



**UNIMORE**  
UNIVERSITÀ DEGLI STUDI DI  
MODENA E REGGIO EMILIA

**DIPARTIMENTO DI INGEGNERIA “ENZO FERRARI”**  
**CORSO DI LAUREA IN INGEGNERIA INFORMATICA**

# Ottimizzazione delle Query per Sistemi di Business Intelligence: Tecniche Avanzate per Migliorare le Performance e il Supporto Decisionale

# Architettura Business Intelligence



Esplorazione e aggregazione dati analitici con qualità, sicurezza e disponibilità

# OLTP vs OLAP

## OLTP

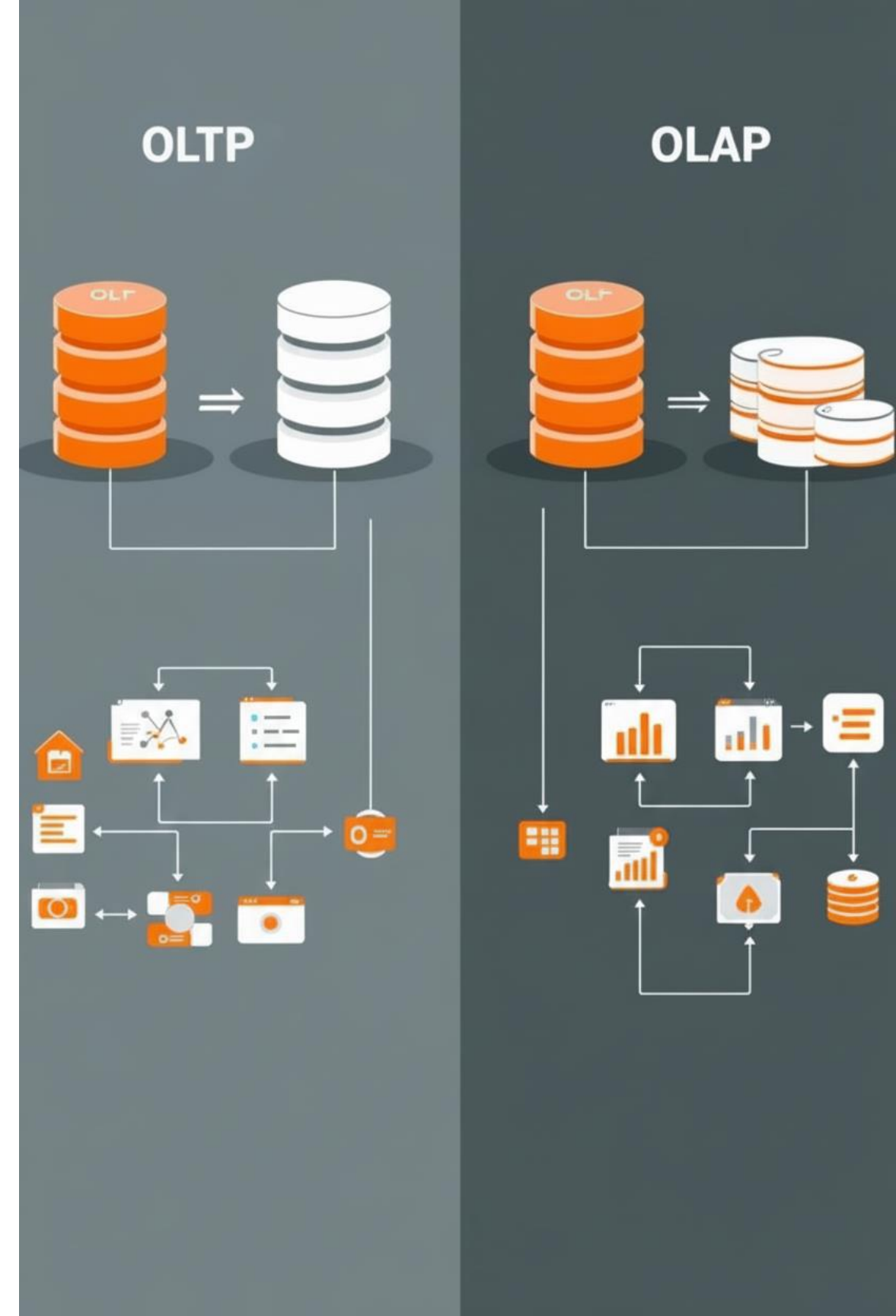
### Online Transaction Processing

- Operazioni veloci, bassa latenza
- **Schemi normalizzati**
- Write-intensive
- **Orizzonte 60-90 giorni**
- Aggiornamenti real-time

