

# Introduzione ad OLAP (On-Line Analytical Processing)

**Metodi e Modelli per il Supporto alle Decisioni  
2002**

**Dipartimento di Informatica Sistemistica e Telematica (Dist)**



- Il termine OLAP e' l'acronimo di On-Line Analytical Processing
- E' il nome con cui sono comunemente identificati strumenti e tecniche di analisi automatica di grosse quantita' di dati
- OLAP e' divenuto sinonimo di *vista multidimensionale di business data*
- OLAP costituisce uno strumento di supporto a decisioni di tipo manageriale



- La definizione di OLAP può essere riassunta in cinque parole chiave
  - Fast
  - Analysis
  - Shared
  - Multidimensional
  - Information
- Le cinque parole chiave sono usate per esprimere un concetto
  - Fast analysis of shared multidimensional information

Ovvero analisi veloce di informazione multidimensionale condivisa



- FAST
  - Una applicazione OLAP deve risultare molto veloce
  - Ricerche hanno evidenziato che un utente finale considera un processo fallito se non viene mostrato un risultato entro 30 secondi
  - In ogni caso, non sempre è possibile ottenere una elevata velocità di elaborazione in presenza di grosse quantità di dati
  - Campo considerato tutt'ora in pieno sviluppo



- ANALYSIS

- Significa che il sistema può far fronte ad ogni logica di business e di analisi statistica importanti per l'applicazione e per gli obiettivi dell'utente
- È certamente necessario permettere all'utente di definire nuovi calcoli *ad hoc* come parte dell'analisi e riportare i risultati in qualunque modo desiderato

- INFORMATION

- Sono tutti i dati e l'informazione necessari, ovunque si trovino e qualunque sia la rilevanza per l'applicazione



- SHARED

- Significa che il sistema implementa tutti i requisiti di sicurezza
- Nonostante la sua importanza è una delle principali debolezze dei prodotti OLAP i quali tendono ad assumere che tutte le applicazioni saranno read-only, con semplici controlli di sicurezza
- Anche prodotti con accessi multi-utente in scrittura e lettura hanno modelli di sicurezza 'rozzi'



- **MULTIDIMENSIONAL**

- È il requisito chiave
- Se si dovesse scegliere una parola per definire OLAP, questa sarebbe *multidimensionale*
- Il sistema deve fornire una vista concettuale multidimensionale dei dati, includendo pieno supporto per gerarchie e gerarchie multiple
- L'approccio multidimensionale è certamente il modo più logico di analizzare *business e organizations*
- Non è imposto nessun vincolo sul numero minimo di dimensioni, come non è imposto nessun vincolo sulla tecnologia del database



- **DATI MULTIDIMENSIONALI**


- I database relazionali sono organizzati secondo una lista di records
- Ogni record contiene informazione correlata che e' organizzata in campi
- Un tipico esempio potrebbe essere una lista di clienti con i seguenti campi: address, telephone number,...

Customer Name	Customer #	Telephone	Address
Jack's Hardware	10456	350-7229	40 Main St.
Value Stores	10114	266-7023	18 Elm St.
Housewares Inc.	11104	267-4040	17 Main St.
Walter Lock	11230	423-7700	6 Charles St.



- **DATI MULTIDIMENSIONALI**


- La tabella precedente ha diverse colonne di informazione, ogni informazione e' relazionata ad un solo customer name
- La tabella ha una sola dimensione
- Se si volesse creare una matrice bi-dimensionale con customer name in verticale ed altri campi (come telephone number) in orizzontale si otterrebbe:



Customer Dimension ↓	→	Telephone Number	Dimension →
Jack's Hardware		350-7229	
Value Stores			266-7023
Housewares Inc.			267-4040
Walter Lock			423-7700

- **DATI MULTIDIMENSIONALI**


- Consideriamo un altro esempio: una tabella relazionale dove sono presenti piu' corrispondenze tra i campi
- Abbiamo i dati relativi alle vendite per ogni prodotto e per ogni regione



