



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

# Architettura Business Intelligence



# OLTP vs OLAP

## OLTP

Online Transaction Processing

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

## OLAP

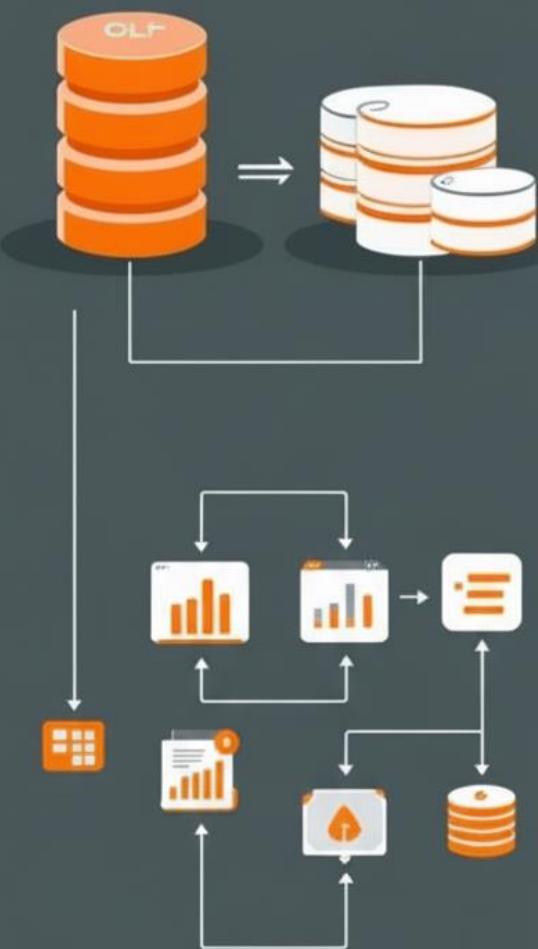
Online Analytical Processing

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

## OLTP



## OLAP



# 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

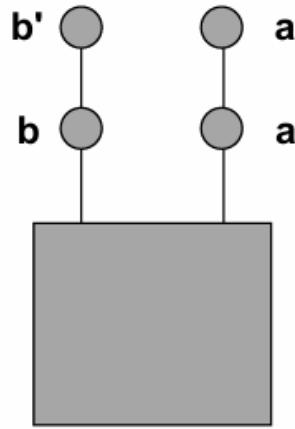
**Arithmetic**: ottima compressione, non veloce

## Partizionamento

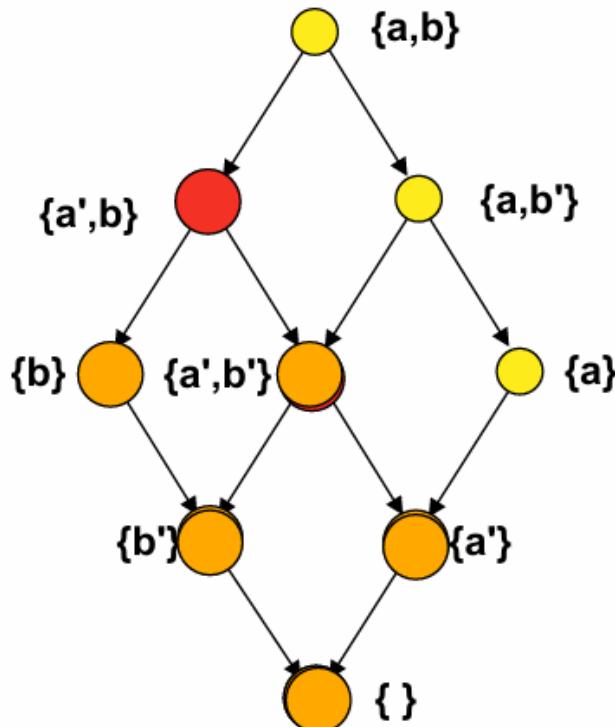
**Orizzontale**: periodi temporali

**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

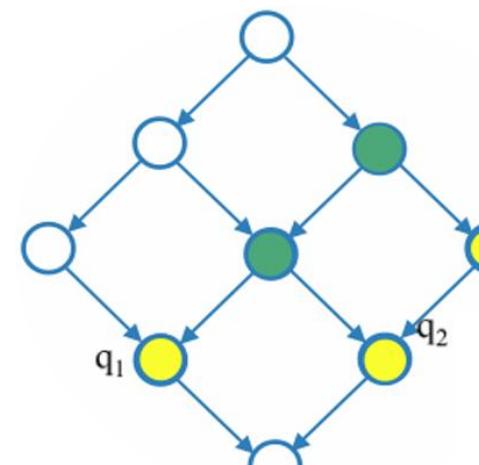
Risultati memorizzati per query successive



