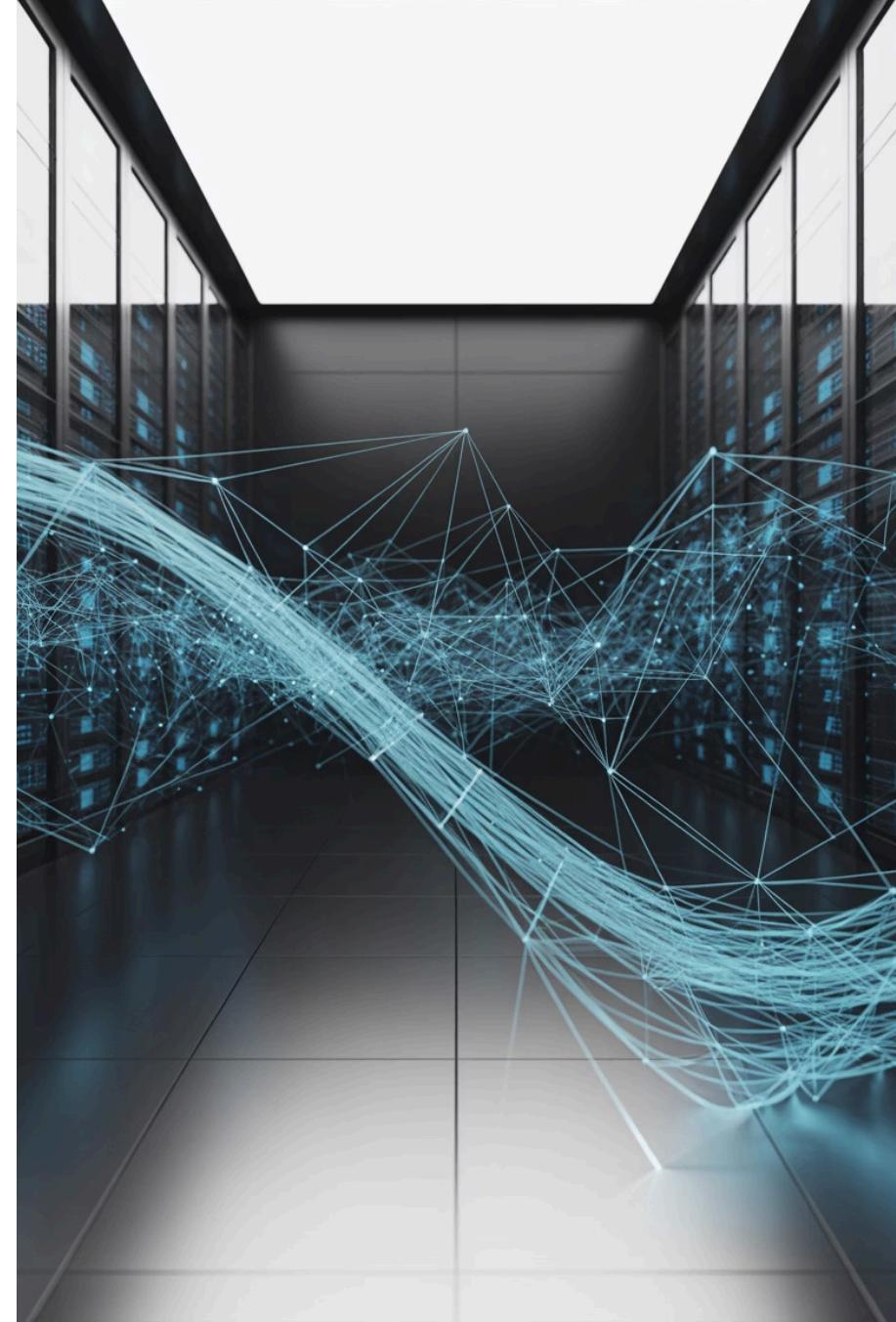


Query OLAP per l'Analisi delle Vendite

Un percorso pratico attraverso le query ROLAP per l'analisi multidimensionale dei dati di vendita. Imparerai a sfruttare la potenza dello schema a stella per estrarre insight strategici dal Data Warehouse TechStore.



