Modulo 5

Modelling Data

Lavorare con 1 singola tabella

- In excel siamo abituati a lavorare con una singola tabella
- Con excel è semplice ed è già un data model
- Ma ci sono delle limitazioni
  - Il numero di righe è minore di un milione (Il limite della dimensione limita la dimensione del modello)
  - La velocità e la memoria non sono ottimali
  - Possiamo fare solo calcoli basici

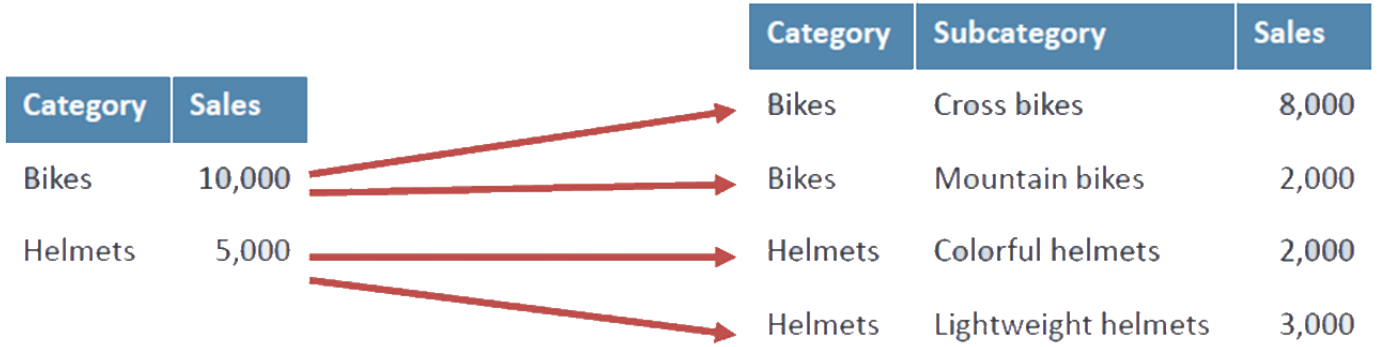
Lavorare con una singola tabella (demo 0): Problema 1

- Problema 1: limite analitico
- Se abbiamo il seguente excel (**L'ESPRESSIVITA**' massima è data dalle colonne che si vedono)
- Se non ci sono le colonne nel tabellone non si possono fare le analisi
  - Se dobbiamo aggiungere l'analisi per colore che non c'è allora dobbiamo portarci il colore dentro la nostra tabella aumentando di fatto le dimensioni. Aumenteremo la granularità e la dimensione della tabella

FullDateLabel	Manufacturer	BrandName	ProductCategoryName	SalesQuantity	SalesAmount	TotalCost
2007-03-31	Adventure Works	Adventure Works	Home Appliances	55	14332.268	7651.84
2008-10-22	Contoso, Ltd	Contoso	Cell phones	2040	23504.88	12648.94
2009-01-31	Adventure Works	Adventure Works	TV and Video	194	51593.106	28146.4
2009-01-21	Fabrikam, Inc.	Fabrikam	Cameras and camcorders	282	163007.2	76709.45
2007-12-31	Adventure Works	Adventure Works	Computers	29	14008.43	7944.32
2007-06-22	Contoso, Ltd	Contoso	Cell phones	680	6107.24	3420.44
2007-06-22	Proseware, Inc.	Proseware	Computers	86	71417.6	30786.94
2007-08-23	Adventure Works	Adventure Works	Computers	43	22672.2	9954.6
2009-03-30	The Phone Company	The Phone Company	Cell phones	198	48500.37	24164.56
2008-03-24	Contoso, Ltd	Contoso	Cell phones	306	7353.594	3914.64
2007-09-30	Fabrikam, Inc.	Fabrikam	Home Appliances	44	4805.604	2824.24
2007-11-13	Adventure Works	Adventure Works	Computers	153	47357.97	28256.02
2008-12-06	Contoso, Ltd	Contoso	Computers	32	10790.4	6477.2
2007-11-14	Contoso, Ltd	Contoso	Cameras and camcorders	146	55397.5	25876
2009-12-30	Adventure Works	Adventure Works	Computers	32	15107.75	7952.97
2009-03-13	Wide World Importers	Wide World Importers	Audio	42	7990.92	3607.26
2009-08-11	Wide World Importers	Wide World Importers	Audio	9	1466.1	749.16
2009-09-28	Contoso, Ltd	Contoso	Home Appliances	78	9955.268	5189.27
2008-02-18	A. Datum Corporation	A. Datum	Cameras and camcorders	345	70989.93	32872.58
2007-08-15	Litware, Inc.	Litware	Home Appliances	69	112603.8	56472.35

Lavorare con 1 singola tabella (demo 0): Problema 2

- Problema 2: limite dimensioni di 1 milione di righe
- **Granularità** = Livello di dettaglio della tabella
- Supponiamo ora di aggiungere la Sottocategoria nel nostro tabellone. Questa non solo aumenta la dimensione della tabella ma mi va ad incidere sulla granularità cioè sul dettaglio minimo di ogni riga della tabella di fatto aumentando le righe e noi sappiamo che abbiamo solo 1 milione di righe
- Più granularità più aumentano le righe oltre alle colonne quindi abbiamo un apumento in profondità e in larghezza



Category	Sales
Bikes	10,000
Helmets	5,000

Category	Subcategory	Sales
Bikes	Cross bikes	8,000
Bikes	Mountain bikes	2,000
Helmets	Colorful helmets	2,000
Helmets	Lightweight helmets	3,000

Che cosa è un data model

- Abbiamo visto i limiti di lavorare con una unica tabella in excel
- Usando Power Pivot o Power BI si sfruttano meglio i dati del modello
- Non ci sono limiti nella dimensione delle tabelle quindi possiamo definire la granularità a piacimento, anche con una sola tabella
- Con il data model quindi:
  - Carichiamo più tabelle in un modello singolo
  - Costruiamo delle relazioni tra le tabelle
  - Anche se non è argomento di questa parte con un data model possiamo sfruttare la potenza di DAX

Basi di modellazione: errori (demo 0)

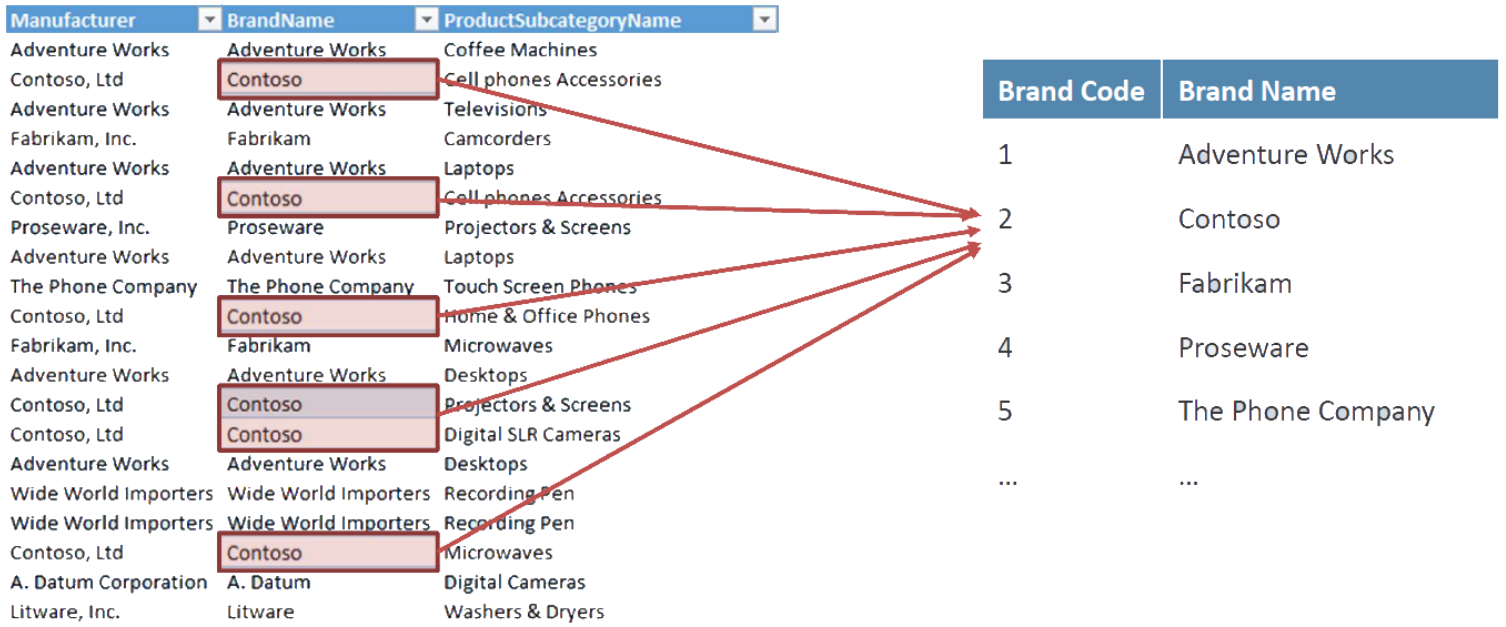
- Ora abbiamo scoperto che con Power BI o Power Pivot per Excel i limiti analitici e i limiti di memoria si possono superare