Product	Region	Sales
Nuts	East	50
Nuts	West	60
Nuts	Central	100
Screws	East	40
Screws	West	70
Screws	Central	80
Bolts	East	90
Bolts	West	120
Bolts	Central	140
Washers	East	20
Washers	West	10
Washers	Central	30

- DATI MULTIDIMENSIONALI

- Una migliore rappresentazione delle informazioni della tabella precedente e' attraverso una matrice bi-dimensionale
- Le dimensioni sono *product* e *region*
- I valori *sales* sono rappresentati attraverso le altre due dimensioni



SALES ←

	East	West	Central
Nuts	50	60	100
Screws	40	70	80
Bolts	90	120	140
Washers	20	10	30

- DATI MULTIDIMENSIONALI

- Supponiamo di voler estrarre le seguenti informazioni:  
*quante viti sono state vendute nell'est?, quante rondelle sono state vendute nell'ovest?*
- Per estrarre un singolo dato potrebbe non esserci bisogno di tabelle multidimensionali
- Se si volesse, invece, rispondere alla domanda: *Quale e' il totale di vendite per l'est? oppure Quale e' il totale di dadi venduti?* L'operazione richiederebbe di estrarre informazioni multiple aggregate fra loro



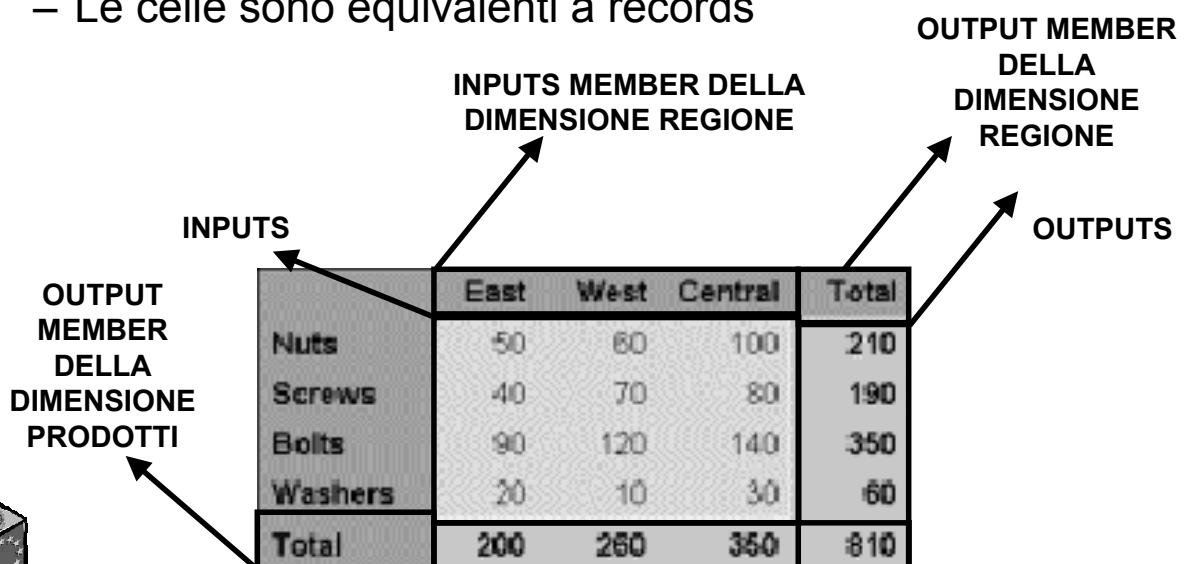
- **DATI MULTIDIMENSIONALI**

- Se si considera un database di grandi dimensioni con migliaia di prodotti il tempo per estrarre l'informazione diverrebbe incredibilmente alto
- Un tipico database relazionale puo' analizzare poche centinaia di records al secondo
- Un tipico database multidimensionale puo' analizzare un insieme di 10000 righe/colonne al secondo
- Per rispondere alla domanda "*Quale e' il totale di vendite per l'est*", un database bi-dimensionale cerca semplicemente la colonna "EAST" ed esegue la somma



