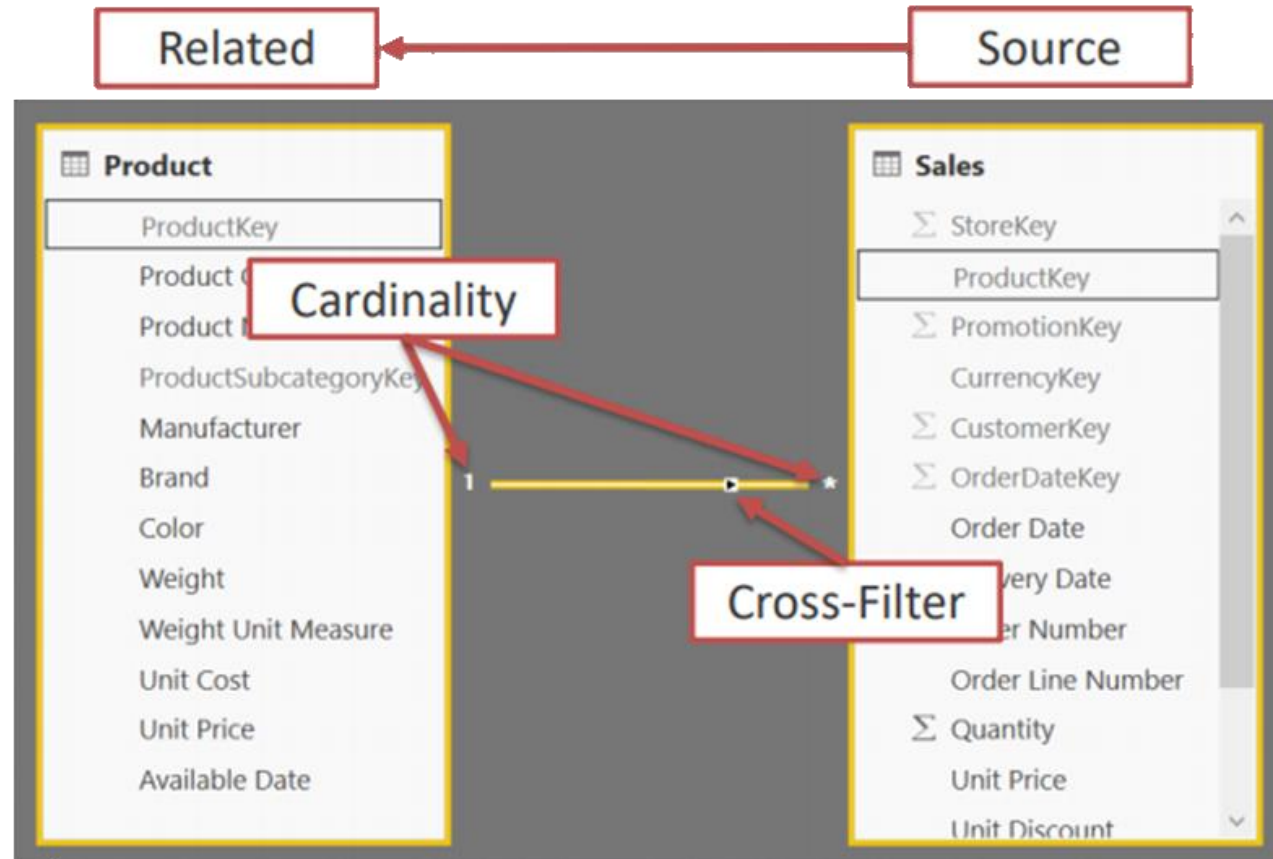


Relazioni in PBI Desktop

- Che cosa è una relazione?
- Cross Filter Direction
- Vedere e Creare relazioni
- Cardinalità (tipi di una relazione)

Che cosa sono le relazioni in PBI Desktop?



Che cosa sono le relazioni in PBI Desktop?