- La prossima fase è pensare a come costruire il data model in maniera corretta in quanto non basta passare ai tool di Power BI e Power Pivot per risolvere i nostri problemi.

- Non possiamo pensare comunque di mettere tutto su una tabella e tra poco vediamo perchè

Normalizzazione e denormalizzazione

- Quello che abbiamo fatto nella demo precedente si chiama normalizzazione e denormalizzazione
- La normalizzazione è il processo di organizzazione di colonne (attributi) e tabelle di un database per ridurre la ridondanza dei dati e migliorare l'integrità dei dati



Manufacturer	BrandName	ProductSubcategoryName
Adventure Works	Adventure Works	Coffee Machines
Contoso, Ltd	Contoso	Cell phones Accessories
Adventure Works	Adventure Works	Televisions
Fabrikam, Inc.	Fabrikam	Camcorders
Adventure Works	Adventure Works	Laptops
Contoso, Ltd	Contoso	Cell phones Accessories
Proseware, Inc.	Proseware	Projectors & Screens
Adventure Works	Adventure Works	Laptops
The Phone Company	The Phone Company	Touch Screen Phones
Contoso, Ltd	Contoso	Home & Office Phones
Fabrikam, Inc.	Fabrikam	Microwaves
Adventure Works	Adventure Works	Desktops
Contoso, Ltd	Contoso	Projectors & Screens
Contoso, Ltd	Contoso	Digital SLR Cameras
Adventure Works	Adventure Works	Desktops
Wide World Importers	Wide World Importers	Recording Pen
Wide World Importers	Wide World Importers	Recording Pen
Contoso, Ltd	Contoso	Microwaves
A. Datum Corporation	A. Datum	Digital Cameras
Litware, Inc.	Litware	Washers & Dryers

Brand Code	Brand Name
1	Adventure Works
2	Contoso
3	Fabrikam
4	Proseware
5	The Phone Company
...	...

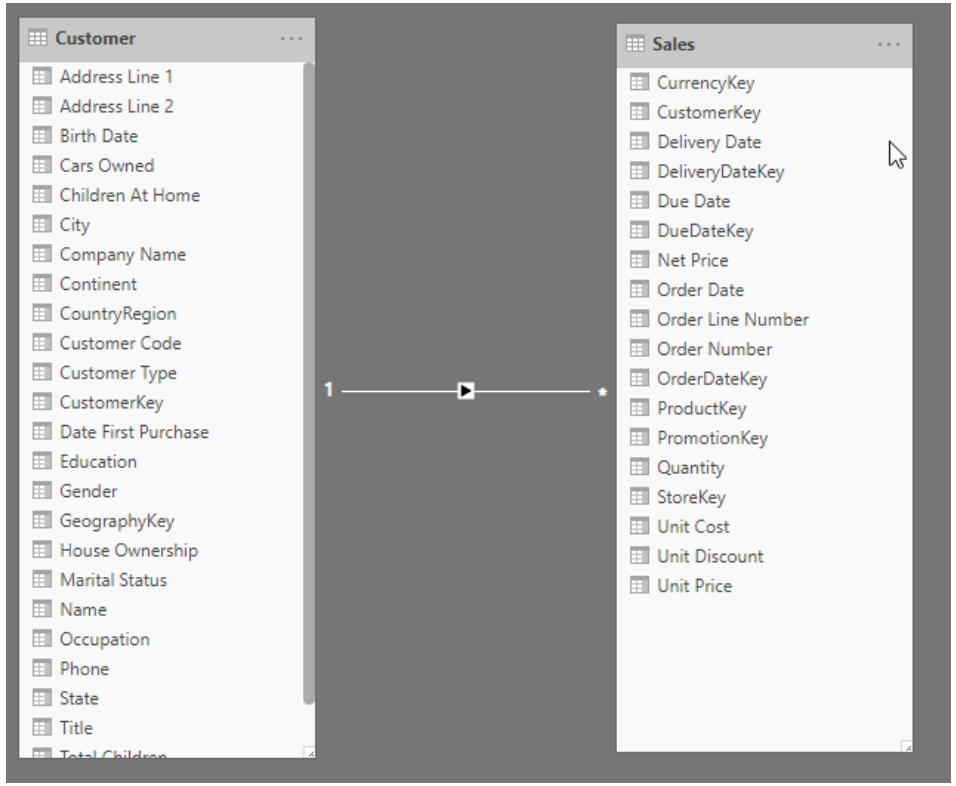
- La denormalizzazione è l'opposto della normalizzazione, quella aumenta la ridondanza dei dati, con l'obiettivo di migliorare la comprensione del modello (Denomalizzazione nei customer)

Abbiamo imparato che se il data model è fatto bene le nostre formule DAX saranno molto semplici ☺

E il primo step per diventare modellatori di dati

Basi di modellazione: Business Entity

- Nella demo che abbiamo fatto la tabella dei cutomer è vista come una business entity
- Tutti i campi della customer sono riferiti al concetto di customer a differenza della tabella delle vendite
- Ogni business entity DEVE avere una tabella per se
- Ogni business ha differenti entità
  - Products,Customer, Resellers
  - Patients, Medications, doctor
  - Claim, Customer
- Ogni business entity ha una caratteristica unica



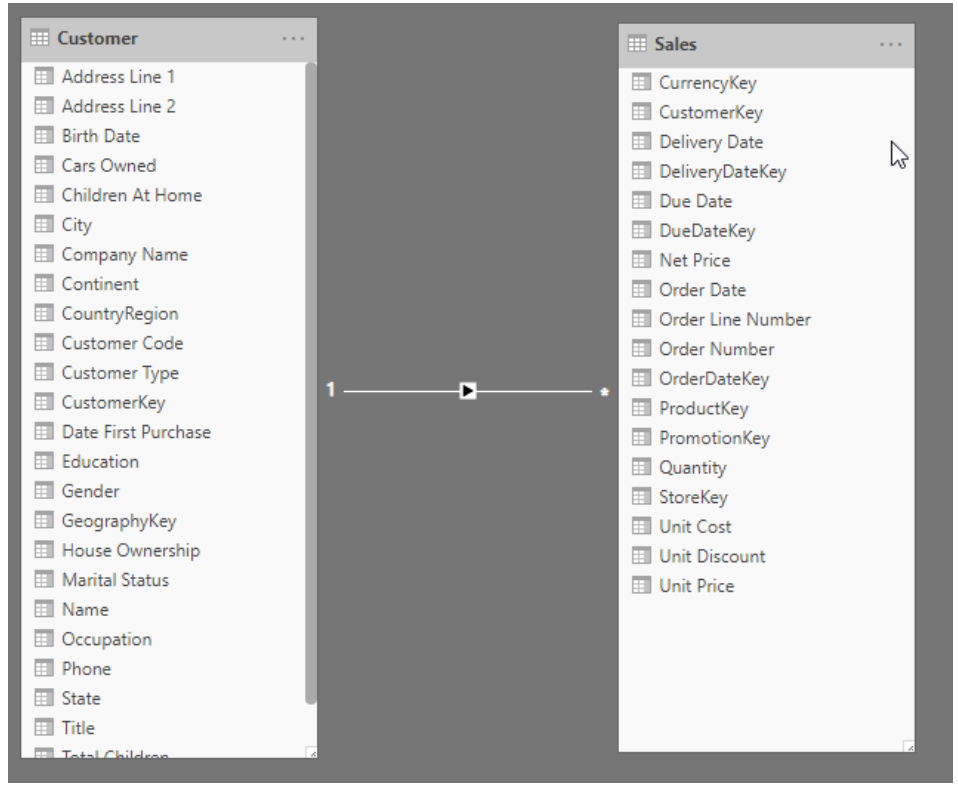
Basi di modellazione: Business Entity

- Perché Education non è una tabella?

