# Il ciclo di sviluppo del Data Warehouse

Sistemi Informativi L

Corso di Laurea in Ingegneria dei Processi Gestionali
A.A. 2003/2004

Docente: Prof. Wilma Penzo

## Perché?

- Molte organizzazioni mancano della necessaria esperienza e capacità per affrontare con successo le sfide implicite nei progetti di data warehousing
- Uno dei fattori che maggiormente minaccia la riuscita dei progetti è la mancata adozione di una approccio metodologico, che minimizza i rischi di insuccesso essendo basato su un'analisi costruttiva degli errori commessi

## Fattori di rischio

- Rischi legati alla gestione del progetto
- Rischi legati alle tecnologie
- Rischi legati ai dati e alla progettazione
- Rischi legati all'organizzazione

- Il rischio di ottenere un risultato insoddisfacente nei progetti di data warehousing è particolarmente alto a causa delle elevatissime aspettative degli utenti
- Nella cultura aziendale contemporanea è infatti diffusissima la credenza che attribuisce al data warehousing il ruolo di panacea
- In realtà una larga parte della responsabilità della riuscita del progetto ricade sulla qualità dei dati sorgente e sulla lungimiranza, disponibilità e dinamismo del personale dell'azienda



## Approccio top-down

- Analizza i bisogni globali dell'intera azienda e pianifica lo sviluppo del DW per poi progettarlo e realizzarlo nella sua interezza
  - Promette ottimi risultati poiché si basa su una visione globale dell'obiettivo e garantisce in linea di principio di produrre un DW consistente e ben integrato
  - Il preventivo di costi onerosi a fronte di lunghi tempi di realizzazione scoraggia la direzione dall'intraprendere il progetto
  - Affrontare contemporaneamente l'analisi e la riconciliazione di tutte le sorgenti di interesse è estremamente complesso
  - Riuscire a prevedere a priori nel dettaglio le esigenze delle diverse aree aziendali impegnate è pressoché impossibile, e il processo di analisi rischia di subire una paralisi
  - Il fatto di non prevedere la consegna a breve termine di un prototipo non permette agli utenti di verificare l'utilità del progetto e ne fa scemare l'interesse e la fiducia

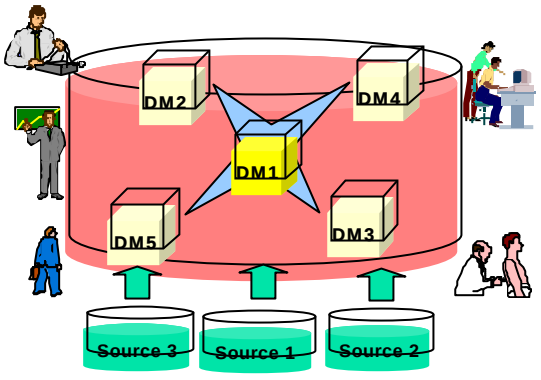
# Approccio bottom-up

- Il DW viene costruito in modo incrementale, assemblando iterativamente più data mart, ciascuno dei quali incentrato su un insieme di fatti collegati a uno specifico settore aziendale e di interesse per una certa categoria di utenti
  - Determina risultati concreti in tempi brevi
  - Non richiede elevati investimenti finanziari
  - Permette di studiare solo le problematiche relative al data mart in oggetto
  - Fornisce alla dirigenza aziendale un riscontro immediato sull'effettiva utilità del sistema in via di realizzazione
  - Mantiene costantemente elevata l'attenzione sul progetto
  - Determina una visione parziale del dominio di interesse

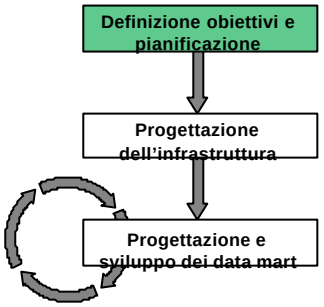


# Il primo data mart da prototipare...

- deve essere quello che gioca il ruolo più strategico per l'azienda
- deve ricoprire un ruolo centrale e di riferimento per l'intero DW
- si deve appoggiare su fonti dati già disponibili e consistenti



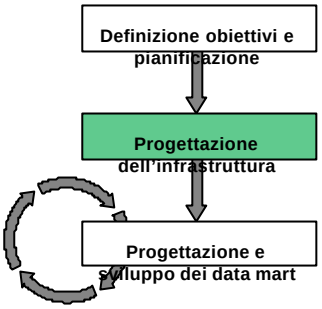
# Il ciclo di sviluppo




- individuazione degli obiettivi e dei confini del sistema
- stima delle dimensioni
- scelta dell'approccio per la costruzione
- valutazione dei costi e del valore aggiunto
- analisi dei rischi e delle aspettative
- studio delle competenze del gruppo di lavoro

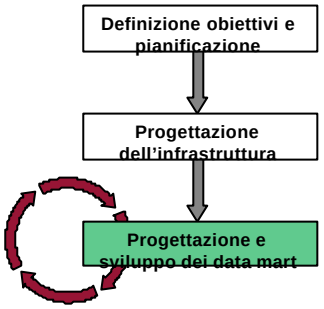


# Il ciclo di sviluppo




Si analizzano e si comparano le possibili soluzioni architetturali valutando le tecnologie e gli strumenti disponibili, al fine di realizzare un progetto di massima dell'intero sistema.