- Una relazione crea un legame tra più tabelle in modo da poterle trattare come se fossero una sola:
- Normalmente sono create in sistemi OLTP come parte del processo di normalizzazione aggiungendo delle chiavi nelle tabelle
- Permettono di prevenire valori duplicati e ogni entità ha solo gli attributi che gli appartengono
- Nei data warehouse vengono usate nelle tabelle dei fatti con delle chiavi che permettono di unire quest'ultima alle tabelle delle dimensioni
- Power BI ha una caratteristica di Autodetect che può riconoscere le relazioni e crearle manualmente

Cross-filter direction in PBI Desktop

- **Single**
 - La propagazione del filter context avviene dal lato uno al lato molti
 - E' il suo comportamento di default
 - Conveniente, sicuro e veloce
- **Both**
 - La propagazione del filtro avviene in entrambe le direzioni
 - Necessita di essere attivato
 - Non è sicuro, lento, molto pericoloso

Vedere e modificare le relazioni in PBI Desktop

- La caratteristica di autodetect di Power BI fa sì che:
 - Le relazioni siano create in modo automatico dopo che il caricamento dei dati è stato completato
 - L' autodetect determina la cardinalità e la direzione del cross filter direction in una relazione
 - Per vedere ed editare le relazioni create con Power BI nella relationships view possiamo usare il relationship diagram
 - Quando Power BI trova una o più relazioni tra due tabelle solamente una può essere attiva ed è settata come default ma possiamo disabilitare le relazioni attivate per errore
 - Si può cancellare una relazione dalla relationship view

Creazione delle relazioni in PBI Desktop

- Creazione manuale della relazione
 - Nella relationships view si trascina dalla prima tabella la colonna della relazione sopra la colonna che si vuole relazionare nella seconda tabella. La cardinalità e la cross filter direction vengono settati automaticamente
 - Cliccare su manage relationships per aprire il dialog in cui mi permette di creare le relazioni

Cardinalità di una relazioni in PBI Desktop

- Si riferisce alla relazione tra due tabelle
- **One**
 - Colonna che necessita di valori univoci
 - E' il target di una tabella espansa (**concetto avanzato**)
 - E' la sorgente della propagazione del filter context
- **Many**
 - Colonna che potrebbe contenere duplicati
 - Sorgente della tabella espansa (**concetto avanzato**)
 - E' il target della propagazione del filter context

Cardinalità di una relazione in PBI Desktop

- **One-to-many**
 - La relazione più importante
 - (*:1): Questo è il tipo predefinito e più comune. La prima tabella presenta più istanze del valore di join. La seconda tabella ha un'istanza del valore;
 - (1:*) : Questa è la relazione inversa del tipo Many to One.
- **One-to-one (1:1):**
 - Meno comune di Molti a uno, in quanto solo una istanza del valore esiste nelle due tabelle correlate.
 - L'espansione (blank row viene forzata) avviene in entrambe le direzioni (**concetto avanzato**)
 - Il Cross-filter necessita di essere di tipo both

Cardinalità di una relazione in PBI Desktop

- Many-to-Many (*:*) Weak relationships!!!!
 - Nuova e pericolosa
 - Non c'è espansione di tabella (blank row non viene forzata)
 - Necessita di decidere il cross-filter direction
 - Presente da Ottobre 2018 (Attenzione non confondete con la many-to-many tra dimensioni!!!)
 - Implementabile con un pattern (*:1) + (1:*)

Demo 6 Relazioni

In questa dimostrazione vedremo le relazioni

Laboratorio che consolida le basi di Power Query e aggiunge trasformazioni che ancora non abbiamo visto (**da svolgere assieme in classe**)

DAX Queries

- Che cosa è DAX?
- Sintassi
- Funzioni
- Contesti

Che cosa è DAX?

- Data Analysis Expressions (DAX) è un formula language:
 - Comprende una libreria di più di 200 funzioni, costanti, e operatori
 - Usare DAX in una formula o in una espressione per calcolare e ritornare un valore singolo o valori multipli
 - Non ci sono nuove caratteristiche in versione Power BI; esiste già in Power Pivot per Excel e SQL Server Analysis Services (SSAS). E' disegnato per lavorare con i dati relazionali
 - Con DAX si possono eseguire calcoli come year-on-year sulle vendite, running totals, like-for-like sulle vendite, e predict profit
 - Ci aiuta ad aumentare la capacità di analisi