- Education non è una busioness entity ma è un **attributo** del **customer**. Customer è la tabella rilevante
- Definire che cosa è un attributo e cosa è una entity non è subito facile ma viene dall'esperienza e dalla pratica dipende dagli scenari ad esempio:
  - Customer è possibile vederla come attributo di un customer in certi scenari mentre in altri potrebbe essere vista come tabella a se per rappresentare una business entity

Basi di modellazione: Business Entity

- Quando si hanno più tabelle in un moello per forza di cose bisogna unirle con una relazione
- Tabelle senza relazioni in un modello non servono a nulla
- Una relazione puà avvenire tra due tabelle
- Una relazione ha una direzione:
  - Customer lato uno
  - Sales: lato molti
- Best Pratice: Dare lo stesso nome di colonna tra le tabelle che devo mettere in relazione



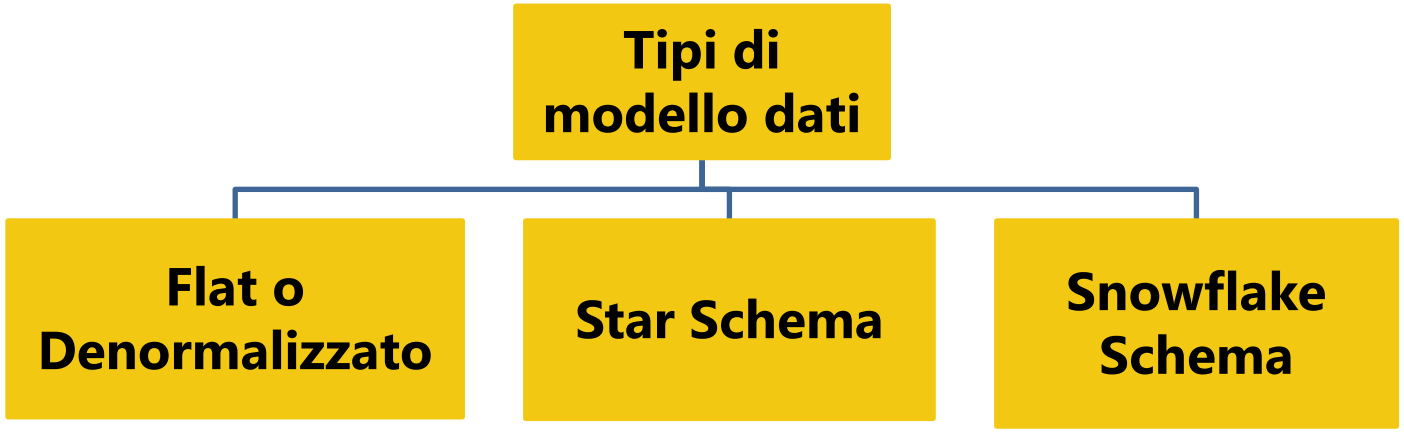
Basi di modellazione: Granularity

- Quando si hanno più tabelle in un modello la granularità è differente

- Ogni tabella ha la sua granularità non come nel tabellone unico
  - Customer ha una granularità a livello di customer
  - Data ha una granularità a livello di data
  - Product ha una granularità a livello di prodotto

- Le vendite (Sales) hanno una granularità definitia dalle relazioni
  - Customer, Date e product level
  - Se abbiamo entrambe le tre tabelle

Tipi di modello



Nota: Questo non è un elenco esaustivo, ma sono i tipi di modello più comuni utilizzati da Power BI.

Flat or Denormalized schema (“Kitchen sink” schema) evitaimolo!!!

	ProductID	Product	Date	CustomerID	Email	Last Name	First Name	Full Name	CampaignID	Units	CatSegID
1	676	Maximus UC-41	9/25/2011	70283	Farrah.Kent@xyza.com	Kent	Farrah	Farrah Kent	22	1	10
2	585	Maximus UC-50	3/24/2014	70283	Farrah.Kent@xyza.com	Kent	Farrah	Farrah Kent	15	1	10
3	585	Maximus UC-50	11/30/2014	138334	Martha.Mcclain@xyza.…	Mcclain	Martha	Martha Mcclain	8	1	10
4	585	Maximus UC-50	6/21/2015	27193	Hedda.Mcintosh@xyza.…	Mcintosh	Hedda	Hedda Mcintosh	22	1	10
5	585	Maximus UC-50	1/6/2013	238970	Lunea.Walker@xyza.com	Walker	Lunea	Lunea Walker	21	1	10
6	585	Maximus UC-50	3/22/2013	182241	Upton.Page@xyza.com	Page	Upton	Upton Page	17	1	10
7	449	Maximus UM-54	9/25/2011	195385	Drake.Wells@xyza.com	Wells	Drake	Drake Wells	22	1	4
8	449	Maximus UM-54	9/30/2014	168009	Wallace.Bender@xyza.…	Bender	Wallace	Wallace Bender	17	1	4
9	449	Maximus UM-54	8/12/2014	110391	Astra.Erickson@xyza.…	Erickson	Astra	Astra Erickson	20	1	4
10	449	Maximus UM-54	4/16/2014	49327	Echo.Bradley@xyza.com	Bradley	Echo	Echo Bradley	7	1	4
11	449	Maximus UM-54	2/28/2013	65952	Yoko.Gross@xyza.com	Gross	Yoko	Yoko Gross	17	1	4
12	449	Maximus UM-54	6/6/2013	97	Yoshi.Grant@xyza.com	Grant	Yoshi	Yoshi Grant	10	1	4
13	449	Maximus UM-54	5/14/2013	56757	Brian.Carrillo@xyza.…	Carrillo	Brian	Brian Carrillo	10	1	4
14	449	Maximus UM-54	4/9/2015	248715	Mark.Hewitt@xyza.com	Hewitt	Mark	Mark Hewitt	19	1	4
15	449	Maximus UM-54	4/28/2013	248715	Mark.Hewitt@xyza.com	Hewitt	Mark	Mark Hewitt	8	1	4
16	449	Maximus UM-54	3/28/2014	240831	Oscar.Avila@xyza.com	Avila	Oscar	Oscar Avila	18	1	4
17	449	Maximus UM-54	2/26/2014	201004	Duncan.Mcintosh@xyza…	Mcintosh	Duncan	Duncan Mcintosh	19	1	4
18	615	Maximus UC-80	5/14/2012	212645	Jacob.Santiago@xyza.…	Santiago	Jacob	Jacob Santiago	22	1	10
19	615	Maximus UC-80	5/14/2012	70666	Hilary.Collier@xyza.…	Collier	Hilary	Hilary Collier	22	1	10
20	615	Maximus UC-80	5/14/2012	114459	Chester.Mitchell@xyz…	Mitchell	Chester	Chester Mitche…	22	1	10
21	615	Maximus UC-80	5/14/2012	221670	Sage.Yang@xyza.com	Yang	Sage	Sage Yang	22	1	10
22	615	Maximus UC-80	6/3/2012	168009	Wallace.Bender@xyza.…	Bender	Wallace	Wallace Bender	22	1	10
23	615	Maximus UC-80	6/3/2012	154439	Iliana.Dunlap@xyza.c…	Dunlap	Iliana	Iliana Dunlap	22	1	10
24	615	Maximus UC-80	6/4/2012	191391	Joelle.Lee@xyza.com	Lee	Joelle	Joelle Lee	22	1	10