Struttura del Corso

01

Esercizi Guidato Pratico

Workshop completo con step progressivi per costruire analisi complesse

02

Es Finale

Integrazione delle competenze apprese in due query multidimensionali avanzata

Creare Analisi OLAP sul Data Warehouse TechStore

100%

Pratico

Approccio hands-on con query reali

Step 1: Comprendere lo Schema

Tabella dei Fatti

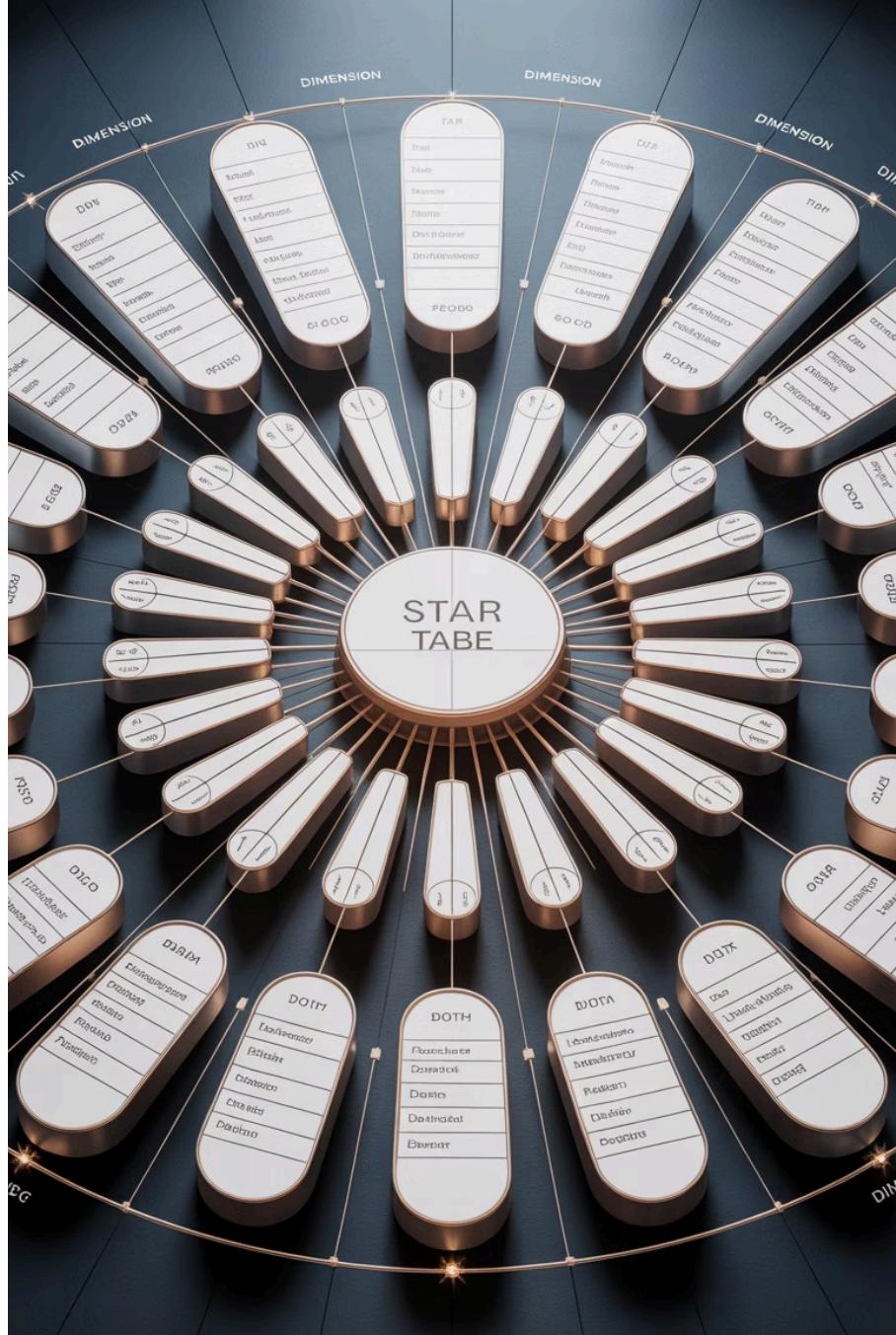
fact_vendite contiene gli eventi misurabili: ogni riga rappresenta una transazione con metriche quantitative (ricavi, quantità).

Tabelle Dimensionali

Le dimensioni forniscono il *contesto*: clienti, prodotti, date e spedizioni. Permettono di analizzare i fatti da diverse prospettive.