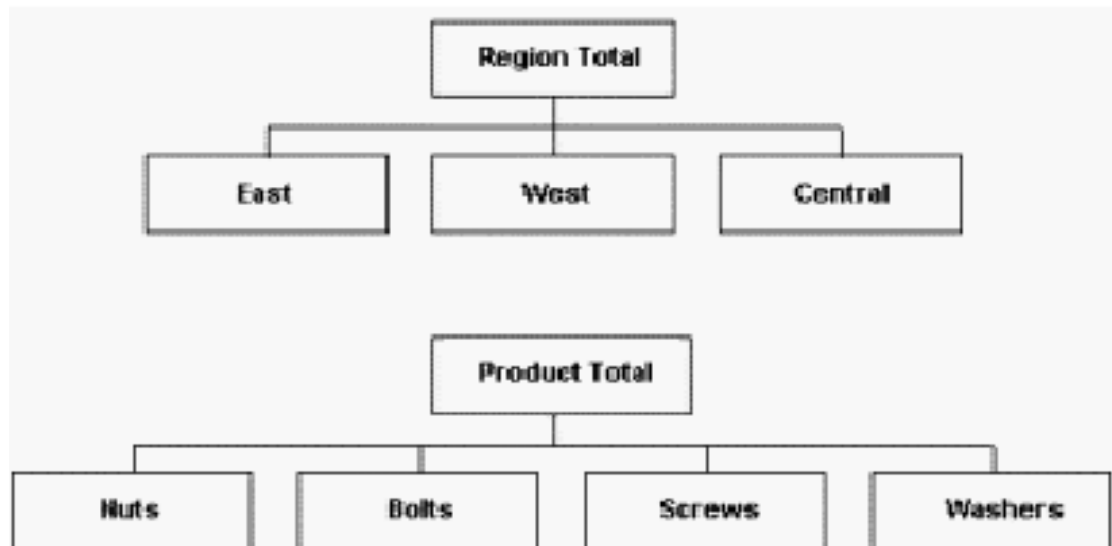
- **TERMINOLOGIA**

- Le dimensioni sono equivalenti ai campi in un database relazionale.
- Le celle sono equivalenti a records



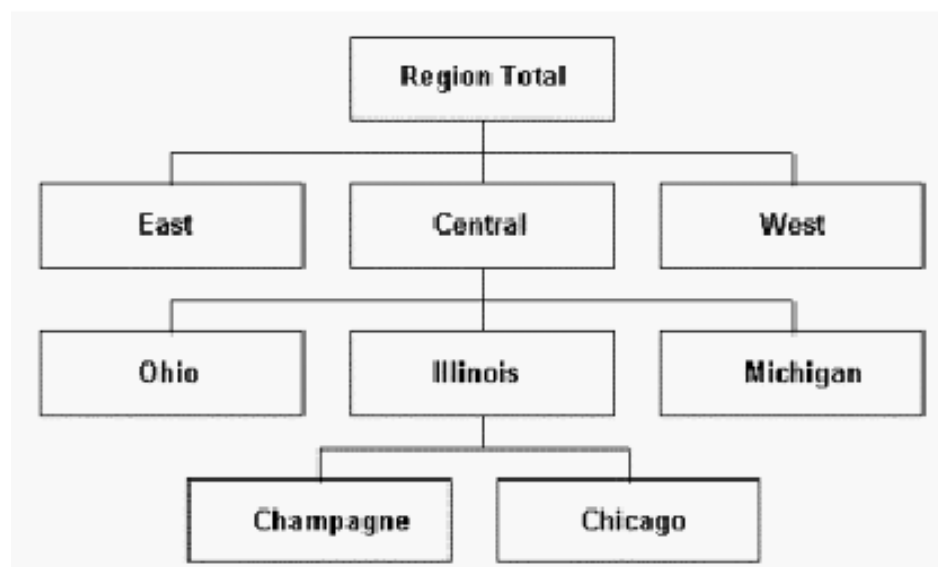
- GERARCHIE

- Nell'esempio precedente era presente una semplice gerarchia per la dimensione Region