- Tutti gli attributi per il modello esistono in una singola tabella
- Altamente inefficiente
- Il modello ha copie extra di dati e prestazioni lente
- Le dimensioni di una tabella piatta possono esplodere molto rapidamente quando il modello di dati diventa complesso

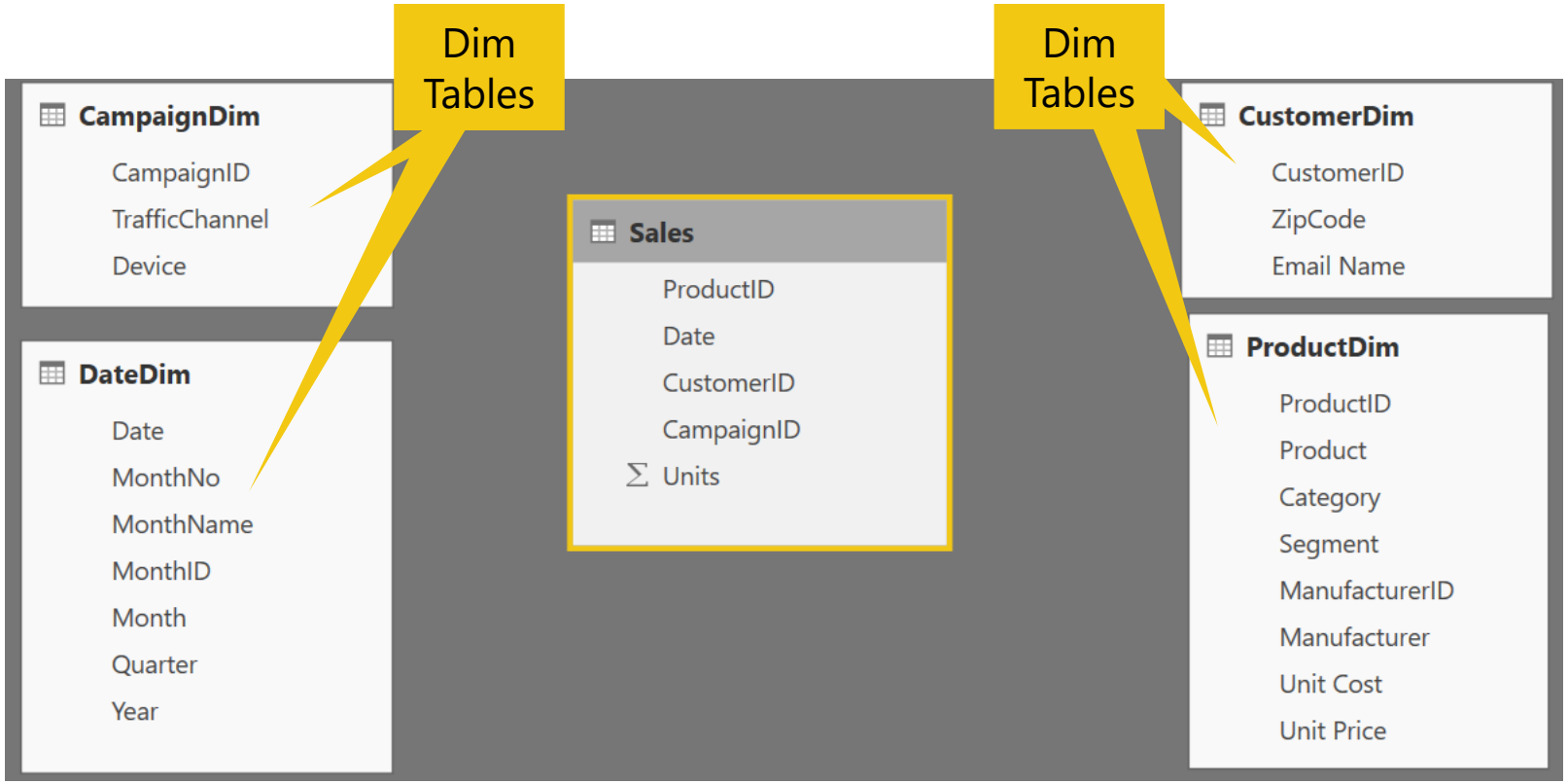
Componenti di un modello di dati – Fact Table (Tabella dei fatti)

Fact Table

- Contiene le misure (oggetti che posso aggregare) di un processo di business
- Esempi:
- Transactions
- Sales Revenue
- Units
- Cost

- Le misure sono normalemente filtrabili Esempio:
- By Month, By Customer

Componenti di un modello di dati – Dim Table

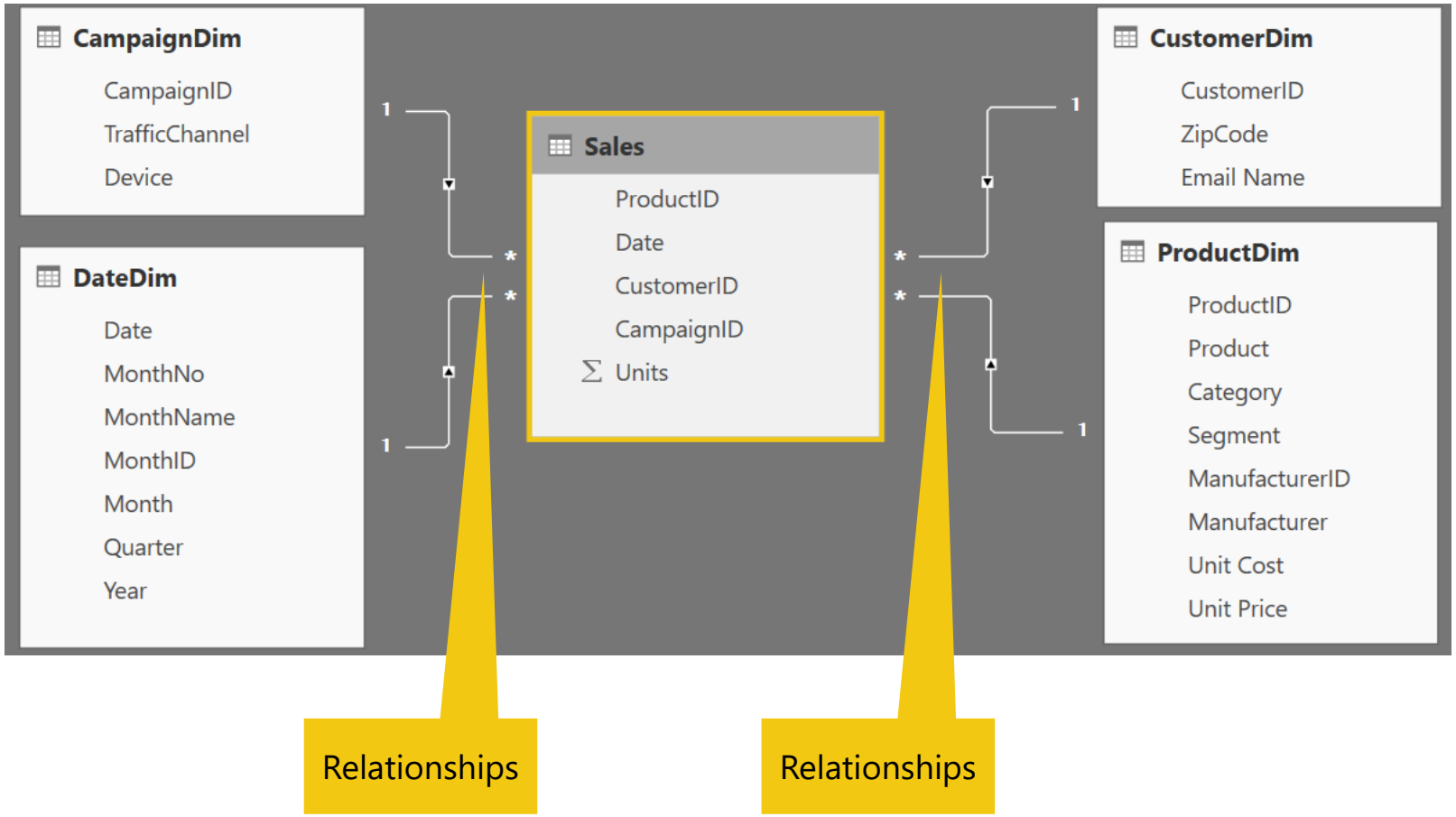


Dim Table

- Una Dim (o Dimensione) La tabella contiene attributi descrittivi che definiscono come un fatto dovrebbe essere aggregato