## OLAP

### Online Analytical Processing

- Analisi complesse, grandi volumi
- **Schemi denormalizzati**
- Read-intensive
- **Orizzonte pluri-annuale**
- Elaborazione batch



# Processi ETL: Criticità e Ottimizzazione



## Estrazione

Fase più critica – fonti eterogenee



## Trasformazione

Nodi critici: join, script, lookup



## Caricamento

Monitoraggio automatizzato  
necessario

## Compromesso Qualità-Performance

Accuratezza totale richiede +45% tempo elaborazione

## Scalabilità

Partizionamento parallelo e pipeline distribuite

# Tecniche di Ottimizzazione Fisica

## Caching

Caching Dinamico

Conserva **risultati frequenti**

**Riduce** drasticamente I/O

## Indicizzazione

**Bitmap** e non RowID

Rappresentazione Compatta

Ideale per bassa cardinalità OLAP

## Compressione

30-35% di riduzione memoria

**Huffman**: bilanciato velocità/compressione

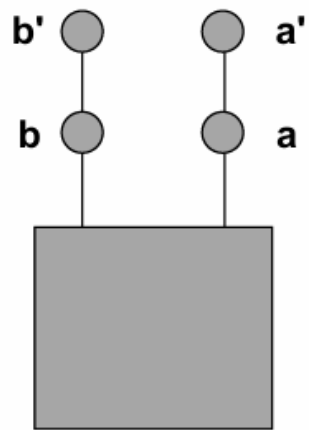
**Arithmetic**: ottima compressione, non veloce

## Partizionamento

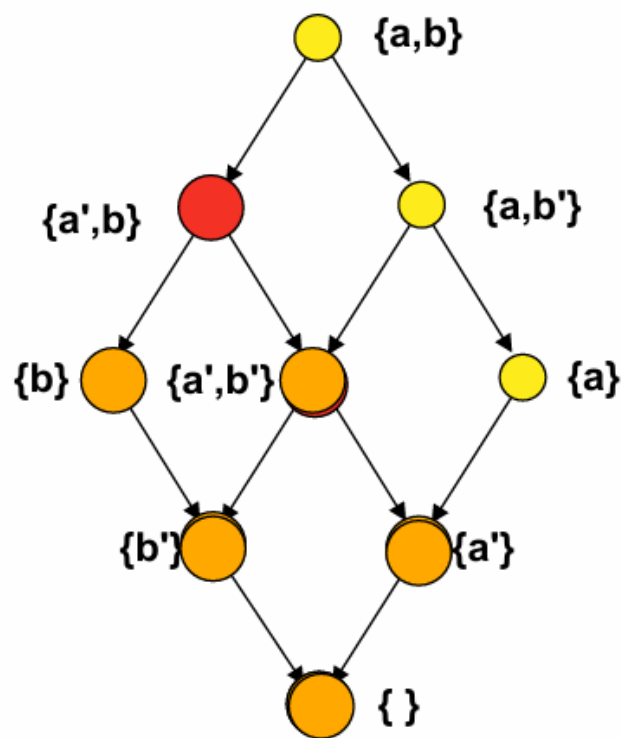
**Orizzontale**: periodi temporali

**Verticale**: attributi specifici

Gestione grandi tabelle velocizzata



**a:** prodotto  
**a':** tipo di prodotto  
**b:** città  
**b':** regione



**Reticolo di roll-up**

# Viste Materializzate

## Uno strumento efficace



### Aggregazione

Diminuiscono righe, aumentano scalabilità



### Drill-Down/Roll-Up

Navigazione livelli dettaglio



### Riutilizzo

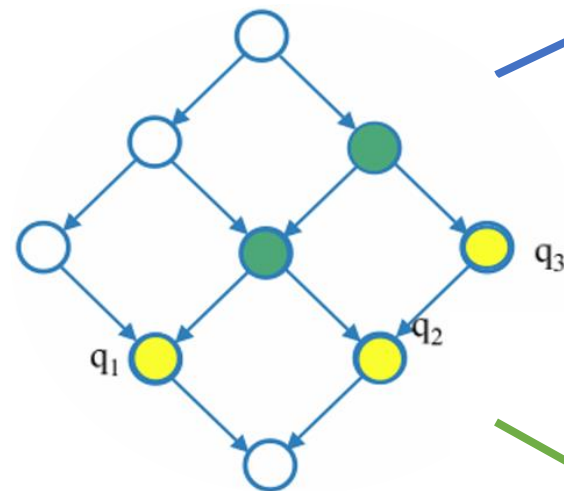
Risultati memorizzati per query successive



