Demo 1

Demo 2

**Chiedi alla classe di provare a classificarli nelle diverse tabelle e vedere come funzionano**. CHIEDERE DI CLASSIFICARLE

------------------------------------------ Slide 60--------------------------

Dobbiamo eseguire diverse trasformazioni per creare rapporti per la società VanArsdel.

Esaminare le differenze di granularità del modello di dati

* SalesFact
  + Per campagna
  + Per data (giorno)
  + Per prodotto e per categoria / segmento di prodotto
  + Per cliente - con zip di Geo
* BudgetFact
  + Per data (mese)
  + Per categoria di prodotto / segmento

------------------------------------------ Slide 53 --------------------------

CONTROLLO VERSIONE !!

StepxStep (Student Handout Slide 2)

**Obiettivo**: importare più fogli di lavoro da un file Excel in un solo passaggio

**Ottieni dati>** da Excel

1. **C: \ Power BI\_Adv\_M \ VansArsdel\_Actuals.xlsx**
2. Seleziona le seguenti tabelle (**tutte tranne CustomerDim**):

* CampaignDim
* GeoDim
* ProductDim
* DateDim
* Sales

1. In GeoDim, modifica il Tipo di dati [ZIP] in Testo
2. **Open VanArsdel\_Budget.xlsx file.** This file provides Budget and Forecast information by Product Category and Segment.

------------------------------------------------Slide 60----------------------------------

1. Creazione di una nuova tabella **Product** **Category** **Segment** **Dimension** by extracting Categories and Segments from the **ProductDim**
2. **Duplicate** the ProductDim query
3. **Highlight** [**Category**] and [**Segment**], and **Remove other columns**
4. **Highlight** [**Category**] and [**Segment**], and **Remove Duplicates**
5. Add Column > **Add Index Column starting at 1**, with column name [**CatSegID**]
6. **Reorder Columns**: [CatSegID], [Category], [Segment]
7. **Rename** the query “**CatSegDim**”

------------------------------------------------Slide 61----------------------------------

Graphical user interface, table

Description automatically generated with medium confidence

------------------------------------------------Slide 62----------------------------------

1. **Aggiungi il Product Category Segment ID calcolato prima nella ProductDim**
2. Select the ***ProductDim*** query
3. From Home Ribbon > **Merge Queries** > **Select** ProductDim
   1. From *ProductDim*, **highligh**t **[Category]** and **[Segment]**
   2. From *CatSegDim*, **highlight** **[Category]** and **[Segment]**
   3. Note the Join Kinds available, and leave **Left Outer**
   4. **Expand** the [NewColumn] > **Select** [CatSegID] and **deselect** “Use Original column name as prefix”
4. **Remove columns:** [Category], [Segment], [Manufacturer ID],and [Manufacturer].
   1. Hint: There is only one manufacturer name and one manufacturer ID, so we don’t need this information!
5. **Reorder columns**: [ProductID], [CatSegID], [Product], [Unit Price], [Unit Cost]

------------------------------------------------Slide 63----------------------------------

1. Creare il First, Last, and Full Name columns based on the Email Name column.
   * Use **Recent Sources** to get ***CustomerDim***
   * **Highlight** the **Email Name** column, then **Split Column >** by Delimiter, select **Custom** and enter **:** (due punti con lo spazio)
   * **Replace** per rimuovere le parentesi tonde **(** e **)**
   * **Add Column** per provare il comando **Text.PositionOf()** della virgola
     + **Add Column > Custom Column**
     + Name = “**Separator**”
     + Formula = Text.PositionOf([Email Name.2],",")   
       Graphical user interface, text, application

       Description automatically generated

* Usare la posizione della virgola trovata prima per divider Last Name e First Name  
  **Add Columns** per 3 volte per tre colonne **First Name, Last Name and Full Name***Last Name = Text.Start([Email Name.2],[Separator])  
  First Name = Text.Range([Email Name.2],[Separator]+2)  
  Full Name = [First Name] & " " & [Last Name]*
* Table

  Description automatically generated
* **Se non funziona facciamo trim prima del split by delimiter and do the “: “ (colon and space) then they can >”Trim** the blank space in front of the last name in the new column by right clicking on the column header **Email Name.2** and select **Transform>Trim”**