- GERARCHIE


- E' possibile avere anche gerarchie a piu' livelli all'interno di una stessa dimensione:





## • GERARCHIE

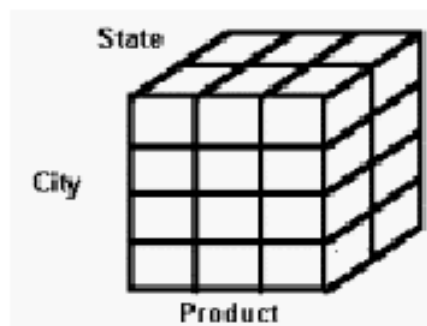
- La ragione per cui abbiamo bisogno di gerarchie multilivelli all'interno di una dimensione, invece di dimensioni addizionali, e' che non si possono considerare come oggetti omogenei entità quali, ad esempio, città, stati e regioni
- Consideriamo la seguente tabella e supponiamo che un utente voglia vedere i prodotti venduti per città o per regione
- Unire città e regioni nella stessa dimensione significa avere una colonna con i totali errati perché i valori della città sono già inclusi in quelli delle regioni



East				
West				
Central				
New York				
New Jersey				
etc.				
	Product			

## • GERARCHIE

- In un database multidimensionale senza gerarchie, la soluzione a questo problema potrebbe essere quella di separare, per esempio, città e stati in diverse dimensioni
- E' comunque concettualmente più complicato
- All'aumentare dei livelli aumentano anche le dimensioni del cubo



- GERARCHIE

- Il modo corretto di risolvere il problema e' appunto quello di usare le gerarchie all'interno delle dimensioni
- States nell'East Region rappresenta un livello inferiore, cities un sotto-livello di states e cosi' via



- IL CONCETTO DI CUBO

- Il principale oggetto di una applicazione OLAP è il CUBO
- Un cubo è una rappresentazione multidimensionale dei dati
- Un cubo consiste di:
  - Una sorgente dei dati
  - Dimensioni
  - Misure (measures)
- La progettazione di un cubo è basata sui requisiti analitici dell'utente. Un'applicazione OLAP può supportare differenti cubi come SALES CUBE, INVENTORY CUBE, ...



- IL CONCETTO DI CUBO

- La sorgente dei dati identifica e connette un cubo ad un database dove è presente l'informazione
- Le dimensioni mappano le informazioni presenti nelle *dimension table* in una gerarchia di livelli, come ad esempio la dimensione GEOGRAFIA con i livelli di CONTINENTE, STATO, CITTÀ
- Le misure (measures) identificano i valori numerici provenienti dalla *fact table* che sono riassunti per l'analisi, es. prezzo, costo, quantità...



- DEFINIZIONE DI DIMENSIONE

- Attributo strutturale di un cubo, la dimensione e' un'organizzazione gerarchica di livelli (categorie) che *descrivono* i dati in una *fact table*
- Le categorie descrivono un insieme di elementi simili fra loro, sul quale l'utente basa la propria analisi



- DEFINIZIONE DI *MEASURES*

- Sono un insieme di valori di un cubo che sono basati su una colonna della *fact table* e sono solitamente numerici
- Le *measures* sono i valori centrali che sono aggregati ed analizzati



- FACT TABLE

- Contengono i dati che descrivono uno specifico evento come una transazione bancaria oppure una vendita
- Alternativamente le fact table possono contenere dati aggregati, ad esempio per mese, per regione, ecc.
- Poiché le fact tables contengono la maggioranza dei dati memorizzati nel data warehouse, è importante che la struttura della tabella sia corretta prima che la tabella sia letta



- **FACT TABLE**

- Le caratteristiche delle fact tables sono:
  - Molte righe (anche milioni e piu')
  - I dati primari sono numerici (raramente caratteri)
  - Foreign keys multiple verso dimension tables
  - Dati statici