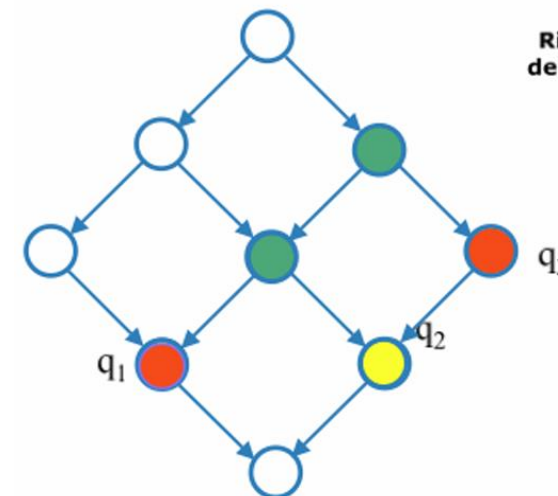
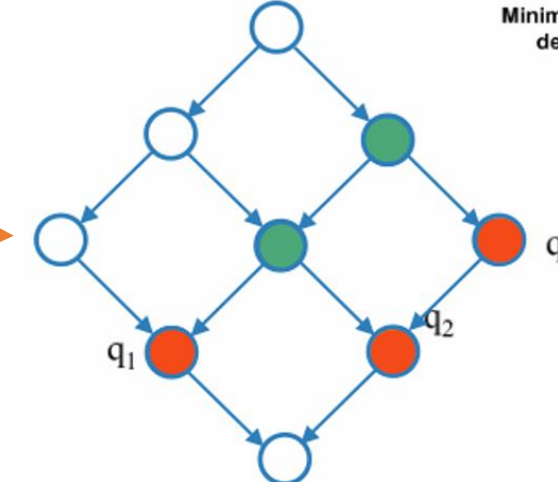
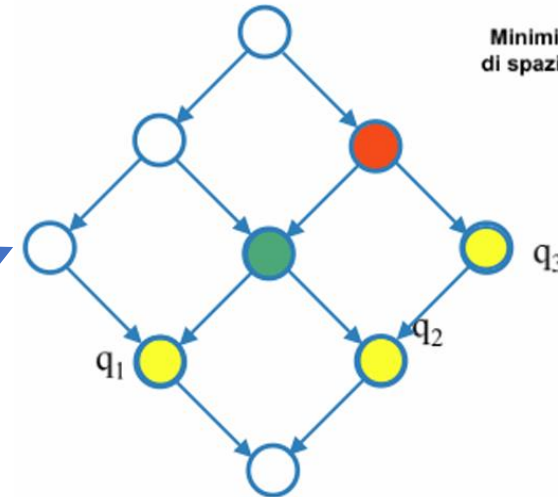
**Problema NP-completo:** Selezione viste richiede algoritmi euristici, genetici o ibridi per gestire crescita esponenziale