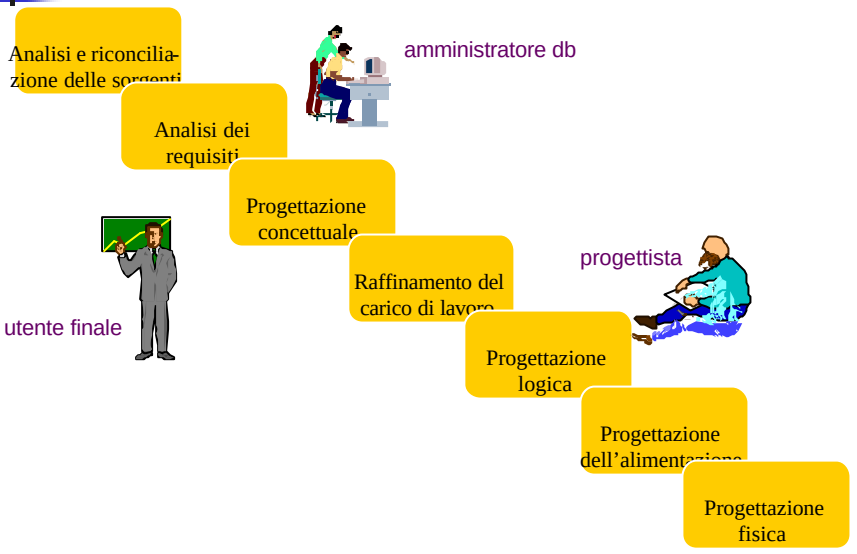
# Il ciclo di sviluppo



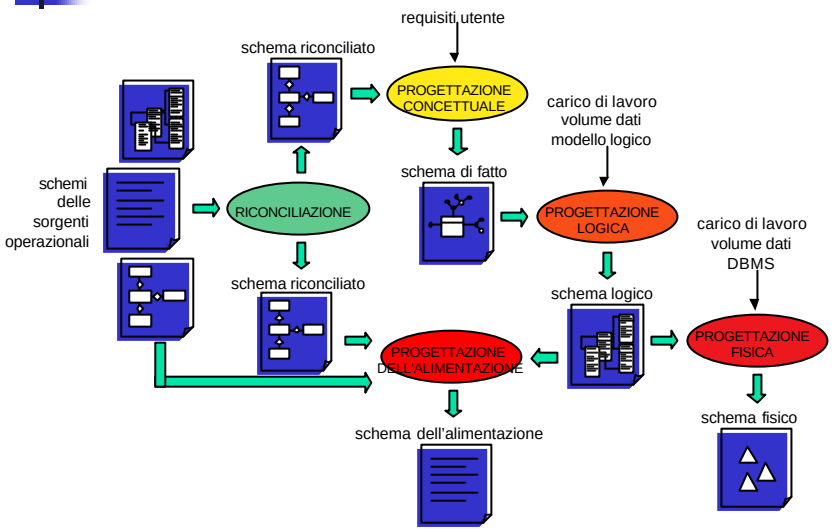

Ciascuna iterazione comporta la creazione di un nuovo data mart e di nuove applicazioni, che vengono via via integrate nel sistema di data warehousing.



# La progettazione del data mart



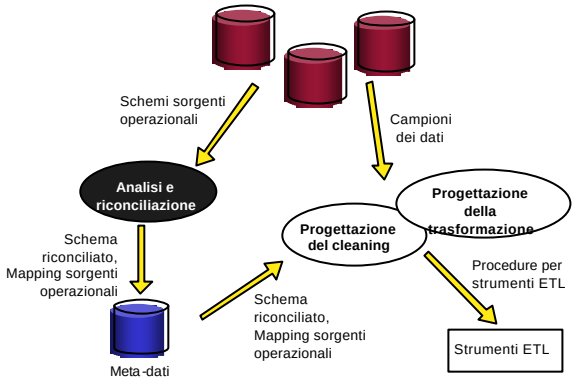
# La progettazione del data mart






# Analisi e riconciliazione delle sorgenti operazionali

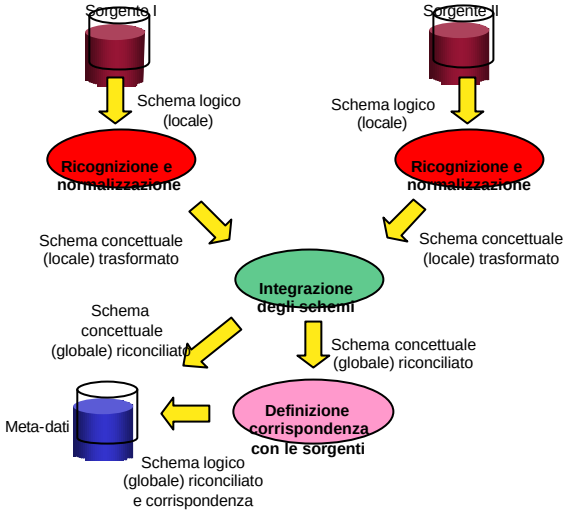
# Progettazione del livello riconciliato




- La fase di integrazione è incentrata sulla componente intensionale delle sorgenti operazionali, ossia riguarda la consistenza degli schemi che le descrivono
- Pulizia e trasformazione dei dati operano a livello estensionale, ossia coinvolgono direttamente i dati veri e propri



# Analisi e riconciliazione delle sorgenti operazionali

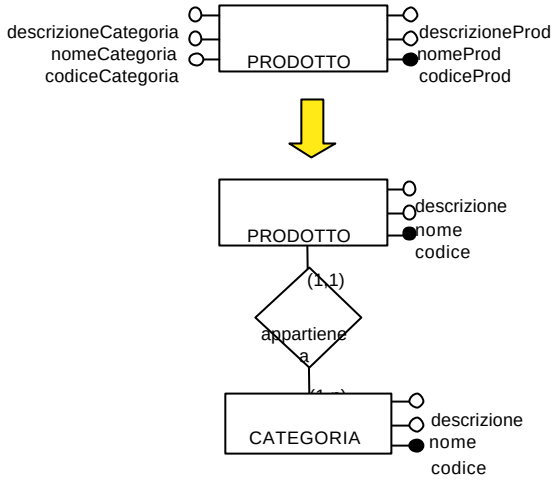


# Ricognizione e normalizzazione

- Il progettista, confrontandosi con gli esperti del dominio applicativo, acquisisce un'approfondita conoscenza delle sorgenti operazionali attraverso:
  - *ricognizione*, che consiste in un esame approfondito degli schemi locali mirato alla piena comprensione del dominio applicativo;
  - *normalizzazione*, il cui obiettivo è correggere gli schemi locali al fine di modellare in modo più accurato il dominio applicativo
- Ricognizione e normalizzazione devono essere svolte anche qualora sia presente una sola sorgente dati; qualora esistano più sorgenti, l'operazione dovrà essere ripetuta per ogni singolo schema



# Ricognizione e normalizzazione

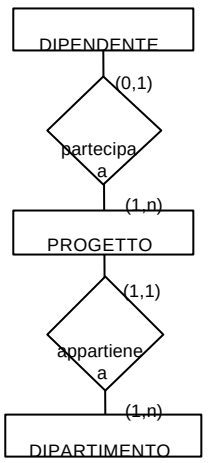


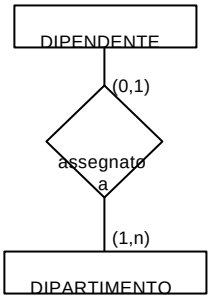
# Integrazione

- L'integrazione di un insieme di sorgenti dati eterogenee (basi di dati relazionali, file dati, sorgenti legacy) consiste nell'individuazione delle corrispondenze tra i concetti rappresentati negli schemi locali e nella risoluzione dei conflitti evidenziati, finalizzate alla creazione di un unico schema globale i cui elementi possano essere correlati con i corrispondenti elementi degli schemi locali (*mapping*)
- La fase di integrazione non si deve limitare a evidenziare le differenze di rappresentazione dei concetti comuni a più schemi locali, ma deve anche identificare l'insieme di concetti distinti e memorizzati in schemi differenti che sono correlati attraverso proprietà semantiche (*proprietà interschema*)
- Per poter ragionare sui concetti espressi negli schemi delle diverse sorgenti dati è necessario utilizzare un unico formalismo in modo da fissare i costrutti utilizzabili e la potenza espressiva