- Esempio:
- By month,
- By Customer, By Geo

Componenti di un modello di dati - Relazioni

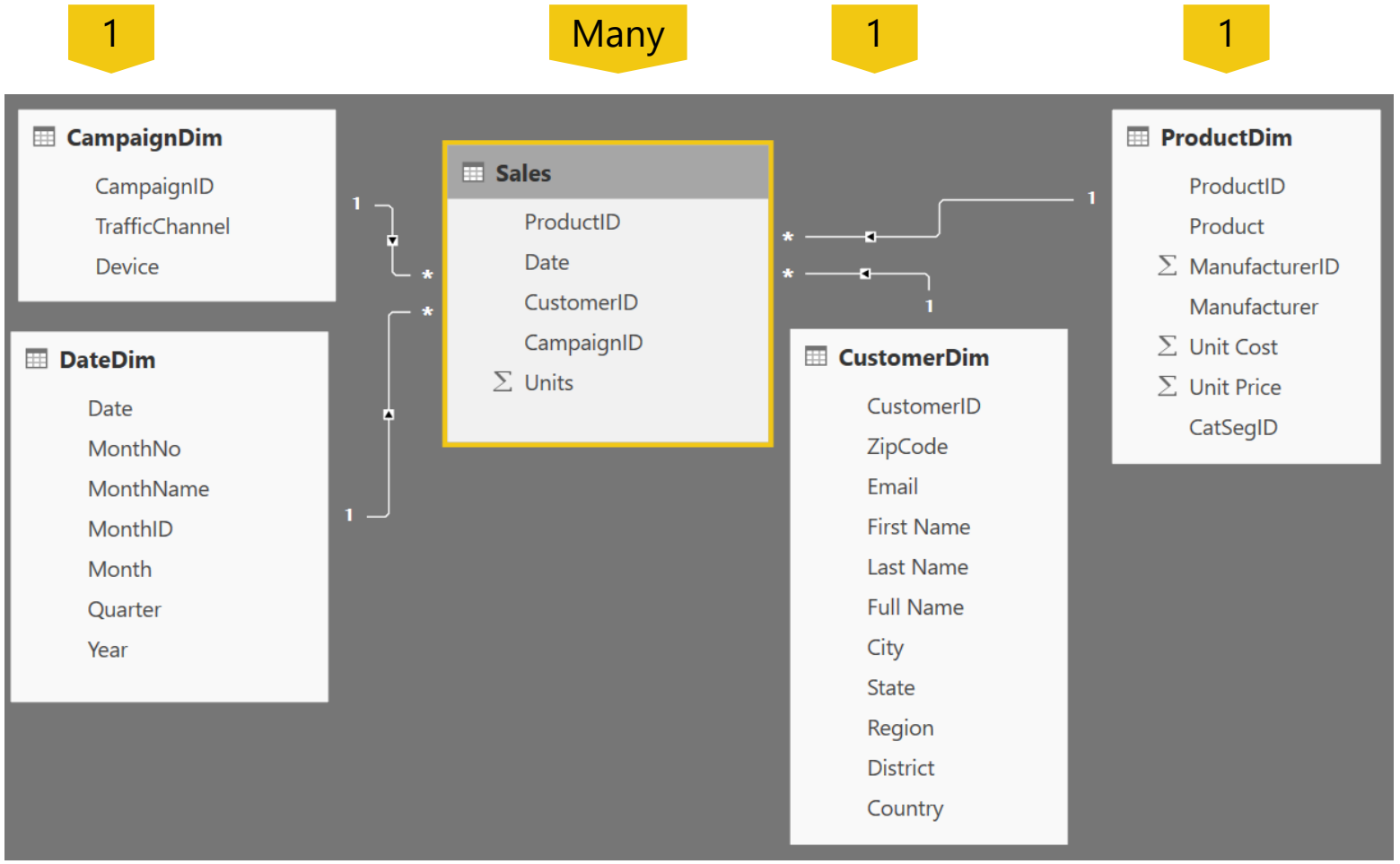


Relazioni

- La connessione tra due 2 tables (normalemente i fatti e le tabelle delle dimensioni) usando una colonna fra di loro

- 3 tipi di relazioni
  - 1 to Many
  - 1 to 1
  - Many to Many (Attenzione!!! Utilizzare una bridge table)

Star Schema



- **Tabella** dei fatti al centro
- Circondato da **dim**
- Sembra una "stella"
- La tabella dei fatti è il lato "Molti" della relazione (da una a molte)

Tipi di modelli di dato: Recap Fatti

Note importanti sulle nostre tabelle dei fatti

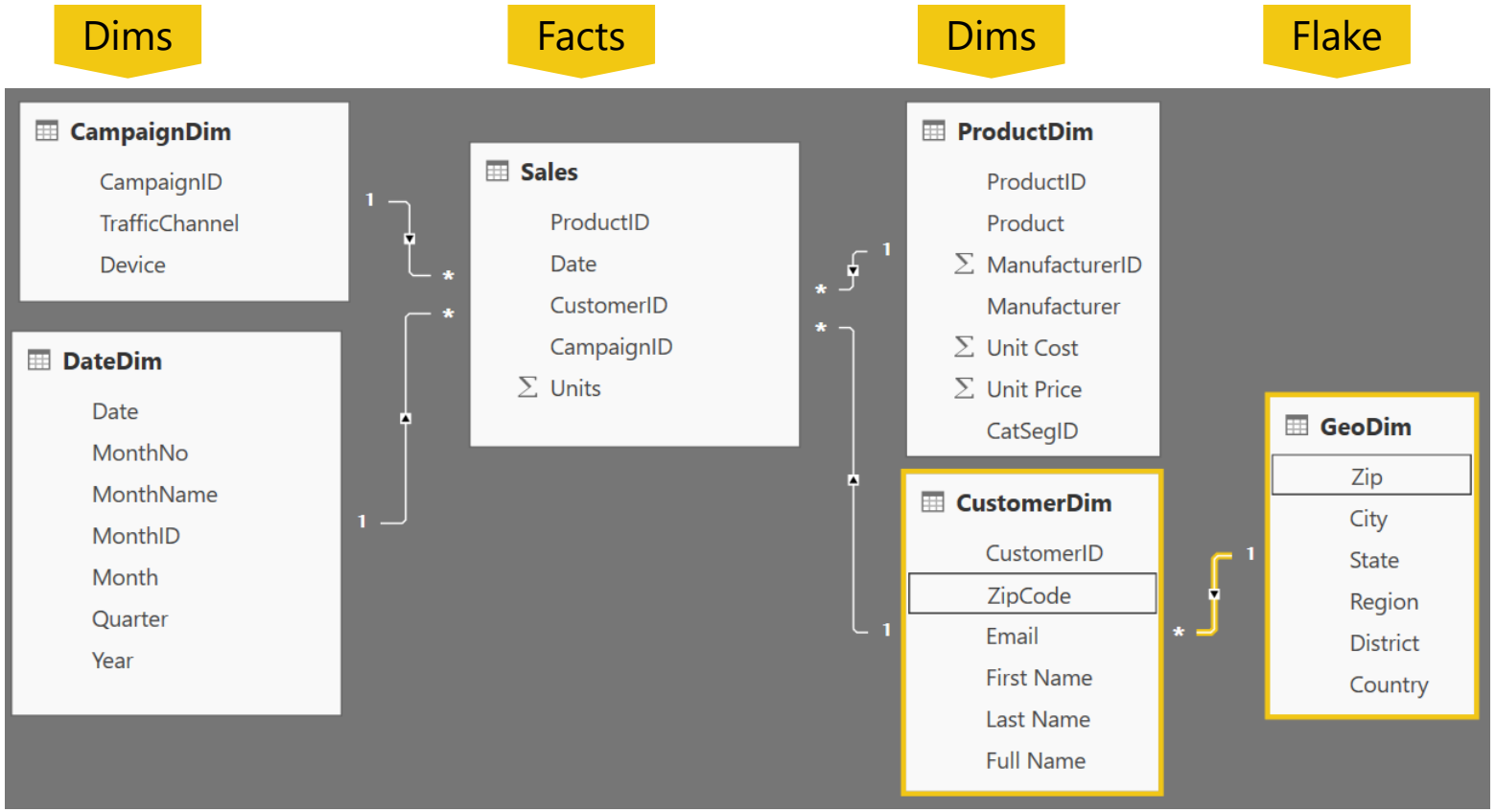
- Le tabelle dei fatti sono tipicamente molto grandi
  - Possiamo avere bilioni di vendite
  - Ma è diverso da avere bilioni di cutomers
- I fatti sono associati a numeri
  - La quantità di vendite
- Non tutti i numeri appartengono alla tabella dei fatti
  - Prezzo di un prodotto sta enlla tabella del prodotto (dimensione)
  - Quantità di prodotti venduti appartiene alla tabella dei Fatti
- Le misure dipicamente sono aggregazioni sulle tabelle SUM(Sales[Quantity])