# Viste Materializzate

Influenza dei parametri di ottimizzazione sulla qualità della soluzione



● + ●  
Viste candidate



# Analysis Services OLAP

## Esempio pratico:

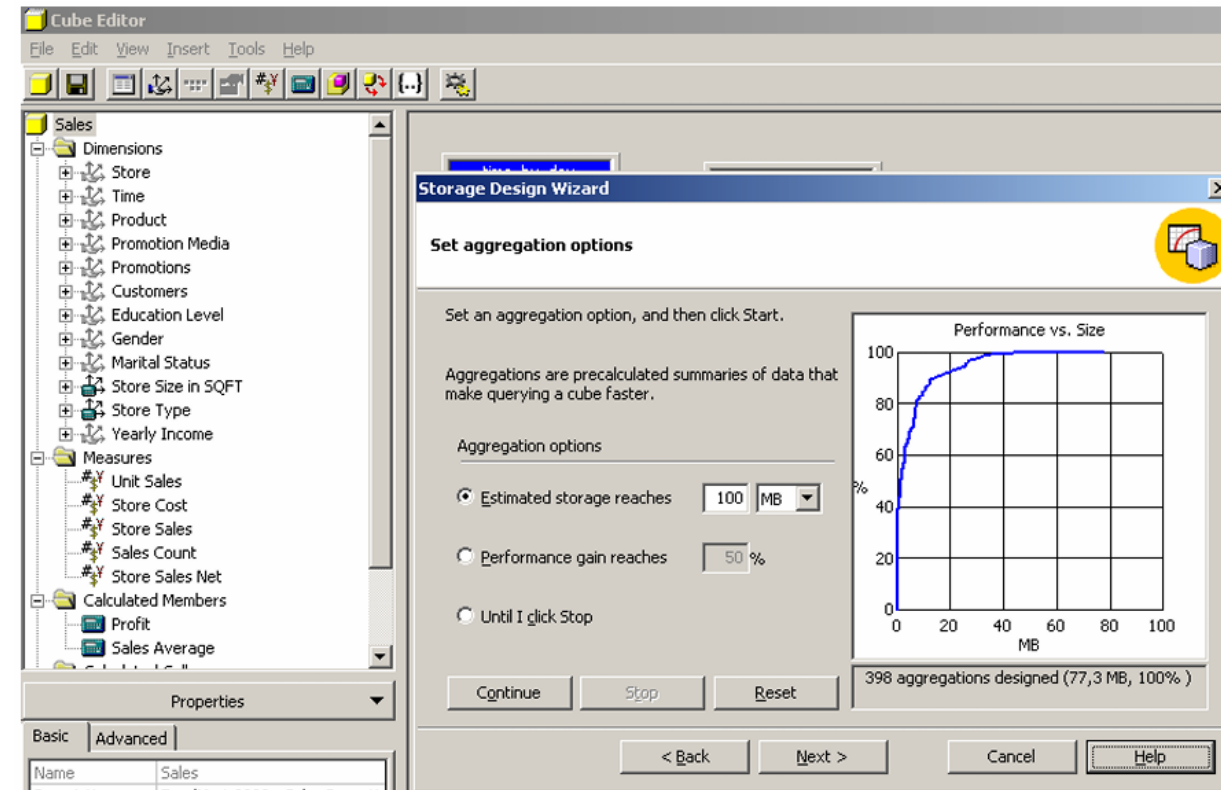
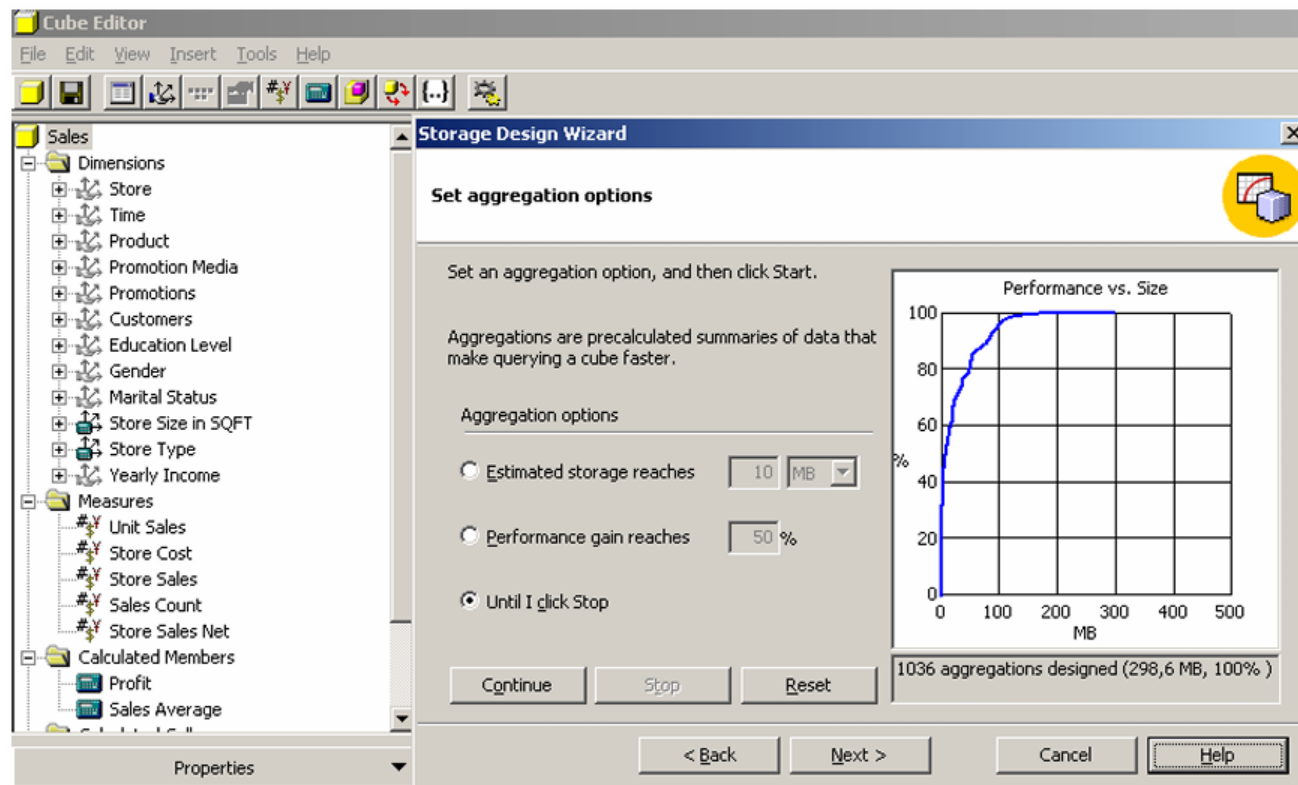