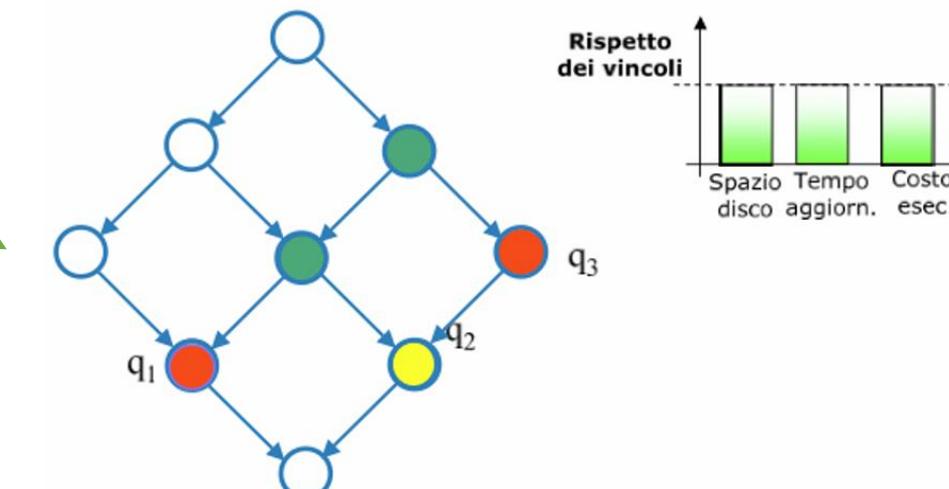
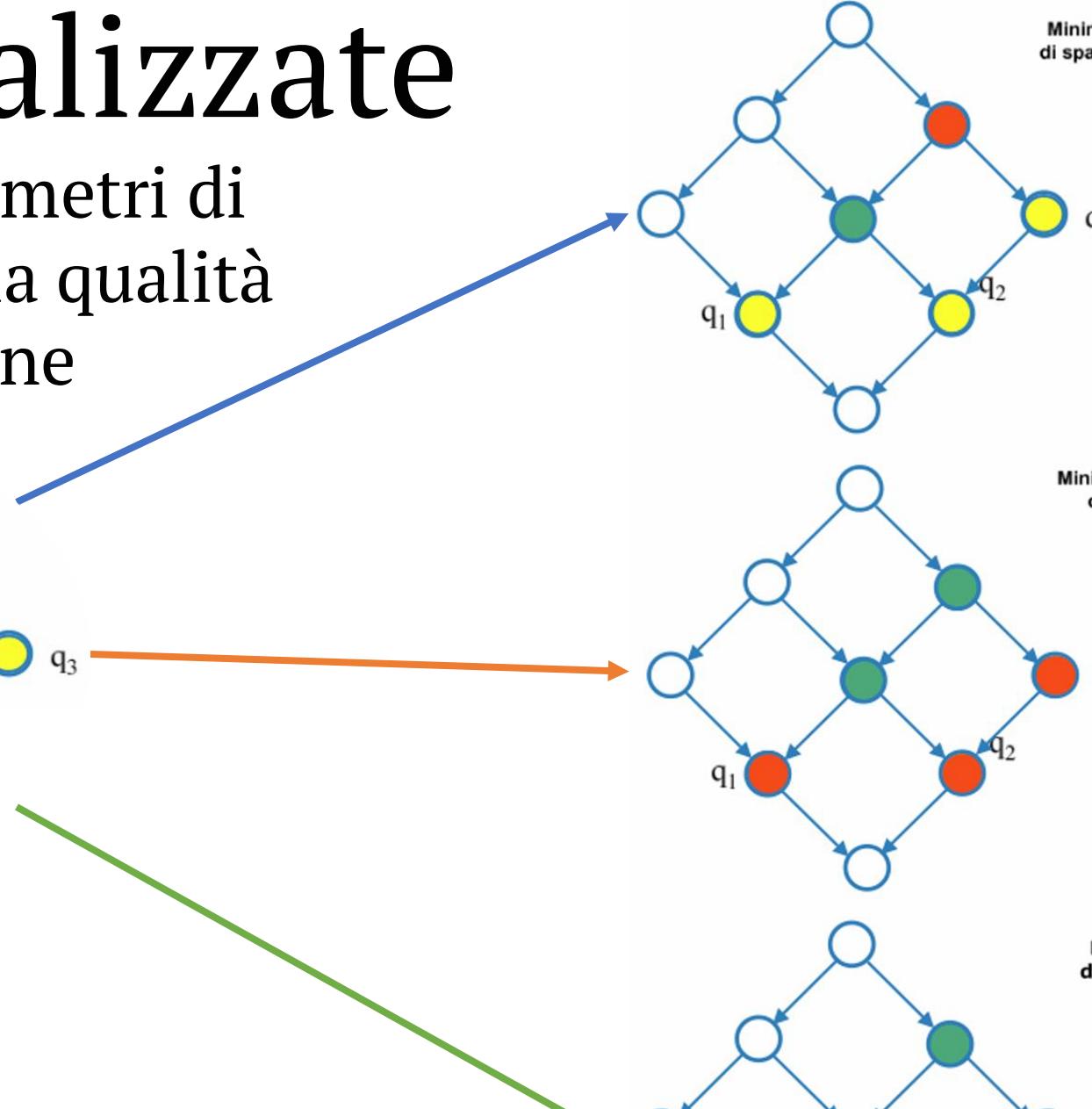
**Problema NP-completo:** Selezione viste richiede algoritmi euristici, genetici o ibridi per gestire crescita esponenziale