💡 🔑 Chiavi Surrogate

Identificate dal suffisso `_sk`, le chiavi surrogate (es. `cliente_sk`, `prodotto_sk`) collegano la fact table alle dimension table in relazioni uno-a-molti (1-N).



Obiettivi Didattici dell'Esercizio



Navigazione Schema a Stella

Comprendere la struttura del Data Warehouse: tabella dei fatti centrale circondata dalle dimensioni di analisi.



Join Multidimensionali

Padroneggiare le tecniche di unione tra fact table e dimension table utilizzando le chiavi surrogate.



Aggregazioni Complesse

Eseguire calcoli aggregati su più dimensioni simultaneamente per ottenere insight di business.



Costruzione KPI

Creare indicatori chiave di performance: ricavi totali, ticket medio, tempi di consegna e altri metriche strategiche.

Prima Query — Ricavi Totali

Obiettivo

Calcolare i ricavi totali generati dall'azienda, sommando tutte le transazioni registrate nella fact table.

```
SELECT  
    SUM(ricavi) AS ricavi_totali  
FROM fact_vendite;
```

Questa query base introduce il concetto di **aggregazione** senza utilizzare ancora le dimensioni. È il punto di partenza per comprendere come interrogare la tabella dei fatti.

Ricavi per Anno

```
SELECT
  dd.anno,
  SUM(f.ricavi) AS ricavi_totali
FROM fact_vendite f
JOIN dim_data dd
  ON f.data_sk = dd.data_sk
GROUP BY dd.anno
ORDER BY dd.anno;
```

1

JOIN con Dimensione

Collegamento della fact table con dim_data attraverso la chiave data_sk

2

Raggruppamento

GROUP BY permette di aggregare i ricavi per ciascun anno distinto

3

Analisi Temporale

Visualizzazione dell'evoluzione dei ricavi anno su anno



Ricavi per Categoria

```
SELECT  
    dp.categoria,  
    SUM(f.ricavi) AS ricavi_totali  
FROM fact_vendite f  
JOIN dim_prodotto dp  
    ON f.prodotto_sk = dp.prodotto_sk  
GROUP BY dp.categoria  
ORDER BY ricavi_totali DESC;
```

In questo step introduciamo la dimensione prodotto per rispondere alla domanda:
"Quale categoria vende di più?"

L'ordinamento decrescente (DESC) posiziona in alto le categorie più performanti, facilitando l'identificazione immediata dei segmenti di prodotto più redditizi per l'azienda.

Query 1: Vendite per Categoria e Anno

Questa query fondamentale risponde alla domanda: quale è il ricavo per anno per categoria?

Ci permette di identificare le categorie di prodotto che generano i maggiori ricavi all'interno di ciascun anno. Analizzeremo i dati per comprendere le dinamiche di mercato e guidare decisioni strategiche sull'assortimento e sulle promozioni.



Query 1: Vendite per Categoria e Anno

```
SELECT
    dd.anno,
    dp.categoria,
    SUM(f.ricavi) AS ricavi_totali
FROM fact_vendite f
JOIN dim_data dd
    ON f.data_sk = dd.data_sk
JOIN dim_prodotto dp
    ON f.prodotto_sk = dp.prodotto_sk
GROUP BY dd.anno, dp.categoria
ORDER BY dd.anno, ricavi_totali DESC;
```

Cosa Mostra

Questa query identifica quali categorie di prodotto generano maggiori ricavi in ciascun anno, permettendo di individuare trend di lungo periodo e categorie strategiche.

Caso d'Uso

Pianificazione strategica dell'assortimento e allocazione budget marketing per categoria.



Query 2: Andamento Mensile per Categoria

Questa query, variante della precedente, si concentra sull'analisi dei ricavi per categoria di prodotto su base mensile. Ci permette di identificare pattern stagionali, tendenze a breve termine e anomalie nelle performance di vendita, fornendo insight più granulari rispetto all'analisi annuale.

Comprendere questi andamenti è cruciale per ottimizzare la gestione delle scorte, pianificare campagne promozionali mirate e prevedere la domanda con maggiore accuratezza.