Linguaggio Funzionale

DAX è un linguaggio funzionale, il flusso di esecuzione è con le chiamate alle funzioni (una funzione chiama una funzione che a sua volta ne chiama un'altra etc...non ci sono iterazioni) qui vediamo un esempio di formula DAX

```
=  
SUMX (  
    FILTER ( VALUES ( 'Date'[Year] ), 'Date'[Year] < 2005 ),  
    IF ( 'Date'[Year] >= 2000, [Sales Amount] * 100, [Sales Amount] * 90 )  
)
```

Tipi in DAX

- Tipi numerici
 - Integer (64 bit)
 - Decimal (floating point)
 - Currency (money)
 - Date (DateTime)
 - TRUE / FALSE (Boolean)
- Altri tipi
 - String
 - Binary Objects

Gestione dei tipi in DAX

- Operator Overloading
 - Gli operatori non sono fortemente tipizzati
 - Il risultato dipende dall'input
- Esempio:
 - "1" + "2" = 3
 - 1 & 2 = "12"
- La conversion avviene quando è necessario
 - Ad esempio quando non vogliamo che essa accada

DateTime

- Valori Floating point
- La parte intera
 - Numero di giorni dopo il Dicembre, 30, 1899
- La parte decimale
 - Secondi: $1 / (24 * 60 * 60)$
- Espressioni DateTime
 - $\text{Data} + 1$ = Rappresenta il giorno dopo
 - $\text{Data} - 1$ = Rappresenta il giorno prima

Sintassi DAX per le colonne

- Il formato originale
 - 'TableName'[ColumnName]
- Gli apici possono essere omessi:
 - Se TableName non contiene spazi
 - **Best practice: non usate spazi e omettete gli apici**
- TableName può essere omessa
 - La ColumnName viene cercata sulla tabella corrente
 - Non fatelo, difficile da capire
- Le parentesi quadre per il ColumnName non si possono omettere

Colonne calcolate e misure

- GrossMargin = SalesAmount - ProductCost
 - Calculated column
- GrossMargin% = GrossMargin / SalesAmount
 - Non può essere calcolata riga per riga
- Necessita di una misura

$$\sum \frac{\text{Margin}}{\text{SalesAmount}} \neq \frac{\sum \text{Margin}}{\sum \text{SalesAmount}}$$

Demo 9

Misure e Colonne Calcolate

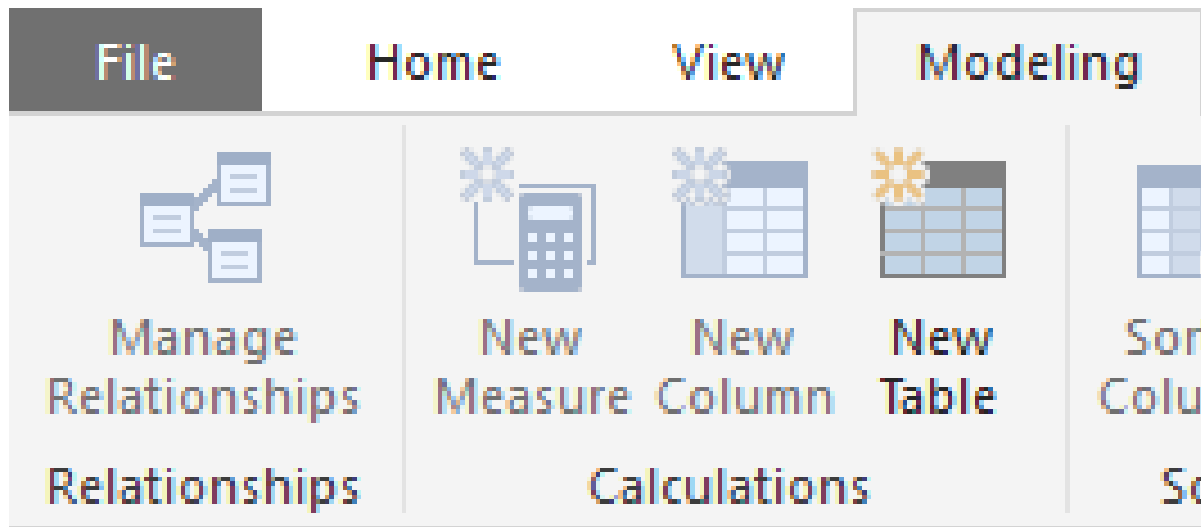
In questa dimostrazione vedremo le misure e le colonne calcolate