# Viste Materializzate

Influenza dei parametri di ottimizzazione sulla qualità della soluzione



● + ●  
Viste candidate



# Analysis Services OLAP

## Esempio pratico:

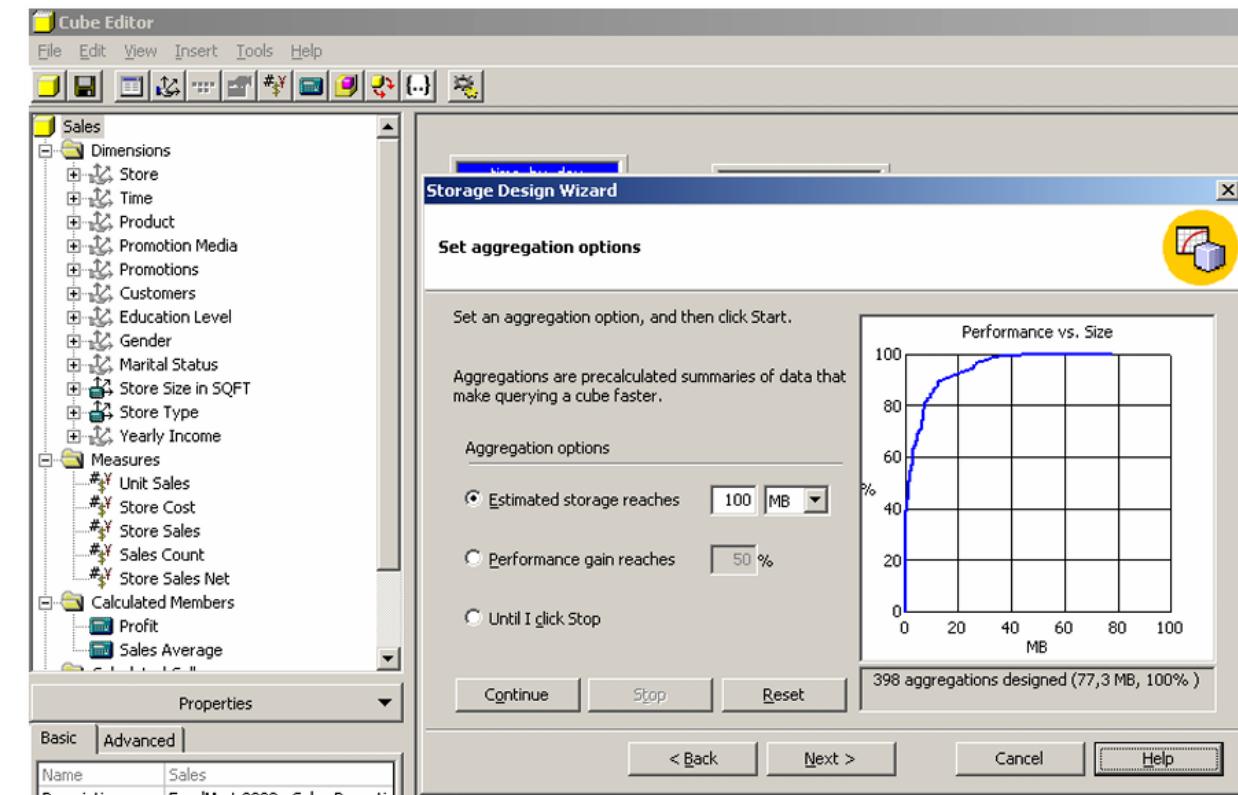
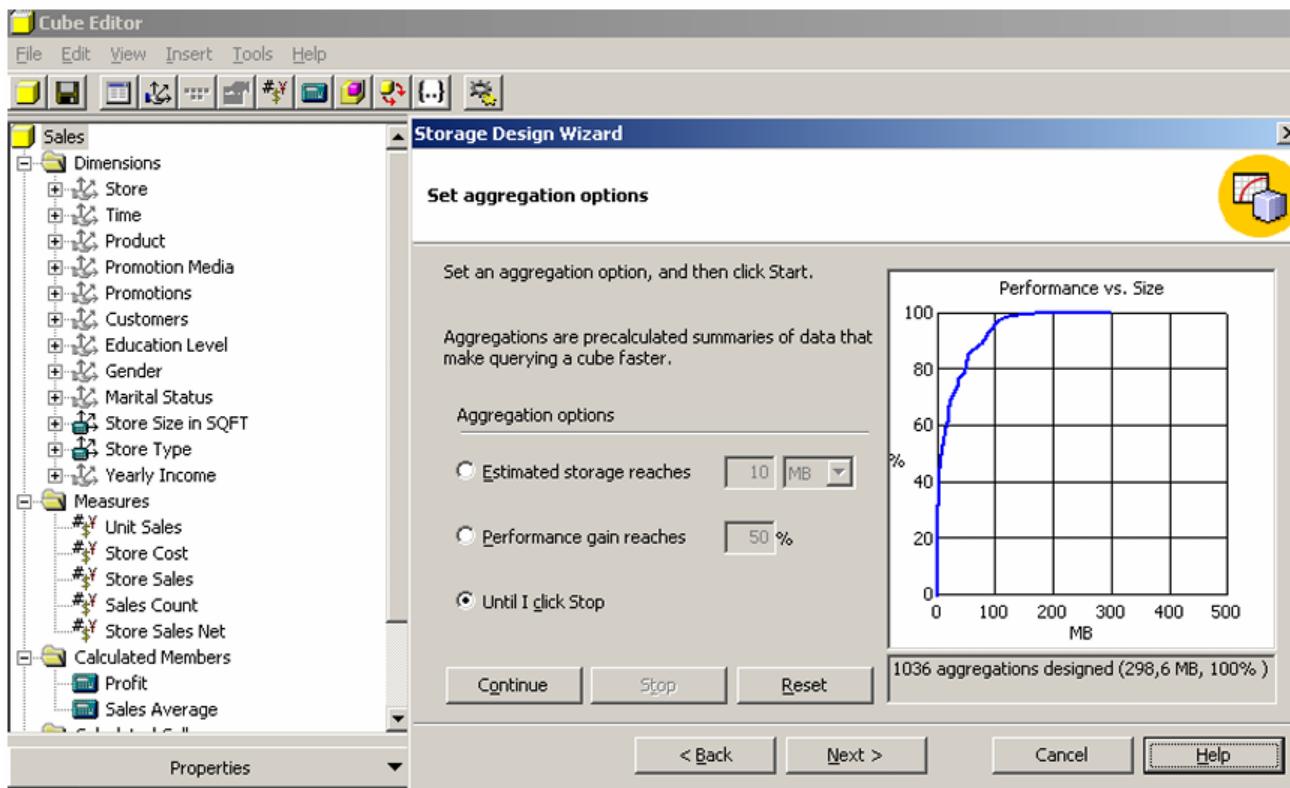