Tipi di modelli di dato: Recap Dimensioni

- Note importanti sulle nostre tabelle delle dimensioni
  - Le dimensioni tipicamente sono piccole tabelle
    - 1 milione di righe per una dimensione è già un numero grande
  - Le dimensioni contengono molte stringhe
    - Nome del customer, indirizzo ecc…
    - Colore del prodotto e dimensione e ccc
  - Sono tipicamente quelle su cui si effettua lo slice
  - Quando dobbiamo calcolare in genere le usiamo per conteggiare su di loro
    - Numero di clienti
    - Numero di prodotti
  - Le dimensioni non sono relazionate ad altro.

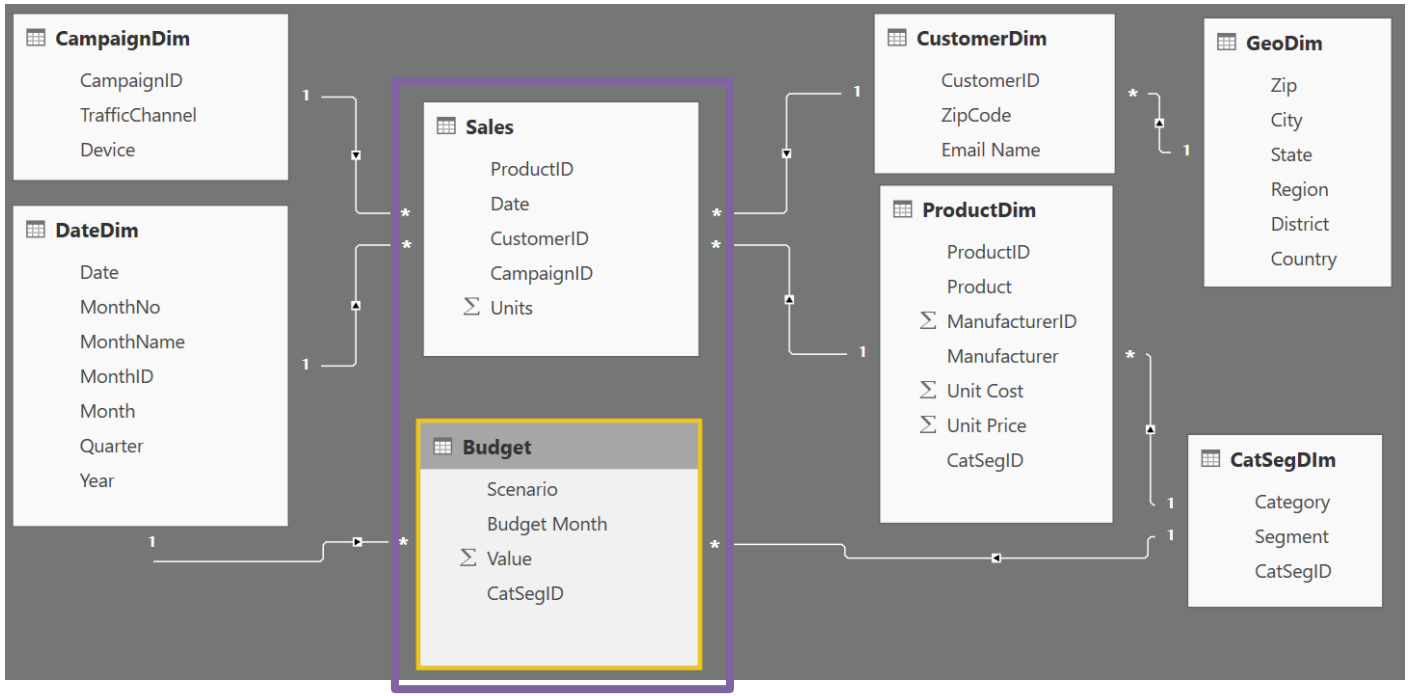
La differenza tra le tabelle dei fatti e le dimensioni sono il cuore di uno star schema in un modello di BI

Snowflake Schema (evitiamolo perchè richiede più attenzioni ☺ !!! Denormalizziamo tutto in una dimensione)



- Il centro è uno schema a stella
- Tabella dei fatti al centro
- Circondato da dim
- Dim "fiocco di neve" collegate ad altre dim
- Se ne hai molti, sembra un "fiocco di neve"
- Le tabelle Dim o Fact possono essere il lato "Molti" della relazione

Granularità & Fatti multipli



SalesFact (Giornaliera per prodotto)

BudgetFact (Mensile per categoria di prodotto e segment di prodotto)

- **Grain (granularità)** misure il livello di dettaglio della tabella

- Esempio:
- una riga per order o per oggetti
- per giorno o mese è il grain

- Se I tuoi fatti hanno differenti granularità, dovete dividerle in differenti tabelle dei fatti e connetterle tra loro da dimensioni condivise (Shared) al livello di granularità più basso di ogni singola tabella dei fatti

Granularità & Fatti multipli

- Tieni presente che le tue misure (calcoli) dovrebbero essere posizionate su una tabella dei fatti, ma in realtà non importa quale.

- Quando si ha un dato di fatto con il grain del mese, come il budget, è possibile creare una colonna calcolata che contiene il primo (o l'ultimo) giorno del mese da utilizzare nella creazione della relazione.

- Uno dei punti chiave è che ciò che fa il slice per tutti proviene dalle tabelle degli attributi (Dimensioni)

Basi di Data Modeling

Un'entità = una dimensione

- Più dimensioni sono spesso inutili e confuse

Regole empiriche:

- Numero di dimensioni
  - 10 dimensioni sono buone
  - 20 dimensioni sono probabilmente errate
  - 30 dimensioni sono totalmente errate
- Rapporto dimensione / attributo:
  - 1 dimensione con 20 attributi è buona
  - 20 dimensioni con 1 attributo non sono valide

Basi di Data Modeling

Multipli eventi uniti assieme

Order No	Customer	Date	Type	Amount
1	Contoso	01/01/2017	Order	100.00
1	Contoso	01/15/2017	Ship	60.00
1	Contoso	01/15/2017	Ship	40.00
2	Imageware	01/16/2017	Order	90.00
3	Imageware	01/20/2017	Order	45.00
4	Contoso	01/22/2017	Order	25.00

Modello corretto

Orders

Order No	Customer	Order Date	Amount
1	Contoso	01/01/2017	100.00
2	Imageware	01/16/2017	90.00
3	Imageware	01/20/2017	45.00
4	Contoso	01/22/2017	25.00

Shipments

Order No	Date	Amount
1	01/15/2017	60.00
1	01/15/2017	40.00

Perchè scegliere il giusto modello?