------------------------------------------------Slide 88 ---------------------------------------

1. Creazione della tabella di budget (Budget Fact)
   * **Import CSV document** “VanArsdel\_Budget.csv”
   * Rename query from VanArsdel\_Budget to **BudgetFact\_Data**
   * Remove Rows > **Remove Top Rows**, enter 3 (to remove the first 3 rows)

------------------------------------------------Slide 89 ---------------------------------------

1. Creazione dell’header dalla combinazione delle prime tre righe
   * **Duplicate** query BudgetFact\_Data **rename** to “**BudgetFact**”
   * **Keep Rows > Keep Top Rows,** enter **3** (to keep the first three rows)

Graphical user interface, application, table, Excel

Description automatically generated

------------------------------------------------Slide 90 ---------------------------------------

Notare gli errori nelle righe 1 e 2 poiché la formula non viene risolta in una data corretta. **Spiega come si avvolge la formula in try / altrimenti risolve l'errore.** Mostra data. Per il testo [Mese del budget], "M / D / Y") con tutte le lettere maiuscole **per discutere di come la formattazione della data sia sensibile al maiuscolo / minuscolo**.

Inoltre, stiamo usando ~ come delimitatore per semplificare la suddivisione del nostro scenario e della data in un secondo momento. Formattazione data: MM = mese; gg = giorno; aa = anno; hh = ora, mm = minuto; ss = Secondo

1. To Create a Header which combines the first three rows
   * **Transform** > **Transpose (tabella** “**BudgetFact**”)
   * **Add Column** to combine month and year into a date
     + **Add Column > Custom Column**
     + Name = “Budget Month”
     + Formula = *try Date.From ([Column3] & [Column2]) otherwise null*
     + **Add Column** to combine Month and Scenario
   * **Add Column > Custom Column**
     + Name = “*FullyCombinedHeader*”
     + Formula = *if Text.Length([Column3]) > 3 then [Column3] else [Column1] & "~" & Date.ToText([Budget Month], "M/D/YYYY")*
     + Hint: Day did not come through correctly, as it is case sensitive. Update to “MM/dd/yy”
   * **Remove** all columns except for [FullyCombinedHeader]
   * **Transform** > **Transpose** to transpose back to wide

Graphical user interface, text, application

Description automatically generated

------------------------------------------------Slide 92 ---------------------------------------

1. To Append the new header row to the data to create the new wide data table with a single header row

* **Append** alla query **BudgetFact** la query **BudgetFact\_Data**
* Use **First Row as Header** to promote the newly fixed header row
* Remove Rows > (to remove the first 3 rows – the old header rows)

**Table

Description automatically generated**

------------------------------------------------Slide 93---------------------------------------

1. Creazione della tabella Budget Fact
   * **Highlight** [**Category**] and [**Segment**] and **Transform** > **Unpivot Other Columns**
   * **Highlight** [**Attribute**] and navigate to Home > Split Column > By Delimiter > **“~”**
   * **Rename:** [Attribute.1] = “**Scenario**”, [Attribute.2] = “**Date**”, [Value] = “**Budget Amount**”
   * **Change** the Data Types: [Budget Amount] = Fixed Decimal, [Date] = Date

Table

Description automatically generated

------------------------------------------------Slide 90---------------------------------------

1. Merge the queries together to create the final **Budget Fact table**
   * Home > **Merge Queries** > Select **CatSegDim**
   * From the CatSegDim **highlight** both [**Category**] and [**Segment**]
   * From BudgetFact, **highlight** both [**Category**] and [**Segment**]
   * Show the Join Kinds available, and leave “Left Outer”
   * **Expand** [NewColumn] > **Select** “CatSegID” and **deselect “**Use Original column name as prefix”
   * **Remove**: [Category], [Segment]
   * **Reorder:** [CatSegID]**,** [Scenario], [Date], [Budget Amount]
   * **Disable** the load of BudgetFact\_Data

Demo 3

1. pbi-sqlserver.database.windows.net user:marcopozzan pwd:Pa$$w0rd
2. Descriviamo il modello