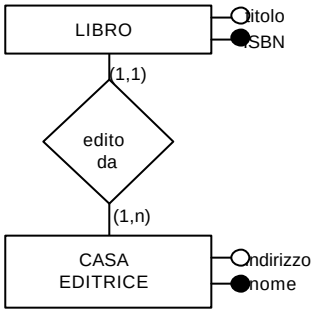


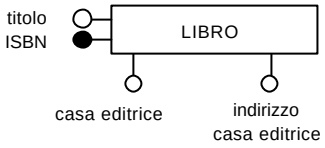
# Problemi: diversa prospettiva



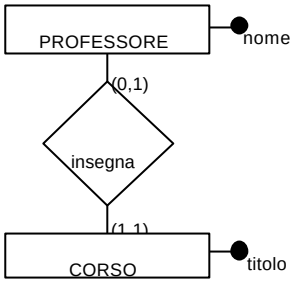
# Problemi: costrutti equivalenti

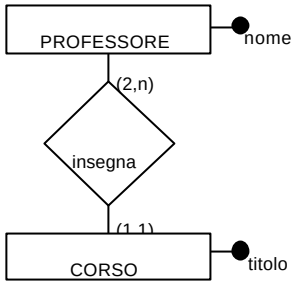





# Problemi: incompatibilità



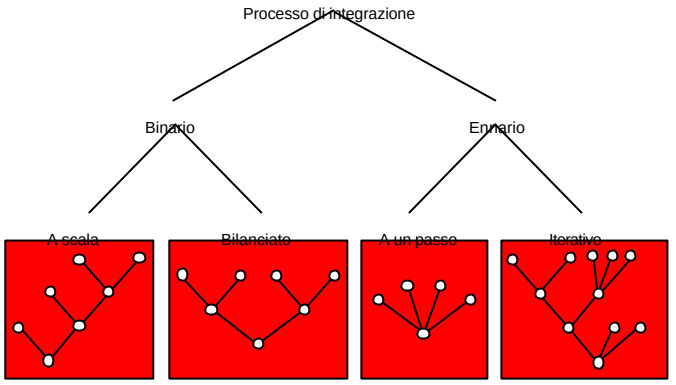
# Fasi dell'integrazione

1. *Preintegrazione*
2. *Comparazione degli schemi*
3. *Allineamento degli schemi*
4. *Fusione e ristrutturazione degli schemi*



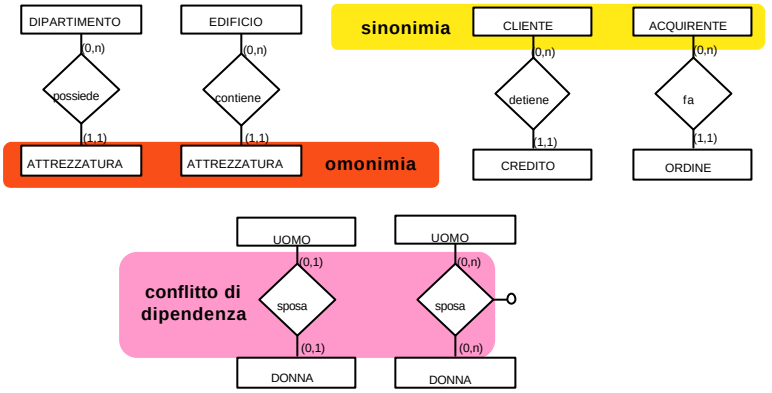
# 1. Preintegrazione

- Viene definita la strategia di integrazione



# 2. Comparazione degli schemi

- Un'analisi comparativa dei diversi schemi che mira a identificare le correlazioni e i conflitti tra i concetti in essi espressi






# 3. Allineamento degli schemi

- Scopo di questa fase è la risoluzione dei conflitti evidenziatisi al passo precedente, che si ottiene applicando primitive di trasformazione agli schemi sorgenti o allo schema riconciliato temporaneamente definito
  - Tipiche primitive di trasformazione riguardano il cambio dei nomi e dei tipi degli attributi, la modifica delle dipendenze funzionali e dei vincoli esistenti sugli schemi
  - Non sempre i conflitti possono essere risolti, poiché derivano da inconsistenze di base del sistema informativo; in questo caso la soluzione deve essere discussa con gli utenti che dovranno fornire indicazioni su qual è la più fedele interpretazione del mondo reale
  - In caso di incertezza si preferiscono le trasformazioni che avvantaggiano gli schemi ritenuti centrali nella struttura del data mart

## 4. Fusione degli schemi

- Gi schemi allineati vengono fusi a formare un unico schema riconciliato; l'approccio più diffuso è quello di sovrapporre i concetti comuni a cui saranno collegati tutti i rimanenti concetti provenienti dagli schemi locali.
- Dopo questa operazione si renderanno necessarie ulteriori trasformazioni mirate a migliorare la struttura dello schema riconciliato rispetto a:
  - Completezza
  - Minimalità
  - Leggibilità



## Analisi dei requisiti

## Obiettivi

- La fase di analisi dei requisiti ha l'obiettivo di raccogliere le esigenze di utilizzo del data mart espresse dai suoi utenti finali
- Essa ha un'importanza strategica poiché influenza le decisioni da prendere riguardo:
  - lo schema concettuale dei dati
  - il progetto dell'alimentazione
  - le specifiche delle applicazioni per l'analisi dei dati
  - l'architettura del sistema
  - il piano di avviamento e formazione
  - le linee guida per la manutenzione e l'evoluzione del sistema.