Durante l'analisi è possibile precalcolare tutte le possibili aggregazioni

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
11     StoreID
12 CREATE MATERIALIZED VIEW dbo.mv_SalesByStoreMonth
13 WITH (DISTRIBUTION = HASH(StoreID))
14 AS
15     SELECT
16         fs.StoreID, fs.MonthKey,
17         SUM(fs.SalesAmount) AS Amt
18     FROM dbo.FactSales AS fs
19     GROUP BY fs.StoreID, fs.MonthKey;
20
21 -- 3) VALIDARE: la query non cita la MV; ottimizzatore riscrive in
22     automatico
23 EXPLAIN
24 SELECT
25     fs.StoreID, fs.MonthKey,
26     SUM(fs.SalesAmount) AS Amt
27 FROM dbo.FactSales AS fs
28 GROUP BY fs.StoreID, fs.MonthKey;
29 -- Atteso: piano che legge da dbo.mv_SalesByStoreMonth (auto-matching).
30
31 -- 4) MONITORARE: overhead incrementale e manutenzione programmata
32 DBCC PDW_SHOWMATERIALIZEDVIEWOVERHEAD('dbo.mv_SalesByStoreMonth');
33 -- Se overhead_ratio supera la soglia operativa:
34 ALTER MATERIALIZED VIEW dbo.mv_SalesByStoreMonth REBUILD;
```

**THANKS  
FOR YOUR  
ATTENTION**