Misure

- Scritte usando DAX
- Non lavorano "riga" per "riga"
- Usano tabelle ed aggregazioni
- Non abbiamo il concetto di "current row"
- Esempio:
 - GrossMargin
 - È una colonna calcolata
 - Ma può essere una misura
 - GrossMargin %
 - Necessita essere una misura

Misure

- Si crea una misura da report view o data view
- Le misure possono essere usate nelle visualizzazioni come altre colonne



Misure vs Colonne Calcolate

- Usiamo una colonna quando:
 - Necessitiamo di fare lo slice o filtrare per un valore
- Usiamo una misura:
 - Calcolo delle percentuali
 - Calcolo dei rapporti
 - Necessitiamo di calcoli complessi
- Uso di spazio e di CPU
 - Le colonne consumano memoria
 - Le misure consumano CPU

Calculated Tables

- Creazione di una tabella calcolata usando i dati esistenti nel modello
 - Creazione di una tabella Report View, o Data View
 - Usare i dati dal modello per creare una nuova tabella rispetto ad utilizzare una query come sorgente dato
 - Dal tab di modellazione e cliccando su New Table nel gruppo Calculations e aggiungendo la formula DAX
 - Usare le funzioni come **UNION**, **NATURALINNERJOIN**, **NATURALLEFTOUTERJOIN**, o **DATATABLE**
 - Le tabelle calcolate e le colonne calcolate possono essere usate come tabelle normali. E' possibile rinominare tabelle e colonne e queste possono anche essere usate nelle relazioni con altre tabelle. E' consentito cambiare il tipo dei dati, aggiungere colonne e misure ed infine si possono usare nelle visualizzazioni

Funzioni

- Le funzioni DAX sono formule predefinite che permettono di eseguire calcoli di uno o più argomenti:
 - Possiamo passare come argomento una colonna, funzione, espressione, formula, costante, number, testo o una condizione booleana
 - Ci sono più di 200 funzioni!!
 - Le funzioni DAX sono simili a Excel con la differenza che si riferiscono all'intera tabella o ad una colonna e si usano i filtri per referenziare i valori selezionati
 - Una funzione ritorna una tabella e non visualizza un risultato
 - La funzione Excel VLOOKUP è sostituita dal modello relazionale costruito dietro le quinte

Funzioni di aggregazione

- Utilizzate per aggregare valori
 - SUM
 - AVERAGE
 - MIN
 - MAX
- Lavorano solo con le colonne numeriche
- Si può aggregare solo su una colonna
 - SUM (Orders[Price])
 - SUM (Orders[Price] * Orders[Quantity])

Le funzioni di aggregazione "X"

- Iteratori: utilizzati per aggregare le formule
 - SUMX
 - AVERAGEX
 - MINX
 - MAXX
- Iterare sopra una tabella e valutare l'espressione per ogni riga
- Spesso riceve due parametri
 - Tabella da iterare
 - Formula da valutare per ogni riga
- Di fatto dietro alle quinte una SUM è una SUMX

Esempio di SUMX

Per ogni riga nella tabella delle vendite viene valutata la formula che somma assieme tutti i risultati della formula interna nella riga corrente

```
=  
SUMX ( Sales, Sales[Price] * Sales[Quantity] )
```

Conteggio di valori

- Utili per conteggiare valori
 - COUNT (Solo per colonne numeriche)
 - COUNTA (conta tutto ma non i blanks)
 - COUNTBLANK (conta solo i blanks)
 - COUNTROWS (conta le righe nella tabella)
 - DISTINCTCOUNT (elabora un distinct count)
- Es.
- =COUNTROWS (Sales)
- = COUNTA (Sales[SalesID])
- = COUNTBLANK (Sales[SalesID])

Funzioni logiche

- Forniscono una logica booleana
 - AND
 - OR
 - NOT
 - IF
 - IFERROR
- IF / IFERROR sono molto utilizzate
- AND/OR/NOT si possono esprimere con operatori
 - $\text{AND}(A, B) = A \ \&\& \ B$