## Fonti

- La "fonte" principale da cui attingere i requisiti sono i futuri utenti del data mart (*business users*)
  - La differenza nel linguaggio usato da progettisti e utenti, e la percezione spesso distorta che questi ultimi hanno del processo di warehousing, rendono il dialogo difficile e a volte infruttuoso

- Per gli aspetti più tecnici, saranno gli amministratori del sistema informativo e/o i responsabili del CED a fungere da riferimento per il progettista
  - In questo caso, i requisiti che dovranno essere catturati riguardano principalmente vincoli di varia natura imposti sul sistema di data warehousing

# Le domande

| *Ruolo* | *Domande chiave* |
|---|---|
| Dirigente | Quali sono gli obiettivi aziendali?<br>Come misuri il successo della tua azienda?<br>Quali sono oggi i principali problemi dell'azienda?<br>In che modo ti aspetti che una maggiore disponibilità di informazioni possa migliorare la situazione aziendale? |
| Direttore di reparto | Quali sono gli obiettivi del tuo reparto?<br>Come misuri il successo del tuo reparto?<br>Descrivi i soggetti coinvolti nel tuo settore di interesse.<br>Ci sono colli di bottiglia nell'accesso ai dati?<br>Che analisi di routine esegui?<br>Che tipi di analisi ti piacerebbe poter eseguire?<br>A che livello di dettaglio occorre vedere le informazioni?<br>Quanta informazione storica è necessaria? |
| Amministratore del sistema informativo | Illustra le caratteristiche delle principali fonti dati disponibili.<br>Che strumenti vengono usati per analizzare i dati?<br>Come vengono gestite le richieste di analisi ad hoc?<br>Quali sono i principali problemi di qualità dei dati? |



# I fatti

- I fatti sono i concetti su cui gli utenti finali del data mart baseranno il processo decisionale; ogni fatto descrive una categoria di eventi che si verificano in azienda
  - Fissare le dimensioni di un fatto è importante poiché significa determinarne la granularità, ovvero il più fine livello di dettaglio a cui i dati saranno rappresentati. La scelta della granularità di un fatto nasce da un delicato compromesso tra due esigenze contrapposte: quella di raggiungere un'elevata flessibilità d'utilizzo e quella di conseguire buone prestazioni
  - Per ogni fatto occorre definire l'intervallo di storicizzazione, ovvero l'arco temporale che gli eventi memorizzati dovranno coprire

# I fatti

| | Data mart | Fatti |
|---|---|---|
| commerciale/ manufatturiero | approvvigionamento | acquisti, inventario di magazzino, distribuzione |
| | produzione | confezionamento, inventario, consegna, manifattura |
| | gestione domanda | vendite, fatturazione, ordini, spedizioni, reclami |
| | marketing | promozioni,fidelizzazione, campagne pubblicitarie |
| finanziario | bancario | conti correnti, bonifici, prestiti ipotecari, mutui |
| | investimenti | acquisto titoli, transazioni di borsa |
| | servizi | carte di credito,domiciliazioni bollette |
| sanitario | scheda di ricovero | ricoveri, dimissioni, interventi chirurgici, diagnosi |
| | pronto soccorso | accessi, esami, dimissioni |
| | medicina di base | scelte, revoche, prescrizioni |
| trasporti | merci | domanda, offerta, trasporti |
| | passeggeri | domanda, offerta, trasporti |
| | manutenzione | interventi |
| telecomunicazioni | traffico | traffico in rete, chiamate |
| | CRM | fidelizzazione, reclami, servizi |
| turismo | gestione domanda | biglietteria, noleggi auto, soggiorni |
| | CRM | frequent-flyers, reclami |
| gestionale | logistica | trasporti, scorte, movimentazione |
| | risorse umane | assunzioni, dimissioni, promozioni, incentivi |
| | budgeting | budget commerciale, budget di marketing |
| | infrastrutture | acquisti, opere |



# Glossario dei requisiti

| *Fatto* | *Possibili dimensioni* | *Possibili misure* | *Storicità* |
|---|---|---|---|
| inventario di magazzino | prodotto, data, magazzino | quantità in magazzino | 1 anno |
| vendite | prodotto, data, negozio | quantità venduta, importo, sconto | 5 anni |
| linee d'ordine | prodotto, data, fornitore | quantità ordinata, importo, sconto | 3 anni |

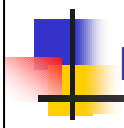
# Il carico di lavoro preliminare

- Il riconoscimento di fatti, dimensioni e misure è strettamente collegato all'identificazione di un *carico di lavoro preliminare*.
  - Oltre che dall'interazione diretta con l'utente, indicazioni al riguardo potranno essere ricavate da un esame della reportistica correntemente in uso in azienda.
  - In questa fase il carico di lavoro può essere espresso in linguaggio naturale; esso sarà comunque utile per valutare la granularità dei fatti e le misure di interesse, nonché per iniziare ad affrontare il problema dell'aggregazione



# Il carico di lavoro

| *Fatto* | *Interrogazione* |
|---|---|
| inventario di magazzino | Quantità media di ciascun prodotto presente mensilmente i tutti i magazzini.<br>Prodotti per i quali è stata esaurita la scorta contemporaneamente in tutti i magazzini in almeno un'occasione durante la settimana passata.<br>Andamento giornaliero delle scorte complessive per ciascu tipo di prodotto. |
| vendite | Quantità totali di ciascun tipo di prodotto vendute durante l'ultimo mese.<br>Incasso totale giornaliero di ciascun negozio.<br>Per un dato negozio, incassi relativi alle diverse categorie d prodotti durante un certo giorno.<br>Riepilogo annuale degli incassi per regione relativamente a dato prodotto. |
| linee d'ordine | Quantità totale ordinata annualmente presso un certo forn<br>Importo giornaliero ordinato nell'ultimo mese per un cert di prodotto.<br>Sconto massimo applicato da ciascun fornitore durante l'u anno per ciascuna categoria di prodotto. |



# Progettazione concettuale

## Quale formalismo?

- Mentre è universalmente riconosciuto che un DW si appoggia sul modello multidimensionale, non c'è accordo sulla metodologia di progetto concettuale.
- Il modello Entity/Relationship è molto diffuso nelle imprese come formalismo per la documentazione dei sistemi informativi relazionali, ma *non può essere usato per modellare il DW*.

# Il Dimensional Fact Model

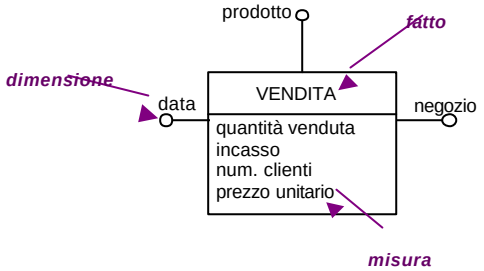
- Il DFM è un modello concettuale grafico per data mart, pensato per:
  - supportare efficacemente il progetto concettuale;
  - creare un ambiente su cui formulare in modo intuitivo le interrogazioni dell'utente;
  - permettere il dialogo tra progettista e utente finale per raffinare le specifiche dei requisiti;
  - creare una piattaforma stabile da cui partire per il progetto logico (*indipendentemente dal modello logico target*);
  - restituire una documentazione a posteriori espressiva e non ambigua.
- La rappresentazione concettuale generata dal DFM consiste in un insieme di schemi di fatto. Gli elementi di base modellati dagli schemi di fatto sono i fatti, le misure, le dimensioni e le gerarchie