Diagram

Description automatically generated

1. Nella tabella dei fatti: **SalesHeader** esiste una colonna che rappresenta lo sconto totale (**TotalDiscount**) perché il business ha fatto si che lo sconto totale venisse applicato all’ordine
2. Esiste una relazione tra le due tabelle dei fatti
3. Una riga di testata (**SalesHeader)** ha più righe di dettaglio e la colonna **Order number** è replicata più volte sulle righe della tabella del **SalesDetail**
4. **SE la tabella dei fatti SalesHeader fosse una dimensione avremmo uno snowflake schema. NON E’ UNO SNOWFLAKE schema perché la SalesHeader è una tabella dei fatti in quanto vi è anche la colonna del TotalDiscount**
5. **DOMANDA:** Quale è il problema di questo modello?
   1. **Entrambe le tabelle dei fatti contengono informazioni e se decidiamo di aggregare sull’intestazione si produce un risultato non corretto quando facciamo lo slice su dimensioni collegate all’altra tabella dei fatti (TAB intro del foglio 01- Header-Details**

Table

Description automatically generated

* 1. Che cosa è successo? Spiegare il motivo del filtro 😊

1. COME RISOLVIAMO IL MODELLO?
   1. SOLUZIONE 1: Usiamo le relazioni bidirezionale
      1. Mettiamo nel modello la bidirezionalità
      2. Ci spostiamo sulla TAB (**bidirezionalità**)
      3. Sembra funzionare bene ma il totale di tutti i brand è **458,265.70** non **395,632,29** che è corretto
      4. Questo succeede perchè quando settiamo un filtro sui prodotti si setta un insieme di ordini… ora se un ordine contiene prodotti che sono in più brand cosa succede sul total discount? Viene sommato per alcuni brand ma anche per altri e quindi si sommano gli stessi numeri più volte
   2. SOLUZIONE 2: Calcolare lo sconto come una percentuale.
      1. Calcoliamo lo sconto a livello di **Testata** come uno sconto percentuale e NON un valore fisso. Così allochiamo in modo PROPORZIONALE lo sconto **(** lo script DAX è in DiscountPct.txt**)**

Graphical user interface, table

Description automatically generated

* + 1. Text, letter

       Description automatically generated
    2. Disabilitiamo la bidirezionalità
       1. Cambiamo la misura DiscoutValue che è **DiscountValue = SUM ( SalesHeader[TotalDiscount] )** nella somma della quantità per il prezzo a livello di dettaglio per lo sconto in percentuale **calcolato sopra dell’ordine collegato al dettaglio che sto iterando!!!! Questo ci permette di calcolare lo sconto in un livello di dettaglio di riga (** lo script DAX è in DiscountValueCorrect.txt**)**
       2. Text, letter

          Description automatically generated
       3. Il risultato è il seguente
       4. Table

          Description automatically generated
  1. SOLUZIONE 3: La soluzione 2 non è male ma dobbiamo fare un po' di ginnastica con DAX questo perché il nostro modello non è uno star schema
     1. Dobbiamo!!! Denormalizzare lo sconto nella tabella dei dettaglio degli ordini
     2. QUESTO TRASFORMA la nostra SalesHeader in una Dimensione!!!!
     3. Diagram

        Description automatically generated
     4. Per fare questo lo posso fare in DAX creando una colonna calcolata nella tabella dei SalesDetail che si chiama **LineDiscount = SalesDetail[Quantity] \* SalesDetail[UnitPrice] \* RELATED(SalesHeader[Discount Pct]) (** lo script DAX è in LineDiscount.txt**)**
     5. Quindi ora la mia misura DAX che calcola lo sconto è più semplice **DiscountValue = SUM ( SalesHeader[LineDiscount] ) (**Lo script DAX è in DiscountValue.txt**)**
     6. **ORA ABBIAMO TRASFORMATO IL NOSTRO MODELLO DI PARTENZA IN UNO SNOWFLAKE SCHEMA!!!!!!!**
  2. SOLUZIONE 4: Otteniamo uno STAR SCHEMA. Una volta denormalizzato correttamente, il modello diventa uno schema a stella di nuovo.
     1. Quello che devo cambiare è la tabella delle vendite Sales che sarà una fusione dei dettagli e delle testate attraverso il merge