Query 2: Andamento Mensile per Categoria

```
SELECT
    dd.anno,
    dd.mese,
    dp.categoria,
    SUM(f.ricavi) AS ricavi_totali
FROM fact_vendite f
JOIN dim_data dd ON f.data_sk = dd.data_sk
JOIN dim_prodotto dp ON f.prodotto_sk = dp.prodotto_sk
GROUP BY dd.anno, dd.mese, dp.categoria
ORDER BY dd.anno, dd.mese;
```

Granularità Mensile

L'aggiunta della dimensione mese permette di analizzare la stagionalità e i pattern ciclici delle vendite.

Visualizzazione Ideale

Perfetta per generare grafici a linee temporali che mostrano l'evoluzione delle vendite per categoria nel tempo.



Query 3: Top 10 Clienti per Ricavi

Questa query fondamentale identifica i dieci clienti che hanno generato il maggiore ricavo totale nel periodo analizzato. Ci permette di focalizzare l'attenzione sui segmenti di clientela più preziosi.

Comprendere chi sono i nostri clienti top è cruciale per sviluppare strategie di fidelizzazione mirate, offrire promozioni personalizzate e allocare le risorse di marketing in modo più efficace. Questi insight sono fondamentali per la crescita sostenibile del business.



Query 3: Top 10 Clienti per Ricavi

```
SELECT
    dc.nome,
    dc.cognome,
    SUM(f.ricavi) AS ricavi_totali
FROM fact_vendite f
JOIN dim_cliente dc
    ON f.cliente_sk = dc.cliente_sk
GROUP BY dc.nome, dc.cognome
ORDER BY ricavi_totali DESC
LIMIT 10;
```



Principio di Pareto

Questa query applica il principio 80/20: spesso il 20% dei clienti genera l'80% dei ricavi. Identificare questi clienti VIP è fondamentale per strategie di retention e programmi di loyalty.

Query 4: Ricavi Medi per Prodotto

Questa query calcola il ricavo medio generato da ciascun prodotto. Comprendere il valore medio di ogni vendita per prodotto è essenziale per ottimizzare le strategie di prezzo, valutare la performance del catalogo e identificare i prodotti "core" che trainano i margini.

Questi insight aiutano i team di prodotto e marketing a prendere decisioni informate su assortimento, promozioni e posizionamento.



Query 4: Ricavi Medi per Prodotto

```
SELECT
    dp.nome AS prodotto,
    dp.categoria,
    AVG(f.ricavi) AS ricavo_medio
FROM fact_vendite f
JOIN dim_prodotto dp
    ON f.prodotto_sk = dp.prodotto_sk
GROUP BY dp.nome, dp.categoria
ORDER BY ricavo_medio DESC;
```

Analisi del Ticket Medio

Questa query calcola il valore medio di transazione per ciascun prodotto, rivelando quali articoli hanno il prezzo unitario più elevato o vengono acquistati in quantità maggiori.

Applicazioni Pratiche

- Identificazione prodotti premium
- Strategie di upselling
- Ottimizzazione mix prodotti



Query 5: Performance dei Corrieri

Questa query si concentra sulla misurazione dell'efficienza logistica, calcolando il tempo medio di consegna per ciascun corriere. Comprendere quali corrieri offrono le performance migliori in termini di velocità è fondamentale per ottimizzare la supply chain e garantire la soddisfazione del cliente.

Analizzeremo i dati per identificare i partner logistici più affidabili e prendere decisioni strategiche per migliorare i tempi di consegna complessivi, riducendo i costi e aumentando la fedeltà dei clienti.

Query 5: Performance dei Corrieri

```
SELECT
    ds.corriere,
    AVG(ds.tempo_consegna_giorni) AS tempo_medio,
    COUNT(*) AS num_spedizioni
FROM fact_vendite f
JOIN dim_spedizione ds
    ON f.spedizione_sk = ds.spedizione_sk
GROUP BY ds.corriere
ORDER BY tempo_medio;
```

KPI Logistico Cruciale

Il tempo medio di consegna è un indicatore diretto della customer satisfaction e dell'efficienza operativa della supply chain.

Decisioni Data-Driven

Permette di confrontare oggettivamente i corrieri e prendere decisioni informate sui partner logistici da privilegiare o sostituire.





Query 6: Segmentazione per Fascia d'Età

Questa query ci permette di segmentare la nostra clientela in base a fasce d'età specifiche, analizzando come il comportamento d'acquisto varia tra i diversi gruppi demografici. Comprendere l'età media o la distribuzione per fasce d'età dei nostri clienti è fondamentale per personalizzare le strategie di marketing, lo sviluppo prodotti e le comunicazioni.

