**W15D2 – Modellare i dati**

**Importanza del modello dati**

Progettare un modello dati è una delle attività più critiche nello sviluppo di soluzioni in Power BI.  
Un buon modello ha un impatto diretto su:

* **Qualità delle informazioni**: i dati esposti sono coerenti, affidabili e privi di ambiguità.
* **Performance**: un modello ben progettato permette di lavorare in maniera fluida anche su dataset grandi.
* **Aggregazioni**: semplifica le operazioni di roll-up (sintesi a livelli più alti, ad esempio sommare vendite per anno) e drill-down (analisi di dettaglio, ad esempio scendere da anno → mese → giorno).
* **Manutenzione**: un modello pulito e chiaro è più facile da mantenere e aggiornare nel tempo.

**Normalizzazione e denormalizzazione**

Questi due concetti derivano dal mondo dei database relazionali, ma in Power BI hanno un impatto diretto sulle performance e sulla fruibilità.

* **Normalizzazione**: i dati vengono suddivisi in più tabelle, ognuna con un ruolo specifico, evitando ridondanze. Ad esempio, in un database normalizzato, le informazioni dei clienti sono in una tabella distinta dalle vendite. Questo migliora la coerenza e riduce gli errori.
* **Denormalizzazione**: le informazioni vengono invece “accorpate” in tabelle più grandi, anche a costo di ripetere alcuni dati. In Power BI è spesso preferibile lavorare con tabelle denormalizzate perché questo semplifica i calcoli e velocizza le analisi, riducendo il numero di join da fare in fase di visualizzazione.

In sintesi: nei database transazionali si privilegia la normalizzazione per garantire integrità e coerenza, mentre nei modelli analitici (come in Power BI) si tende alla denormalizzazione per ottimizzare le performance delle query.

**Star Schema**

Lo **Star Schema** è il modello logico più diffuso in ambito BI, perché ottimizza l’analisi dei dati.  
Si basa su due tipi di tabelle:

* **Fatti**: contengono gli eventi da misurare (es. vendite, acquisti, ordini, movimenti di magazzino). Sono numeriche e soggette ad aggregazioni.
* **Dimensioni**: descrivono i fatti da diversi punti di vista (es. prodotto, cliente, area geografica, tempo). Servono per filtrare e segmentare i fatti.

Le relazioni tra fatti e dimensioni sono generalmente **uno-a-molti** e le dimensioni sono di solito **denormalizzate**, per rendere più immediate le analisi.

**Relazioni**

Una **relazione** è un legame logico tra due tabelle, che consente ai filtri applicati su una tabella di influenzarne un’altra.

In Power BI esistono diversi tipi di relazioni:

* **One-to-many (1:\*)**
* **Many-to-one (\*:1)**
* **One-to-one (1:1)**
* **Many-to-many (*:*)**, supportata in modo esplicito in Power BI.

Il modo in cui i filtri si propagano tra le tabelle dipende dalla **cardinalità** e dalla **direzione della relazione**.

**Cardinalità**

La cardinalità indica il grado di univocità dei valori di una colonna. In altre parole, quante volte i valori si ripetono.

* Se in una colonna ogni valore è unico, si parla di cardinalità **alta**.
* Se i valori si ripetono spesso, si parla di cardinalità **bassa**.

In Power BI la cardinalità non è solo una caratteristica dei dati, ma influenza direttamente la **direzione di propagazione dei filtri**: stabilisce cioè come i filtri applicati a una tabella influenzano le altre.

**Direzione di propagazione dei filtri**

La **propagazione dei filtri** (cross-filtering direction) definisce come i filtri applicati su una tabella influenzano i dati delle altre.

* **Direzione singola**: il filtro si propaga dal lato “uno” della relazione al lato “molti”. Ad esempio, se filtro una tabella “Clienti”, questo influisce sulla tabella “Vendite” collegata, ma non il contrario.
* **Direzione doppia**: il filtro si propaga in entrambe le direzioni. Usata quando serve che le tabelle si influenzino a vicenda (es. tra Prodotti e Regioni passando dalle Vendite).

**Esempio (semplificato come per un bambino)**

Immagina tre scatole:

* una scatola **Regioni**,
* una scatola **Vendite**,
* una scatola **Prodotti**.

Ogni vendita collega un prodotto a una regione.

Se metto un filtro sulla scatola **Regioni** (es. “Italia”), automaticamente tutte le vendite mostreranno solo quelle avvenute in Italia. Questo è il filtro “singolo”.

Ma se volessi sapere **quali prodotti** sono stati venduti in Italia, dovrei far arrivare il filtro anche fino alla scatola Prodotti. Questo avviene solo se imposto la relazione come **doppia direzione**.

La propagazione doppia, però, ha un costo: Power BI tratta le tabelle quasi come un’unica tabella, e quindi le prestazioni peggiorano. Per questo va usata solo quando è davvero necessaria.

**Misure implicite vs misure DAX**

* **Misura implicita**: quando in un visual inserisco un campo numerico e Power BI applica automaticamente un’aggregazione (somma, media, conteggio). È “implicita” perché viene creata in automatico, senza che l’utente scriva nulla.
* **Misura DAX**: è una misura definita manualmente dall’utente con linguaggio DAX (Data Analysis Expressions). Questo consente di costruire calcoli più complessi, personalizzati e riutilizzabili.

Esempio:

* Trascinando il campo “Vendite” in una tabella, Power BI calcola la **somma implicita** delle vendite.
* Creando una misura TotaleVendite = SUM(Sales[SalesAmount]), si ottiene una **misura DAX** esplicita, che può essere usata in più visualizzazioni con maggiore controllo.