# Il DFM: costrutti di base

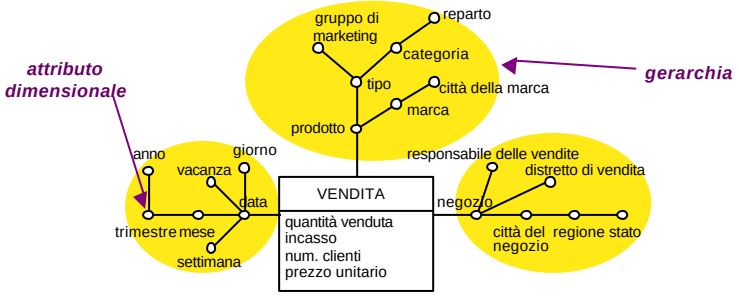
- Un *fatto* è un concetto di interesse per il processo decisionale; tipicamente modella un insieme di eventi che accadono nell'impresa (ad esempio: vendite, spedizioni, acquisti, ...). È essenziale che un fatto abbia aspetti dinamici, ovvero evolva nel tempo
- Una *misura* è una proprietà numerica di un fatto e ne descrive un aspetto quantitativo di interesse per l'analisi (ad esempio, ogni vendita è misurata dal suo incasso)
- Una *dimensione* è una proprietà con dominio finito di un fatto e ne descrive una coordinata di analisi (dimensioni tipiche per il fatto vendite sono prodotto, negozio, data)

Un fatto esprime una associazione molti-a-molti tra le dimensioni



# Il DFM: costrutti di base

- Con il termine generale *attributi dimensionali* si intendono le dimensioni e gli eventuali altri attributi, sempre a valori discreti, che le descrivono (per esempio, un prodotto è descritto dal suo tipo, dalla categoria cui appartiene, dalla sua marca, dal reparto in cui è venduto)
- Una *gerarchia* è un albero direzionato i cui nodi sono attributi dimensionali e i cui archi modellano associazioni molti-a-uno tra coppie di attributi dimensionali. Essa racchiude una dimensione, posta alla radice dell'albero, e tutti gli attributi dimensionali che la descrivono

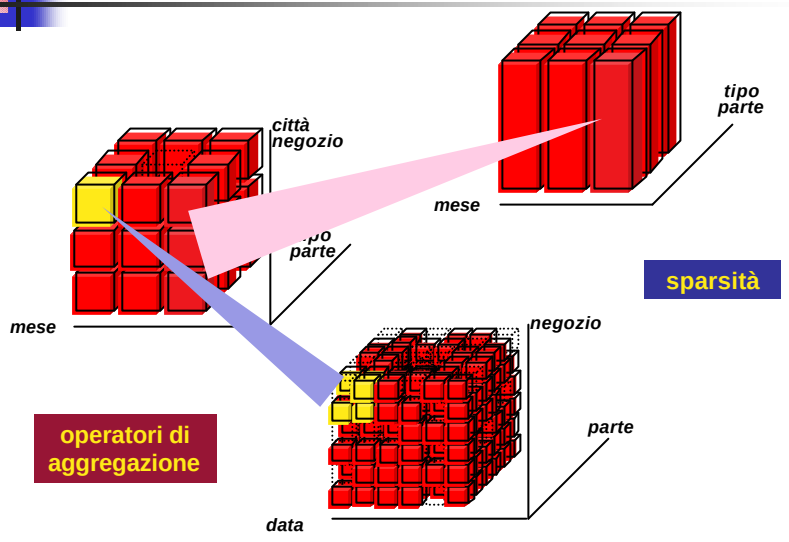





# Eventi e aggregazione

- Un *evento primario* è una particolare occorrenza di un fatto, individuata da una ennupla costituita da un valore per ciascuna dimensione. A ciascun evento primario è associato un valore per ciascuna misura
  - Con riferimento alle vendite, un possibile evento primario registra per esempio che, il 10/10/2001, nel negozio NonSoloPappa sono state vendute 10 confezioni di detersivo Brillo per un incasso complessivo pari a 25 euro
- Dato un insieme di attributi dimensionali (*pattern*), ciascuna ennupla di loro valori individua un *evento secondario* che aggrega tutti gli eventi primari corrispondenti. A ciascun evento secondario è associato un valore per ciascuna misura, che riassume in sé tutti i valori della stessa misura negli eventi primari corrispondenti
  - Pertanto, le gerarchie definiscono il modo in cui gli eventi primari possono essere aggregati e selezionati significativamente per il processo decisionale; mentre la dimensione in cui una gerarchia ha radice ne definisce la granularità più fine di aggregazione, agli altri attributi dimensionali corrispondono granularità via via crescenti

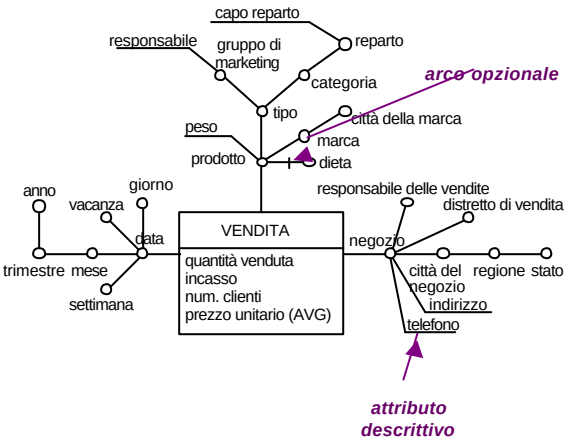
# Eventi e aggregazione





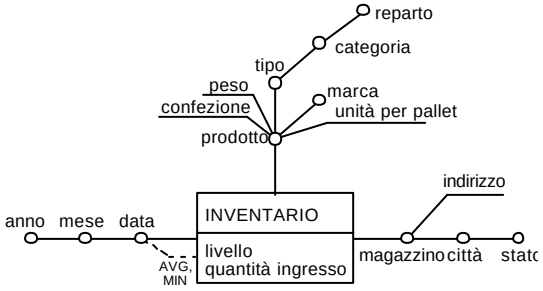

# Il DFM: costrutti avanzati

- Un *attributo descrittivo* contiene informazioni aggiuntive su un attributo dimensionale di una gerarchia, a cui è connesso da una associazione -a-uno. Non viene usato per l'aggregazione poiché ha valori continui e/o poiché deriva da un'associazione uno-a-uno
- Alcuni archi dello schema di fatto possono essere *opzionali*