- **DIMENSION TABLE**

- Contengono dati usati come riferimento ai dati memorizzati nella fact table come descrizione di prodotti, nomi di clienti, indirizzi, fornitori, ecc.
- Le dimension tables non contengono tante righe come le fact tables
- I dati sono soggetti a cambiamenti
- Le dimension tables sono strutturate per permettere cambiamenti



- DIMENSION TABLE

- Le caratteristiche delle dimension tables sono:
  - Meno righe delle fact tables (centinaia di migliaia)
  - I dati primari sono caratteri
  - Colonne multiple sono usate per gestire le gerarchie delle dimensioni
  - Una chiave primaria (dimensional key)

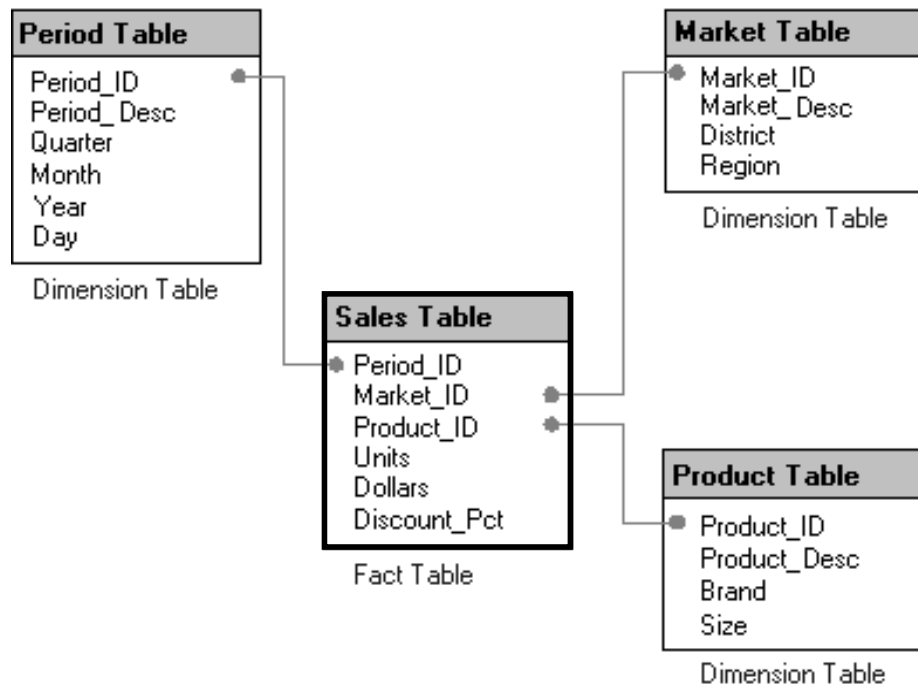


- STAR SCHEMA

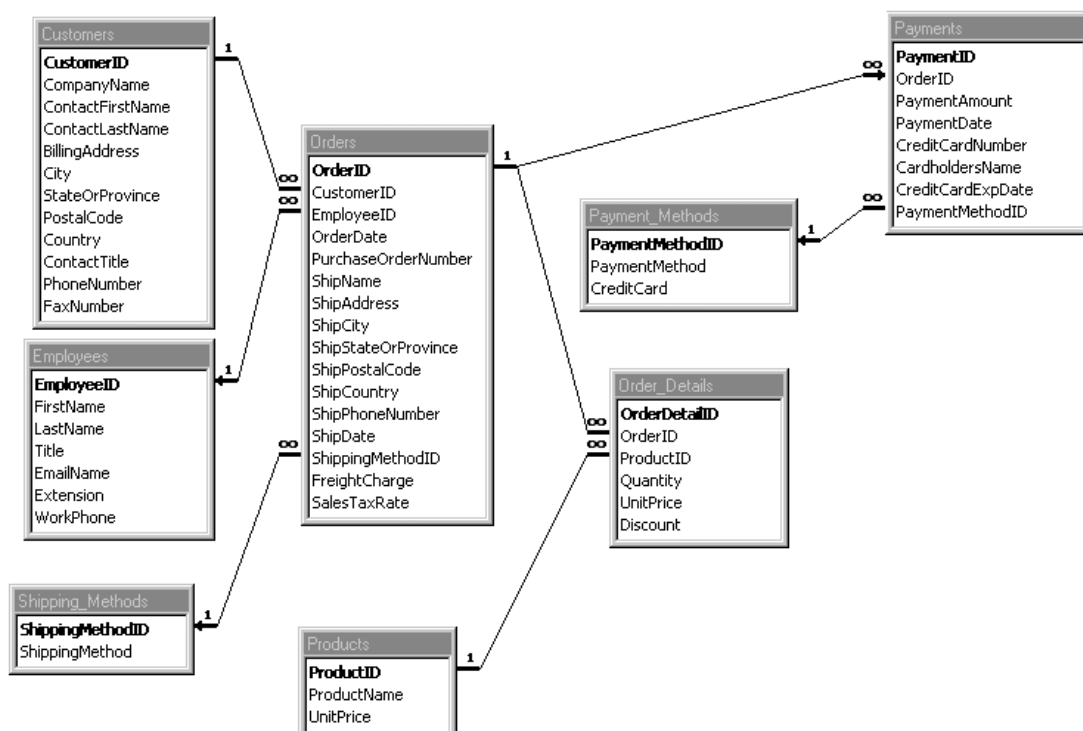
- La piu' popolare tecnica usata per implementare un data warehouse e' lo *star schema*
- La struttura comprende una fact table centrale per una particolare area, e molte dimension tables contenenti descrizione dei fatti



- STAR SCHEMA