Questi insight demografici sono preziosi per identificare i segmenti di clientela più redditizi e per adattare l'offerta alle esigenze e preferenze di ogni gruppo, massimizzando l'engagement e il valore a lungo termine del cliente.

Query 6: Segmentazione per Fascia d'Età

```
SELECT  
dc.fascia_eta,  
SUM(f.ricavi) AS ricavi  
FROM fact_vendite f  
JOIN dim_cliente dc  
ON f.cliente_sk = dc.cliente_sk  
GROUP BY dc.fascia_eta  
ORDER BY ricavi DESC;
```

Marketing Analytics Avanzato

La segmentazione demografica è fondamentale per personalizzare le campagne marketing e ottimizzare il targeting pubblicitario.

Insight Strategici

Identifica quali segmenti demografici hanno il maggior valore economico, permettendo di concentrare gli investimenti marketing sui target più profittevoli e di sviluppare strategie di acquisizione mirate.

Query 7: Serie Temporale dei Ricavi

Questa query è progettata per analizzare l'andamento dei ricavi nel tempo, fornendo una visione chiara dei trend, della stagionalità e di eventuali anomalie. È uno strumento indispensabile per la pianificazione strategica, la previsione delle vendite e la valutazione dell'efficacia delle iniziative commerciali. La seguente slide mostrerà l'implementazione SQL concreta per costruire una serie temporale completa dei ricavi mensili, segmentata per fascia d'età.

La capacità di visualizzare i ricavi come una serie temporale consente di identificare modelli di crescita a lungo termine, cicli di mercato e punti di svolta critici, permettendo decisioni proattive per il business.



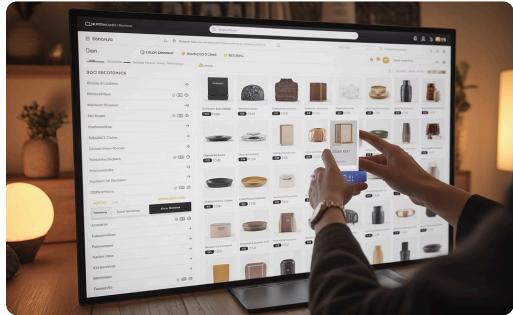


Query 7: Serie Temporale dei Ricavi

```
SELECT
    dd.anno,
    dd.mese,
    dc.fascia_eta,
    SUM(f.ricavi) AS ricavi_mensili
FROM fact_vendite f
JOIN dim_data dd
    ON f.data_sk = dd.data_sk
JOIN dim_cliente dc
    ON f.cliente_sk = dc.cliente_sk
GROUP BY
    dd.anno,
    dd.mese,
    dc.fascia_eta
ORDER BY
    dd.anno,
    dd.mese,
    dc.fascia_eta;
```

Questa query costruisce una serie temporale completa dei ricavi mensili, per fascia di età.

Riepilogo Query OLAP Essenziali



Analisi per Categoria

Vendite aggregate per categoria e periodo temporale



Clienti Top

Identificazione clienti ad alto valore economico



Performance Logistica

KPI operativi dei corrieri e tempi di consegna



Segmentazione

Analisi demografica per marketing targeting

Step 7: Esercizio Finale

Query Multidimensionale Avanzata

Costruisci una tabella analitica completa che integri molteplici dimensioni e metriche calcolate. L'obiettivo è ottenere: **Anno, Categoria, Ricavi Totali, Quantità Totale, Ticket Medio**

💪 Metti alla Prova le Tue Competenze

Questa query finale combina tutto ciò che hai imparato: join multipli, aggregazioni diverse (SUM), e calcoli derivati (ticket medio = ricavi / quantità). Rappresenta un caso reale di analisi multidimensionale complessa.

Soluzione Guidata — Esercizio Finale

```
SELECT
dd.anno,
dp.categoria,
SUM(f.ricavi) AS ricavi_totali,
SUM(f.quantita) AS quantita_totale,
SUM(f.ricavi) / SUM(f.quantita) AS ticket_medio
FROM fact_vendite f
JOIN dim_data dd
ON f.data_sk = dd.data_sk
JOIN dim_prodotto dp
ON f.prodotto_sk = dp.prodotto_sk
GROUP BY dd.anno, dp.categoria
ORDER BY dd.anno, ricavi_totali DESC;
```

Analisi della Soluzione

La soluzione guidata dell'esercizio finale dimostra come combinare più dimensioni e metriche per un'analisi approfondita. Esaminiamo gli elementi chiave che rendono questa query efficace per estrarre insight multidimensionali.