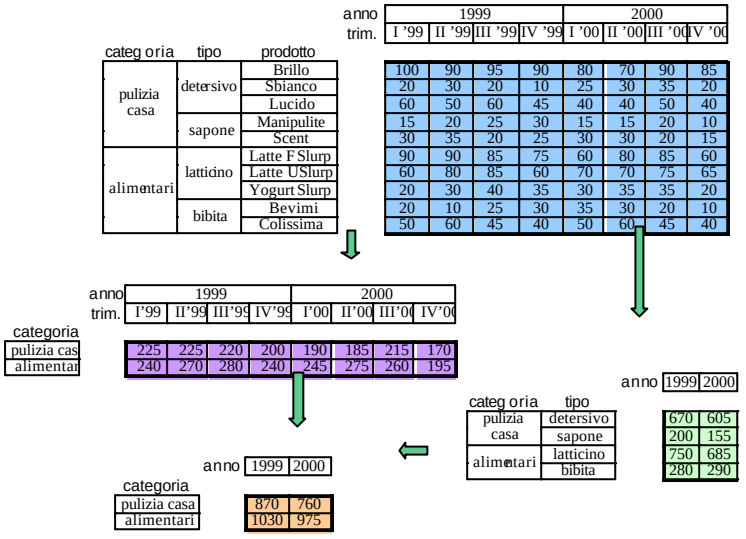
# Additività

- Una misura è detta additiva su una dimensione se i suoi valori possono essere aggregati lungo la corrispondente gerarchia tramite l'operatore di somma, altrimenti è detta non-additiva. Una misura non-additiva è non-aggregabile se nessun operatore di aggregazione può essere usato su di essa

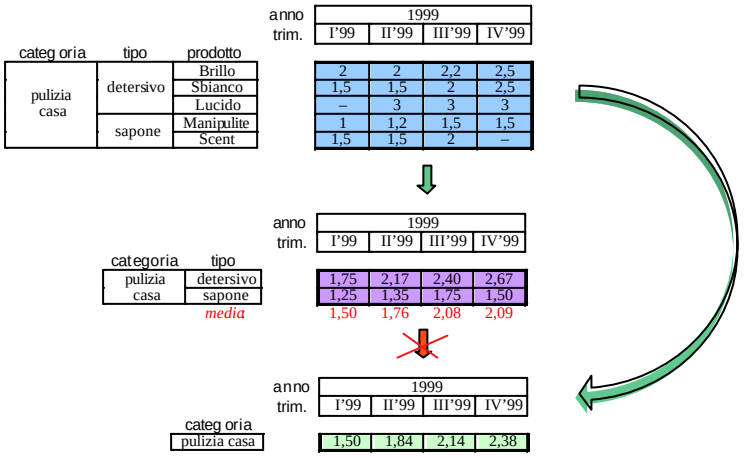





# Misure additive



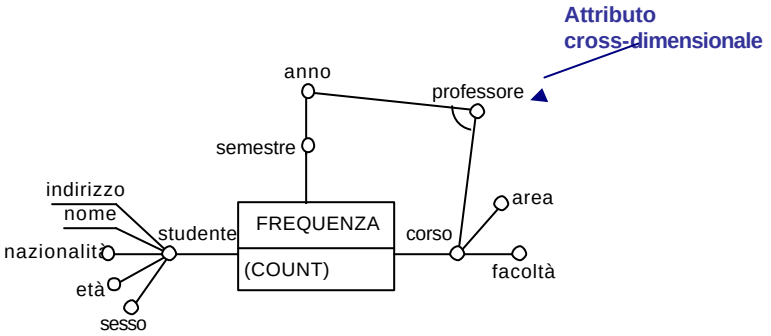
# Misure non-additive






# Schemi di fatto vuoti

- Uno schema di fatto si dice **vuoto** se non ha misure
  - In questo caso, il fatto registra solo il verificarsi di un evento



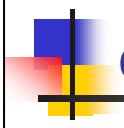
## Progettazione concettuale: approcci

- Basata sui requisiti
  - Il progettista deve essere in grado di enucleare, dalle interviste condotte presso l'utente, un'indicazione precisa circa i fatti da rappresentare, le misure che li descrivono e le gerarchie attraverso cui aggregarli utilmente. Il problema del collegamento tra lo schema concettuale così determinato e le sorgenti operazionali viene affrontato in un secondo tempo
- Basata sulle sorgenti 
  - È possibile definire lo schema concettuale in funzione della struttura delle sorgenti, evitando il complesso compito di stabilire il legame con esse a posteriori. Inoltre, è possibile derivare uno schema concettuale prototipale dagli schemi operazionali in modo pressoché automatico



## Progettazione concettuale: come

- La progettazione concettuale viene effettuata a partire dalla documentazione relativa al database riconciliato:
  - Schemi E/R
  - Schemi Relazionali
  - Schemi XML
  - .....
- Passi di progettazione:
  1. Definizione dei fatti
  2. Per ogni fatto:
     1. Costruzione di un *albero degli attributi*
     2. Editing dell'albero degli attributi
     3. Definizione delle dimensioni
     4. Definizione delle misure
     5. Creazione dello schema di fatto

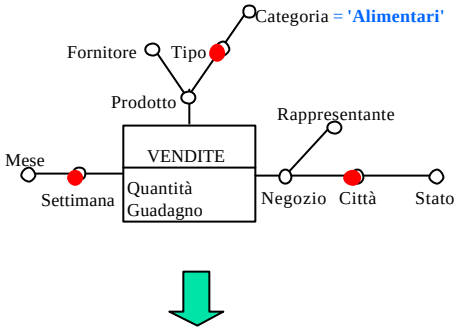


# Carico di lavoro e volume dati

# Il carico di lavoro

- Il carico di lavoro di un sistema OLAP è per sua natura estemporaneo
- È necessario identificare in fase di progettazione un carico di lavoro di riferimento
  - Reportistica standard
  - Colloqui con gli utenti
- Le interrogazioni OLAP sono facilmente caratterizzabili
  - Attributi di GROUP-BY
  - Misure richieste
  - Clausole di selezione

# Il carico di lavoro




*Totale della quantità venduta per i diversi tipi di prodotto, in ogni settimana e città ma solo per i prodotti alimentari*



# Dinamicità del carico di lavoro

- Il carico di lavoro preliminare non è di per sé sufficiente a ottimizzare le prestazioni del sistema
  - L'interesse degli utenti cambia nel tempo
  - Il numero di interrogazioni aumenta al crescere della confidenza degli utenti con il sistema
- Per ottimizzare la struttura logica del data mart è necessaria una fase di tuning attuabile solo dopo che il sistema è stato messo in funzione
- Il carico di lavoro reale può essere desunto dal log delle interrogazioni sottoposte al sistema

## Il volume dati

- Consiste nelle informazioni necessarie a determinare/stimare la dimensione del data mart.
  - Numero di valori distinti degli attributi nelle gerarchie
  - Lunghezza degli attributi
  - Numero di eventi di ogni fatto
- Deve essere calcolato considerando la quantità di dati necessari a coprire l'intervallo temporale deciso per il data mart.
- È utilizzato sia durante la progettazione logica sia durante la progettazione fisica per determinare:
  - la dimensione delle tabelle
  - la dimensione degli indici
  - i costi di accesso
- La bontà delle stime è spesso compromessa a causa del problema della sparsità.