- ESEMPIO DI CREAZIONE DI UNO STAR SCHEMA



- ESEMPIO DI CREAZIONE DI UNO STAR SCHEMA

- I passi per creare uno star schema, partendo da uno schema relazionale, sono:

- Determinare le fact e dimension tables
    - Progettare le fact tables
    - Progettare le dimension tables



- DETERMINARE LE FACT E DIMENSION TABLES

- I principali passi da seguire sono identificare:

- Transazioni fondamentali che il data warehouse focalizzerà' (fact tables)
    - I dati associati con le transazioni che determinano come i dati saranno analizzati





- DETERMINARE LE FACT E DIMENSION TABLES

- Identificare le transazioni fondamentali

- Riferendosi all'esempio precedente, l'informazione necessaria a descrivere la vendita di un prodotto (transazione principale) e' rappresentata dalla tabella ORDER\_DETAILS

Order_Details	
OrderDetailID	
OrderID	
ProductID	
Quantity	
UnitPrice	
Discount	



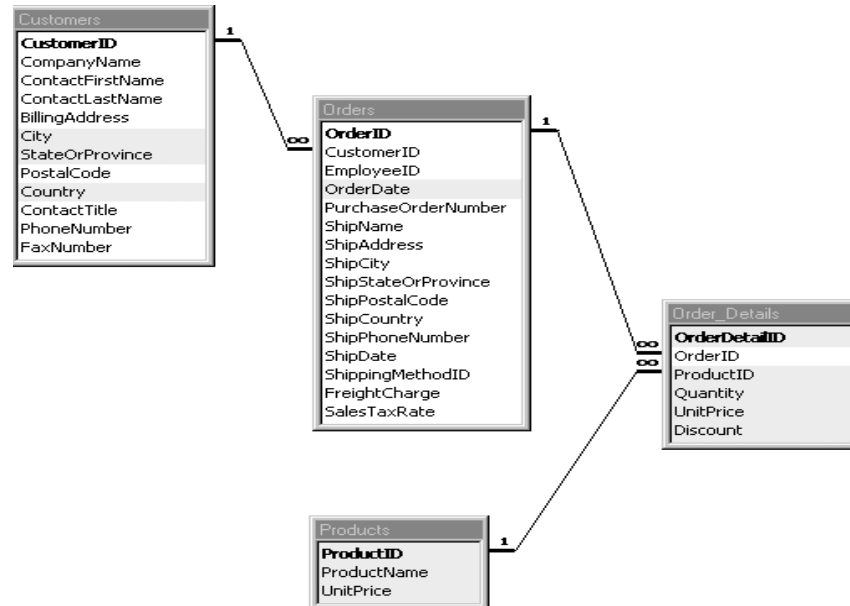
- ESEMPIO DI CREAZIONE DI UNO STAR SCHEMA