Se il modello non è quello giusto

- Il codice DAX tende ad essere molto complesso
- Le formule sono difficili da pensare
- La complessità si trasforma in problemi di prestazioni

Con il modello corretto

- Il codice DAX è semplice, come dovrebbe essere
- Le prestazioni sono eccezionali

Costruire il modello giusto richiede esperienza → altrimenti a cosa servo ☺

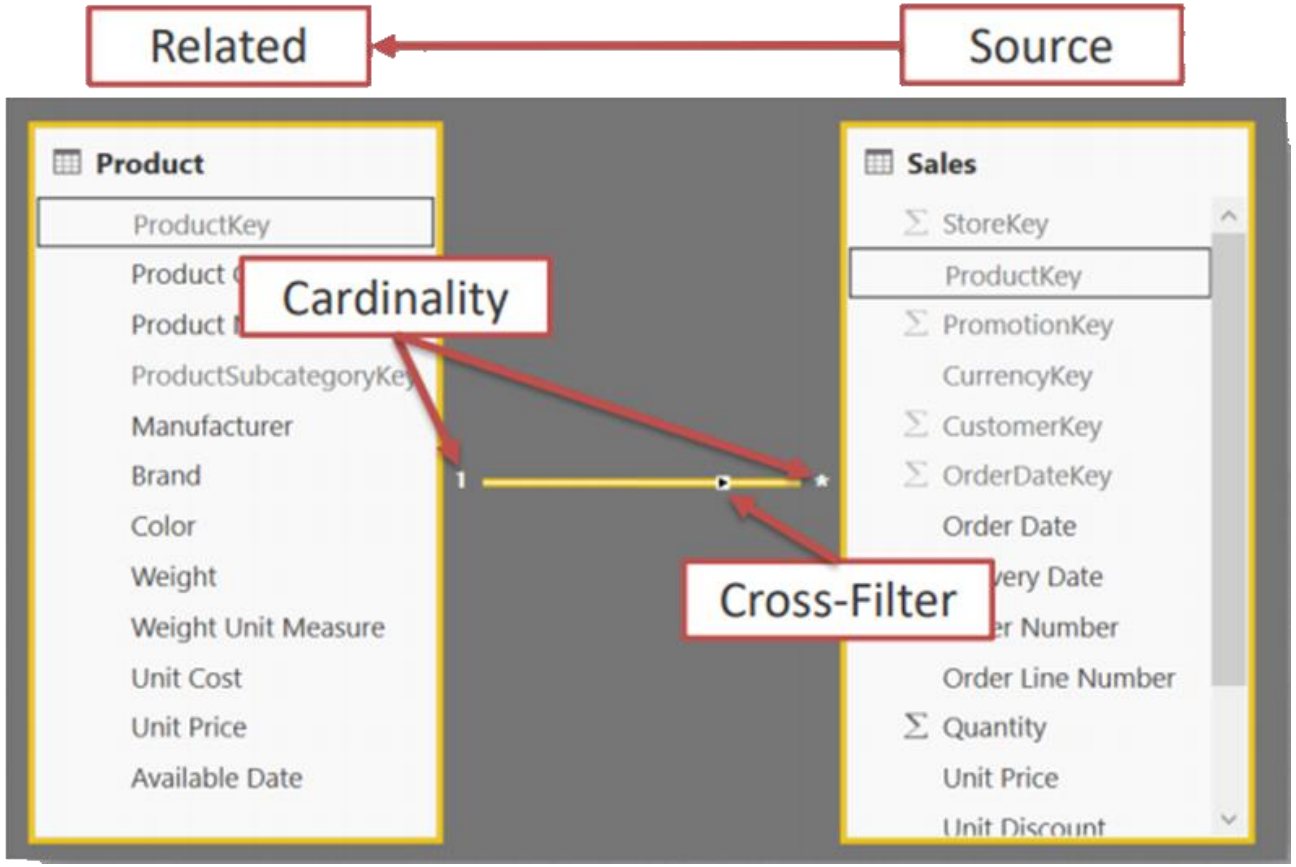
Perchè è importante avere il data model?

- Migliora la comprensibilità dei dati
- Aumenta le prestazioni dei processi
- Aumenta la resilienza al cambiamento

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 trattatere 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 si 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 relazionononare 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 dupplicati
  - 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) avvine 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 2-PQ e M

Laboratorio che consolida le basi di Power Query e aggiunge trasformazioni che ancora non abiamo visto **(da svolgere assiemein 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 verisione 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
  - Data + 1 = Rappresenta il giorno dopo
  - 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

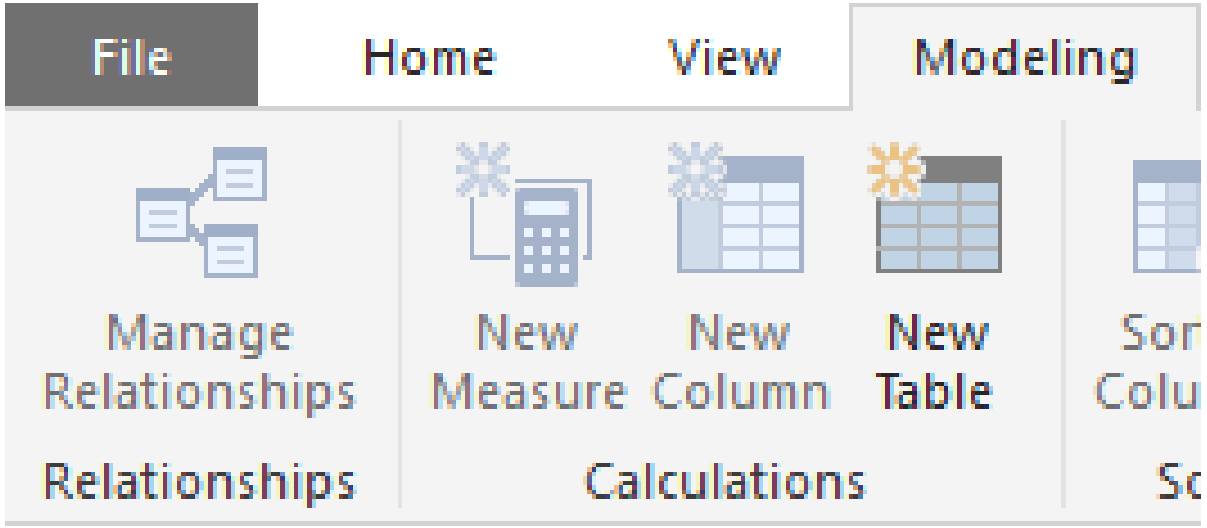
In questa dimostrazione vedremo le misure e le colonne calcolate

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 rapport
  - 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
  - AND (A, B) = A && B

Funzioni infromative

- Forniscono informazioni sulle espressioni
  - ISBLANK
  - ISNUMBER
  - ISTEXT
  - ISNONTEXT
  - ISERROR
- Non è molto utlie conoscere il tipo di espressioni che si valuta
- Sono difficili da usare perchè dipendono dal 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 arotondamento

- 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 comaprare valori
La seguente formula calcola il massimo del Sales Amount