## Progettazione logica

# Modelli logici per il Data Mart

- Mentre la modellazione concettuale è indipendente dal modello logico prescelto per l'implementazione, evidentemente lo stesso non si può dire per i temi legati alla modellazione logica.
- La struttura multidimensionale dei dati può essere rappresentata utilizzando due distinti modelli logici:
  - MOLAP (*Multidimensional On-Line Analytical Processing*) memorizzano i dati utilizzando strutture intrinsecamente multidimensionali (es. vettori multidimensionali).
  - ROLAP (*Relational On-Line Analytical Processing*) utilizza il ben noto modello relazionale per la rappresentazione dei dati multidimensionali.



# Sistemi MOLAP

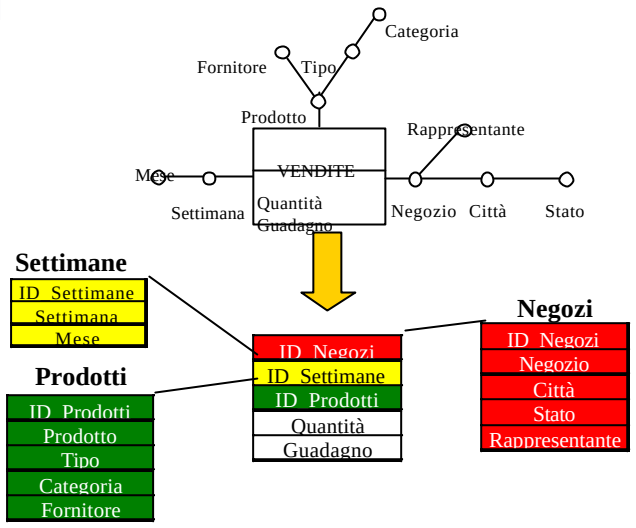
- L'utilizzo di soluzioni MOLAP:
  - Rappresenta una soluzione naturale e può fornire ottime prestazioni poiché le operazioni non devono essere "simulate" mediante complesse istruzioni SQL.
  - Pone il problema della sparsità: in media solo il 20% delle celle dei cubi contiene effettivamente informazioni, mentre le restanti celle corrispondono a fatti non accaduti.
  - È frenato dalla mancanza di strutture dati standard: i diversi produttori di software utilizzano strutture proprietarie che li rendono difficilmente sostituibili e accessibili mediante strumenti di terze parti.
  - Progettisti e sistemisti sono riluttanti a rinunciare alla loro ormai ventennale esperienza sui sistemi relazionali.

# ROLAP: lo schema a stella

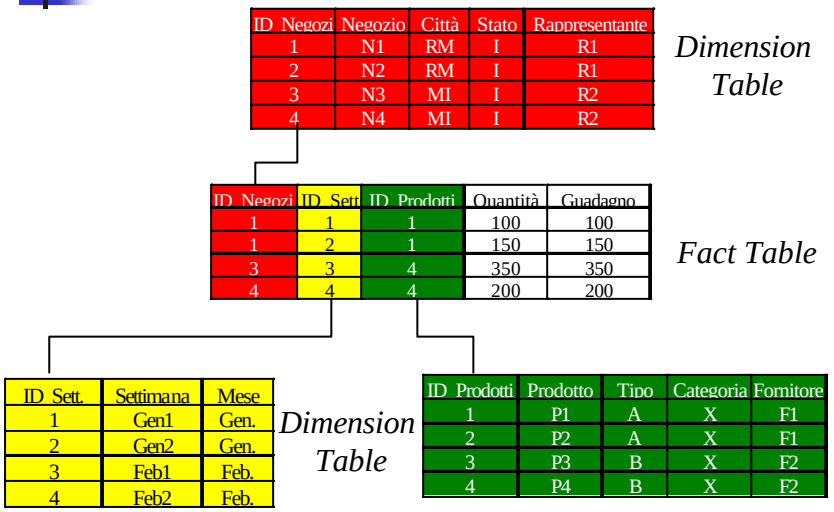
- La modellazione multidimensionale su sistemi relazionali è basata sul cosiddetto *schema a stella* (*star schema*) e sulle sue varianti.
- Uno schema a stella è composto da:
  - Un insieme di relazioni $DT_1$…, $DT_n$, chiamate *dimension table*, ciascuna corrispondente a una dimensione. Ogni $DT_i$ è caratterizzata da una chiave primaria (tipicamente surrogata) $d_i$ e da un insieme di attributi che descrivono le dimensioni di analisi a diversi livelli di aggregazione.
  - Una relazione *FT*, chiamata *fact table*, che importa le chiavi di tutte le dimension table. La chiave primaria di *FT* è data dall'insieme delle chiavi esterne dalle dimension table, $d_1$…, $d_n$; *FT* contiene inoltre un attributo per ogni misura.



# Lo schema a stella



# Lo schema a stella

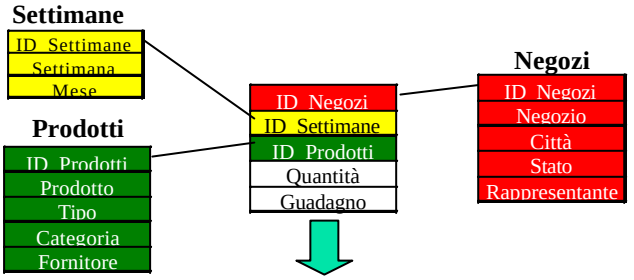





# Lo schema a stella: considerazioni

- Le Dimension Table sono completamente denormalizzate (es. Prodotto → Tipo)
  - È sufficiente un join per recuperare tutti i dati relativi a una dimensione
  - La denormalizzazione introduce una forte ridondanza nei dati
- La Fact Table contiene tuple relative a diversi livelli di aggregazione
  - L'elevata dimensione incide sui tempi di accesso ai dati
- Non si hanno problemi di sparsità in quanto vengono memorizzate soltanto le tuple corrispondenti a punti dello spazio multi-dimensionale per cui esistono eventi

## Interrogazioni OLAP su schemi a stella




VENDITE(Negozi.Città, Settimane, Prodotti.Tipo;
Prodotto.Categoria='Alimentari').Quantità