Integrazione Multidimensionale

Attraverso i JOIN con `dim_data` e `dim_prodotto`, abbiamo collegato i fatti di vendita a dimensioni temporali e di prodotto, consentendo l'analisi dei ricavi e delle quantità per anno e categoria.

Metriche Aggregate e Calcolate

Le funzioni di aggregazione `SUM()` sono state utilizzate per ottenere i `ricavi_totali` e la `quantita_totale`. Il `ticket_medio` è una metrica calcolata dinamicamente che offre un insight sul valore medio per singola transazione.

Raggruppamento e Ordinamento

Il `GROUP BY dd.anno, dp.categoria` è fondamentale per ottenere i totali per ogni combinazione di anno e categoria, mentre l'`ORDER BY` ci permette di visualizzare i dati in modo strutturato e gerarchico.

Potenza Analitica

Questa query è un esempio concreto di come le capacità OLAP permettano di navigare attraverso i dati e di derivare KPI cruciali per la performance del business, supportando decisioni strategiche su assortimento prodotti, pricing e strategie di vendita.

Analisi della Soluzione



Due Join Dimensionali

La query collega fact_vendite con entrambe le dimensioni data e prodotto per ottenere prospettive temporali e di categoria.



Aggregazioni Multiple

SUM() viene applicato a due metriche distinte (ricavi e quantità) per calcolare i totali su ciascuna dimensione.



Metrica Calcolata

Il ticket medio è una metrica derivata, ottenuta dividendo ricavi totali per quantità totale — rappresenta il valore medio per unità venduta.

Ordinamento Strategico

I risultati sono ordinati prima per anno (cronologico) e poi per ricavi (valore economico), facilitando l'analisi dei trend.

Esercizio Finale Avanzato

Sfida Multidimensionale Completa

Ora che hai familiarizzato con le query OLAP di base e intermedie, mettiti alla prova con un'analisi più complessa. L'obiettivo è costruire una query SQL che combini diverse dimensioni per ottenere insight strategici.

Il Tuo Obiettivo

- Calcola il **numero di ordini** per ogni regione e categoria di prodotto.
- Determina i **ricavi totali** per ciascuna categoria all'interno di quella regione.
- Calcola lo **scontrino medio** per ordine (Ricavi Totali / Numero di Ordini).
- Considera solo i dati relativi all'anno **2024**.
- Ordina i risultati per **ricavi totali in modo decrescente**.



Pattern Comuni nelle Query OLAP

JOIN

Collega fact e dimensioni tramite chiavi surrogate (_sk)

WHERE

Filtra i dati prima dell'aggregazione (opzionale)

GROUP BY

Definisce le dimensioni di analisi per l'aggregazione

AGGREGATE

Applica funzioni (SUM, AVG, COUNT) alle metriche

ORDER BY

Ordina i risultati per migliorare la leggibilità

Questo pattern ricorrente rappresenta la struttura base di ogni query OLAP: seleziona le dimensioni rilevanti, aggrega le metriche, e presenta i risultati in modo ordinato e significativo.

Competenze Acquisite

Tecniche SQL Avanzate

- JOIN tra tabelle multiple
- Funzioni di aggregazione (SUM, AVG, COUNT)
- Raggruppamenti multidimensionali
- Calcolo di metriche derivate
- Ordinamento e limitazione risultati

Concetti OLAP Fondamentali

- Architettura star schema
- Fact table vs dimension table
- Chiavi surrogate e relazioni
- Analisi multidimensionale
- KPI e metriche di business



Prossimi Passi

01

Pratica Autonoma

Esercitati modificando le query presentate: cambia le dimensioni di analisi, aggiungi filtri WHERE, sperimenta con altre funzioni di aggregazione.

02

Query Avanzate

Approfondisci con window functions, subquery, CTE (Common Table Expressions) e operazioni di drill-down/roll-up tipiche dell'OLAP.

03

Strumenti BI

Applica queste competenze utilizzando tool di Business Intelligence come Power BI, Tableau o Looker per creare dashboard interattive.

04

Progetti Reali

Cerca opportunità di stage o progetti pratici dove implementare analisi OLAP su dati reali, consolidando le competenze acquisite.