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

Di seguito la corrsipondente 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 )
```

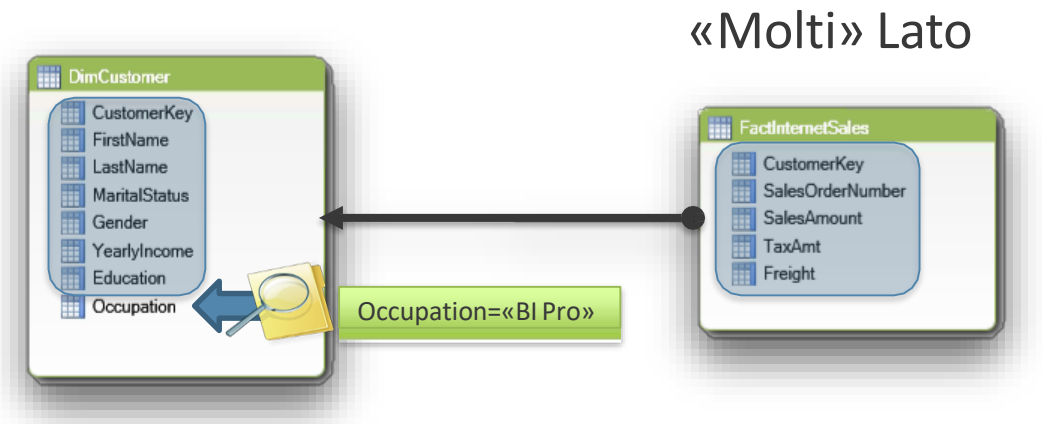
Funzioni Relazionali

- 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

FILTER e CROSSFILTER «Uno» Lato

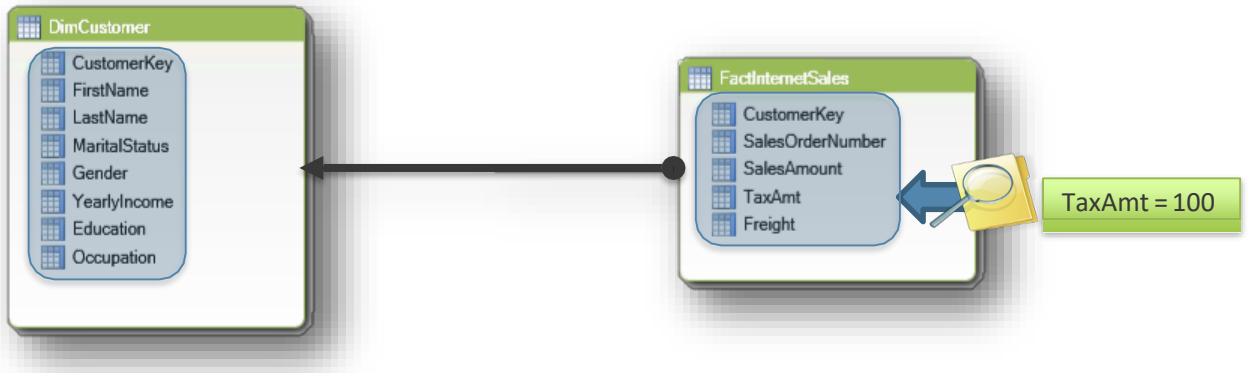


- ISFILTERED (DimCustomer[Occupation]) = TRUE ISFILTERED Indica se c'è il filtro diretto!!!
- ISFILTERED (DimCustomer[FirstName]) = FALSE
- ISCROSSFILTERED (DimCustomer[FirstName]) = TRUE
- ISCROSSFILTERED (DimCustomer) = TRUE (Solo per DAX 2015)
- ISFILTERED (FactInternetSales[SalesOrderNumber]) = FALSE
- ISCROSSFILTERED (FactInternetSales[SalesOrderNumber]) = TRUE
- ISCROSSFILTERED (FactInternetSales) = TRUE (Solo per DAX 2015)

FILTER e CROSSFILTER

«Uno» Lato

«Molti» Lato



- ISFILTERED (FactInternetSales[TaxAmt]) = TRUE
- ISFILTERED (FactInternetSales[SalesAmount]) = FALSE
- ISCROSSFILTERED (FactInternetSales[SalesAmount]) = TRUE
- ISCROSSFILTERED (FactInternetSales) = TRUE (Solo per DAX 2015)
- ISFILTERED (DimCustomer[CustomerKey]) = FALSE (il filtro non si propaga da M a 1 ☺)
- ISCROSSFILTERED (DimCustomer[CustomerKey]) = FALSE
- ISCROSSFILTERED (DimCustomer) = FALSE (Solo per DAX 2015)

BLANK in DAX

- Qualsiasi tipo di dati di colonna in DAX può avere un valore **vuoto**. Questo è il valore assegnato a una colonna quando l'origine dati contiene un valore **NULL**.
- La seguente tabella mostra qual è il risultato di diverse espressioni contenenti un valore vuoto. È possibile sostituire la chiamata alla funzione BLANK con qualsiasi espressione DAX che restituisce un valore vuoto, incluso un riferimento di colonna.

Expression	Result
`BLANK ()`	*Blank value*
`BLANK () = 0`	True
`BLANK () && TRUE`	False
`BLANK () || TRUE`	True
`BLANK () + 4`	4
`BLANK () - 4`	-4
`4 / BLANK ()`	*Infinite*
`BLANK () * 4`	*Blank value*
`BLANK () / 4`	*Blank value*
`INT ( BLANK () )`	*Blank value*

Comportamento del BLANK nelle comparazioni

- Quando si utilizza BLANK in un confronto a 0 allora **Blank() = 0**
- Quando si utilizza BLANK in un confronto a stringa vuota allora **blank() = stringa vuota**
- Se non vogliamo problemi il confronto di un'espressione BLANK richiede la funzione ISBLANK, che restituisce true ogni volta che il suo argomento è un'espressione vuota.
- La tabella seguente mostra un numero di esempi che illustra come si ottiene un match positivo

Expression	Result
`IF ( BLANK (), "true", "false" )`	false
`IF ( ISBLANK ( BLANK () ), "true", "false" )`	true
`IF ( BLANK () = True, "true", "false" )`	false
`IF ( BLANK () = False, "true", "false" )`	true
`IF ( BLANK () = 0, "true", "false" )`	true
`IF ( BLANK () = "", "true", "false" )`	true

Comparazioni con il BLANK

- La conversione automatica a zero o a una stringa vuota di un valore vuoto potrebbe avere effetti collaterali indesiderati. Ad esempio, si consideri la seguente espressione e il risultato per diversi valori di [measure].

```
=
IF (
    [measure] = BLANK (),
    "is blank",
    IF ( VALUE ( [measure] ) = 0, "is zero", "other" )
)
```

[measure]	Result
0	is blank
“”	is blank
BLANK	is blank

- La funzione VALUE converte una stringa in un numero

```
=
IF (
    VALUE ( [measure] ) = 0,
    "is blank",
    IF ( [measure] = BLANK (), "is zero", "other" )
)
```

[measure]	Result
0	is blank
“”	error
BLANK	is blank

Comparazioni con il BLANK

- Quando si utilizza SWITCH, il comportamento è diverso, poiché applica una logica di confronto diversa

Expression	Result
`SWITCH ( BLANK(), 0, "zero", BLANK(), "blank", True, "true", False, "false" )`	blank
`SWITCH ( BLANK(), 0, "zero", True, "true", BLANK(), "blank", False, "false" )`	blank
`SWITCH ( BLANK(), 0, "zero", True, "true", False, "false", BLANK(), "blank" )`	false

Conversione di una espressione tabellare

- Un'espressione di tabella convertita in un valore scalare genera un valore BLANK quando la tabella è vuota.

- Ad esempio, il FILTER sotto restituisce sempre una tabella vuota, e il risultato fornito da COUNTROWS è vuoto anziché zero.

- Aggiungere un valore zero se vogliamo che il risultato sia un numero che si può usare in una funzione SWITCH (Vedi sotto)

Expression	Result
`COUNTROWS ( FILTER ( Table, False ) )`	*Blank value*
`COUNTROWS ( FILTER ( Table, False ) ) + 0`	*0*

Conversione di una espressione tabellare

L'esempio della slide precedente è utile per capire come COUNTROWS è usato in uno statement SWICTH

Expression	Result
`SWITCH ( COUNTROWS ( FILTER ( Query1, False ) ), 0, "zero", 1, "one", "other" )`	other
`SWITCH ( COUNTROWS ( FILTER ( Query1, False ) ), BLANK(), "blank", 0, "zero", 1, "one", "other" )`	blank
`SWITCH ( COUNTROWS ( FILTER ( Query1, False ) ) + 0, BLANK(), "blank", 0, "zero", 1, "one", "other" )`	zero

Gestire BLANK in espressione booleana

Un BLANK in un'espressione logica viene convertito in un valore FALSE.

Come risultato di questa logica, nei seguenti esempi è possibile vedere che un valore BLANK non viene mai restituito da IF nel caso in cui il risultato debba essere un tipo di dati booleano.

Expression	Result
`IF ( 1 = 1, BLANK(), TRUE )`	*FALSE*
`IF ( 1 <> 1, BLANK(), TRUE )`	*TRUE*
`IF ( 1 = 1, TRUE, BLANK() )`	*TRUE*
`IF ( 1 <> 1, TRUE, BLANK() )`	*FALSE*

Laboratorio 1-Primi Passi con DAX

In questo Laboratorio riprenderemo il lavoro svolto nel Laboratorio del data model (0-Data Model) e ci aggiungeremo la parte della logica di business realizzata in DAX