**Role-Playing Dimensions**

In Power BI è possibile avere solo una relazione attiva tra due tabelle. Se ci sono più relazioni possibili (es. una tabella Ordini con “DataOrdine” e “DataSpedizione”), solo una può essere attiva. Le altre rimangono inattive, ma possono essere attivate temporaneamente con funzioni DAX (es. USERELATIONSHIP).

**Proprietà dei campi**

Best practice per la gestione dei campi in Power BI:

* **Definire l’aggregazione corretta** per i campi numerici (somma, media, conteggio, ecc.), altrimenti Power BI applica di default la somma o il conteggio.
* **Categorizzare i campi**, ad esempio per le mappe è fondamentale che campi geografici siano riconosciuti come tali.
* **Rinominare e ordinare correttamente** i campi, per migliorare chiarezza e usabilità dei report.

**Gerarchie**

Ogni dimensione contiene spesso una **gerarchia**, che rappresenta livelli di dettaglio crescenti.  
Esempio classico: una dimensione “Tempo” può avere la gerarchia **Anno → Trimestre → Mese → Giorno**.

Le gerarchie permettono di eseguire agevolmente **drill-down** (scendere nel dettaglio) e **roll-up** (risalire al livello di sintesi).

**APPROFONDIMENTI PERSONALI**

**Normalizzazione vs Denormalizzazione in Power BI**

* **Normalizzazione**:  
  È la tecnica tipica dei database transazionali. I dati vengono suddivisi in tante tabelle collegate fra loro da chiavi, in modo da ridurre al minimo la ridondanza. Ad esempio:
  + Una tabella "Clienti" contiene solo le informazioni anagrafiche.
  + Una tabella "Ordini" contiene solo i riferimenti agli ordini con il codice cliente.
  + Una tabella "Prodotti" contiene solo i dettagli dei prodotti.

Vantaggi: integrità e coerenza dei dati.  
Svantaggi in Power BI: se tutto è troppo normalizzato, ogni visual richiede molte join tra tabelle e le performance calano.

* **Denormalizzazione**:  
  Consiste nel raggruppare più informazioni in un’unica tabella anche a costo di duplicare dei dati. In Power BI è preferibile avere tabelle denormalizzate perché:
  + il motore analitico lavora meglio con meno relazioni;
  + le misure e i calcoli sono più semplici;
  + i report risultano più veloci.

Per questo si tende ad avere **dimensioni denormalizzate** (es. tabella "Prodotti" che include categoria, sottocategoria, colore, ecc.) collegate alla tabella dei fatti.

**Cardinalità**

La **cardinalità** misura l’univocità dei valori in una colonna:

* **Alta cardinalità**: ogni valore è quasi unico (es. numeri di serie dei prodotti).
* **Bassa cardinalità**: i valori si ripetono molto (es. genere: maschio/femmina).

In Power BI la cardinalità è importante perché determina la direzione in cui i filtri si propagano:

* Dal lato con pochi valori unici (lato “1”) → al lato con molti valori ripetuti (lato “\*”).
* È la regola tipica delle relazioni 1:\* tra dimensioni e fatti.

Esempio: un campo **IDCliente** unico nella tabella “Clienti” è collegato a più righe nella tabella “Vendite”. Quando filtro un cliente, si riflettono tutte le sue vendite.

**Esempio pratico di propagazione dei filtri (parte della tabella Region, Sales, Product)**

Immaginiamo tre tabelle:

* **Region** (con i nomi delle regioni, es. Italia, Canada, Australia).
* **Sales** (con le vendite e le chiavi che collegano prodotto e regione).
* **Product** (con l’elenco prodotti).

Se creo una tabella che mostra per ogni regione:

* Somma delle vendite (SalesAmount)
* Conteggio dei prodotti (ProductKey)

Accade questo:

* Il filtro su **Region** si propaga correttamente a **Sales**, quindi le vendite per regione sono corrette.
* Ma il filtro **non si propaga automaticamente a Product**, quindi il conteggio dei ProductKey non tiene conto della regione.

In altre parole: ottengo vendite filtrate per regione, ma il conteggio prodotti risulta globale (uguale per tutte le regioni).

Per risolvere il problema posso impostare la relazione Region–Sales–Product come **bidirezionale**: in questo modo, quando filtro una regione, il filtro passa da Sales anche a Product. Così vedo per ogni regione il numero corretto di prodotti venduti.

**Direzione di propagazione dei filtri (spiegata semplice)**

Pensa ai filtri come a un rubinetto che fa scorrere l’acqua:

* **Filtro a senso unico**: l’acqua scorre solo in una direzione (da “uno” a “molti”). Se filtro i clienti, vedo le loro vendite; ma se filtro le vendite non riesco a risalire ai clienti.
* **Filtro a doppio senso**: l’acqua scorre in entrambe le direzioni. Se filtro i clienti, vedo le vendite; se filtro le vendite, vedo i clienti.

Questa doppia propagazione rende più comoda l’analisi, ma consuma più risorse e può rallentare il report. È quindi da usare solo quando indispensabile.

**Differenza tra misura implicita e misura DAX**

* **Misura implicita (dato aggregato automaticamente)**  
  Quando trascino un campo numerico in una visualizzazione, Power BI lo aggrega da solo (di default con una somma, o un conteggio se non può sommare).  
  Esempio: trascino “Vendite” in una tabella → Power BI mostra la somma delle vendite.
* **Misura esplicita (scritta in DAX)**  
  Creo io la misura con una formula DAX.  
  Esempio: TotaleVendite = SUM(Sales[SalesAmount]).

Differenza:

* Le misure implicite sono veloci e automatiche, ma meno flessibili.
* Le misure DAX sono riutilizzabili, più potenti e permettono calcoli avanzati (percentuali, cumulati, medie mobili, condizioni complesse).