Funzioni informative

- Forniscono informazioni sulle espressioni
 - ISBLANK
 - ISNUMBER
 - ISTEXT
 - ISNONTTEXT
 - ISERROR
- Non è molto utile conoscere il tipo di espressioni che si valuta
- Sono difficili da usare perchè dipendono dal tipo di dato della colonna e non dal valore della cella come Excel

Funzioni matematiche

- Si comportano esattamente come ci aspettiamo
 - ABS, EXP
 - FACT, LN
 - LOG, LOG10
 - MOD, PI
 - POWER, QUOTIENT
 - SIGN, SQRT

Funzioni di arrotondamento

- Molte funzioni di arrotondamento
 - FLOOR (Value, 0.01)
 - TRUNC (Value, 2)
 - ROUNDDOWN (Value, 2)
 - MROUND (Value, 0.01)
 - ROUND (Value, 2)
 - CEILING (Value, 0.01)
 - ISO.CEILING (Value, 0.01)
 - ROUNDUP (Value, 2)
 - INT (Value)

Funzioni di testo

- Molto simili alle funzioni di Excel
 - CONCATENATE,
 - FIND, FIXED, FORMAT,
 - LEFT, LEN, LOWER, MID,
 - REPLACE, REPT, RIGHT,
 - SEARCH, SUBSTITUTE, TRIM,
 - UPPER, VALUE, EXACT
 - CONCATENATE, CONCATENATEX

Funzioni Date

- Molte funzioni utili
 - DATE, DATEVALUE, DAY, EDATE,
 - EOMONTH, HOUR, MINUTE,
 - MONTH, NOW, SECOND, TIME,
 - TIMEVALUE, TODAY, WEEKDAY,
 - WEEKNUM, YEAR, YEARFRAC

La funzione Switch

- Si riesce facilmente a realizzare IF annidati
- Internamente è convertito in un set di IF annidati

```
SizeDesc =  
SWITCH (  
    Product[Size],  
    "S", "Small",  
    "M", "Medium",  
    "L", "Large",  
    "XL", "Extra Large",  
    "Other"  
)
```

Switch to perform CASE WHEN

Si può utilizzare lo switch per realizzare un CASE WHEN

```
DiscountPct =  
SWITCH (  
    TRUE (),  
    Product[Size] = "S", 0.5,  
    AND ( Product[Size] = "L", Product[Price] < 100 ), 0.2,  
    Product[Size] = "L", 0.35,  
    0  
)
```

MAX e MIN

Usati entrambi per aggregare e confrontare valori

La seguente formula calcola il massimo del Sales Amount

```
=  
MAX ( Sales[SalesAmount] )
```

Di seguito la corrispondente formula scritta con una sintassi di DAX 2015 (excel 2016, Power BI desktop e SSAS 2016)

```
=  
MAXX ( Sales, Sales[SalesAmount] )
```

Si possono anche comparare due valori

```
=  
MAX ( Sales[SalesAmount], Sales[ListPrice] )
```

Che corrisponderebbe ad eseguire il seguente test

```
=  
IF ( Sales[Amount] > Sales[ListPrice], Sales[Amount], Sales[ListPrice] )
```

La funzione DIVIDE

DIVIDE è utile perchè Evita di usare l'IF all'interno delle espressioni per controllare che il denominatore non sia 0

```
=  
IF ( Sales[SalesAmount] <> 0, Sales[GrossMargin] / Sales[SalesAmount], 0 )
```

Si può Scrivere in un modo migliore con DIVIDE

```
=  
DIVIDE ( Sales[GrossMargin], Sales[SalesAmount], 0 )
```

Funzioni Relazionali

- **RELATED**
 - Segue la relazione e ritorna il valore di una Colonna. Si usa quando si vuole portare l'informazione nel lato 1 della relazione.
- **RELATEDTABLE**
 - Segue le relazioni e ritorna tutte le righe in relazione con la riga corrente. Si usa quando si vuole portare l'informazione nel lato M della relazione.
- Non importa quanto lunga è la catena delle relazioni

Demo 10 - Related e Relatedtable

In questa dimostrazione vedremo le funzioni per lavorare con le relazioni