- Identificare le dimension tables

- Il passo successivo comprende l'identificazione delle entita' che descrivono come i dati saranno analizzati
    - Ad esempio, se la transazione principale e' la vendita di un prodotto, le dimensioni da scegliere potrebbero essere i metodi di pagamento, le date di vendita, o i modi di consegna
    - In ogni caso, le dimensioni scelte devono rappresentare la parte centrale dell'analisi di business



- ESEMPIO DI CREAZIONE DI UNO STAR SCHEMA
  - Identificare le dimension tables
    - Riferendosi all'esempio precedente, le dimension tables potrebbero essere



- ESEMPIO DI CREAZIONE DI UNO STAR SCHEMA
  - Design di una dimension table
    - Il design di una fact table comprende la riduzione della dimensione della tabella ottenuta:
      - Riducendo il numero di colonne
      - Riducendo la dimensione di ogni colonna, quando e' possibile
      - Archiviando dati storici i tabelle separate



- ESEMPIO DI CREAZIONE DI UNO STAR SCHEMA

- Design di una dimension table

- L'obiettivo primario per progettare le dimension tables e' *denormalizzare* i dati riferiti a fact tables in singole tabelle

- Esempio:

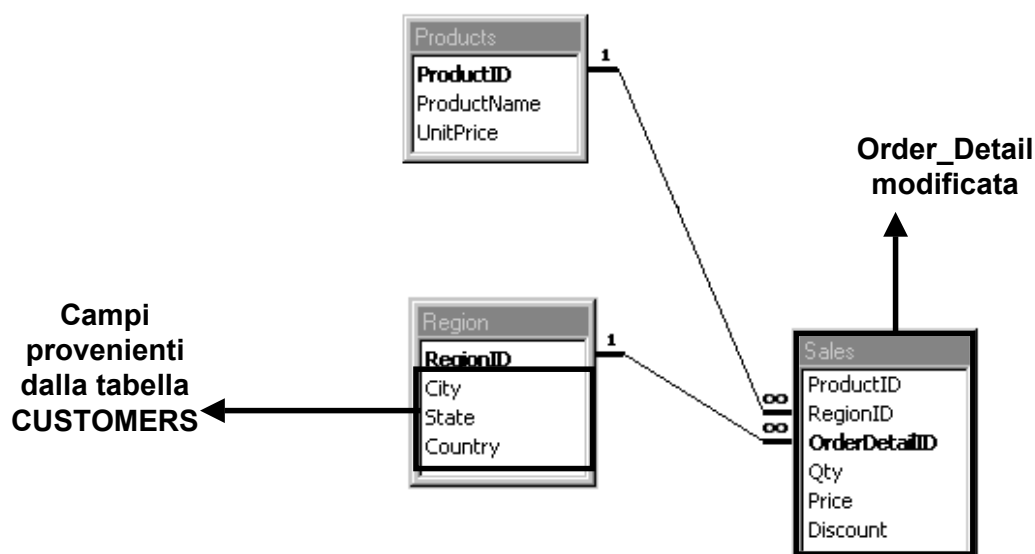
- Supponiamo che si voglia supportare le seguenti queries

- Vendite di un specifico prodotto per regione

- Tutte le vendite per regione



- ESEMPIO DI CREAZIONE DI UNO STAR SCHEMA



- ESEMPIO