**Durante l'analisi è possibile precalcolare tutte le possibili aggregazioni**

In questo cubo ci sono 1036 aggregazioni

**Oppure si fissa un limite allo spazio di archiviazione stimato**

Fissando 100MB si ottengono 398 aggregazioni





# Metodologia Operativa in 4 Fasi

01

## Profilare

EXPLAIN per individuare operatori dominanti e movimenti dati

02

## Progettare

Viste materializzate su join/aggregazioni con distribuzione coerente

03

## Validare

Verificare auto-matching query verso viste

04

## Monitorare

DBCC overhead e REBUILD programmati nelle finestre ETL

Valutazione: viste correttamente progettate riducono stabilmente latenza nel data warehouse

```
1  -- 1) PROFILARE: piano stimato e raccomandazioni
2  EXPLAIN WITH_RECOMMENDATIONS
3  SELECT
4      fs.StoreID, fs.MonthKey,
5      SUM(fs.SalesAmount) AS Amt
6  FROM dbo.FactSales AS fs
7  JOIN dbo.DimStore AS ds ON ds.StoreID = fs.StoreID
8  GROUP BY fs.StoreID, fs.MonthKey;
9
10 -- 2) PROGETTARE: MV che cattura join+aggregazione condivise, co-locata su
    StoreID
11 CREATE MATERIALIZED VIEW dbo.mv_SalesByStoreMonth
12 WITH (DISTRIBUTION = HASH(StoreID))
13 AS
14 SELECT
15     fs.StoreID, fs.MonthKey,
16     SUM(fs.SalesAmount) AS Amt
17 FROM dbo.FactSales AS fs
18 GROUP BY fs.StoreID, fs.MonthKey;
19
20 -- 3) VALIDARE: la query non cita la MV; ottimizzatore riscrive in
    automatico
21 EXPLAIN
22 SELECT
23     fs.StoreID, fs.MonthKey,
24     SUM(fs.SalesAmount) AS Amt
25 FROM dbo.FactSales AS fs
26 GROUP BY fs.StoreID, fs.MonthKey;
27 -- Atteso: piano che legge da dbo.mv_SalesByStoreMonth (auto-matching).
28
29 -- 4) MONITORARE: overhead incrementale e manutenzione programmata
30 DBCC PDW_SHOWMATERIALIZEDVIEWOVERHEAD('dbo.mv_SalesByStoreMonth');
31 -- Se overhead_ratio supera la soglia operativa:
32 ALTER MATERIALIZED VIEW dbo.mv_SalesByStoreMonth REBUILD;
```

The image features a central 3D database cylinder with three orange rings. The background is a dark blue gradient with faint, stylized icons of a laptop, a target, and a bar chart. The text "THANKS FOR YOUR ATTENTION" is written in large, white, bold, sans-serif capital letters across the center of the cylinder.

**THANKS  
FOR YOUR  
ATTENTION**