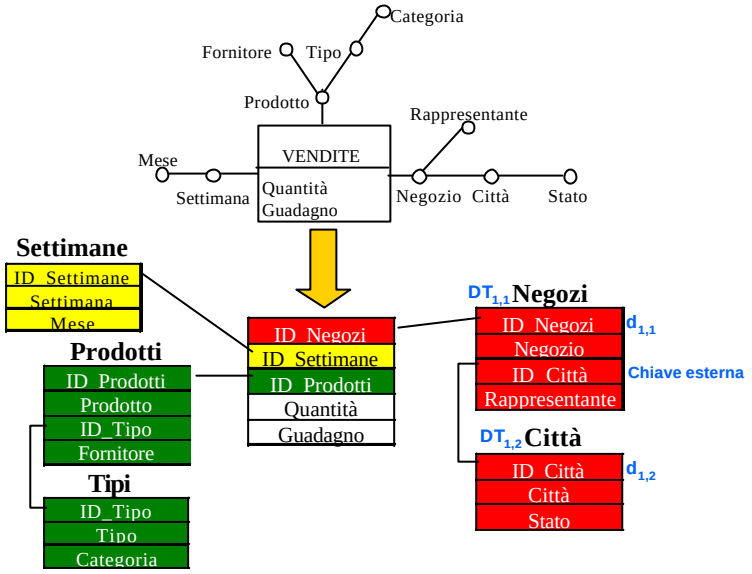
```
select    Città, Settimana, Tipo, sum(Quantità)
from      Settimane, Negozi, Prodotti, Vendite
where     Settimane.ID_Settimane=Vendite.ID_Settimane and
          Negozi.ID_ Negozi =Vendite.ID_ Negozi and
          Prodotti.ID_ Prodotti =Vendite.ID_ Prodotti and
          Prodotti.Categoria = 'Alimentari'
group by  Città, Settimana, Tipo;
```



## Lo snowflake schema

- Lo schema a fiocco di neve (*snowflake schema*) riduce la denormalizzazione delle dimension table $DT_i$ degli schemi a stella eliminando alcune delle dipendenze transitive che le caratterizzano.
- Le dimension table $DT_{i,j}$ di questo schema sono caratterizzate da:
  - una chiave primaria (tipicamente surrogata) $d_{i,j}$
  - il sottoinsieme degli attributi di $DT_i$ che dipendono funzionalmente da $d_{i,j}$.
  - zero o più chiavi esterne importate da altre $DT_{i,k}$ necessarie a garantire la ricostruibilità del contenuto informativo di $DT_i$.
- Denominiamo *primarie* le dimension table le cui chiavi sono importate nella fact table, *secondarie* le rimanenti.

# Lo snowflake schema






# Lo snowflake schema

| ID_Tipi | Tipo | Categoria |
|---|---|---|
| 1 | A | X |
| 2 | B | X |

| ID_Prodotti | Prodotto | Fornitore | ID_Tipi |
|---|---|---|---|
| 1 | P1 | F1 | 1 |
| 2 | P2 | F1 | 1 |
| 3 | P3 | F2 | 2 |
| 4 | P4 | F2 | 2 |

| ID_Sett. | Settima | Mese |
|---|---|---|
| 1 | Gen1 | Gen. |
| 2 | Gen2 | Gen. |
| 3 | Feb1 | Feb. |
| 4 | Feb2 | Feb. |

| ID_Negozi | ID_Sett | ID_Prodotti | Quantità | Guadagno |
|---|---|---|---|---|
| 1 | 1 | 1 | 100 | 100 |
| 1 | 2 | 1 | 150 | 150 |
| 3 | 3 | 4 | 350 | 350 |
| 4 | 4 | 4 | 200 | 200 |

| ID_Negozi | Negozio | ID_Città | Rappresentante |
|---|---|---|---|
| 1 | N1 | 1 | R1 |
| 2 | N2 | 1 | R1 |
| 3 | N3 | 2 | R2 |
| 4 | N4 | 2 | R2 |

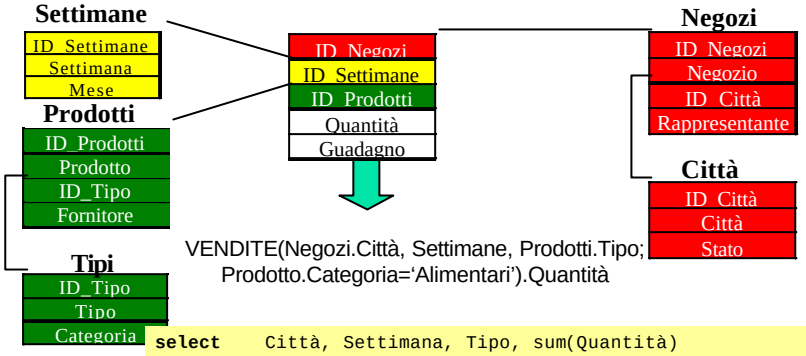
| ID_Città | Città | Stato |
|---|---|---|
| 1 | RM | I |
| 2 | MI | I |

# Lo snowflake schema: considerazioni

- Lo spazio richiesto per la memorizzazione dei dati si riduce grazie alla normalizzazione
- È necessario inserire nuove chiavi surrogate che permettano di determinare le corrispondenze tra dimension table primarie e secondarie
- L'esecuzione di interrogazioni che coinvolgono solo gli attributi contenuti nella fact table e nelle dimension table primarie è avvantaggiata
- Il tempo di esecuzione delle interrogazioni che coinvolgono attributi delle dimension table secondarie aumenta



# Interrogazioni OLAP su schemi snowflake




VENDITE(Negozi.Città, Settimane, Prodotti.Tipo; Prodotto.Categoria='Alimentari').Quantità

```
select    Città, Settimana, Tipo, sum(Quantità)
from      Settimane, Negozi, Città, Prodotti, Tipi, Vendite
where     Settimane.ID_Settimane=Vendite.ID_Settimane and
          Negozi.ID_ Negozi = Vendite.ID_ Negozi and
          Negozi.ID_ Tipo = Tipi.ID_ Tipo and
          Prodotti.ID_ Prodotti = Vendite.ID_ Prodotti and
          Prodotti.ID_ Città = Città.ID_ Città and
          Prodotti.Categoria = 'Alimentari'
group by  Città, Settimana, Tipo;
```

## Le viste

- L'analisi dei dati al massimo livello di dettaglio è spesso troppo complessa e non interessante per gli utenti che richiedono dati di sintesi
- L'aggregazione rappresenta il principale strumento per ottenere informazioni di sintesi
- L'elevato costo computazionale connesso con l'aggregazione induce a precalcolare i dati di sintesi maggiormente utilizzati

**Con il termine *vista* si denotano le fact table contenenti dati aggregati**



## Aggregate navigator

- La presenza di più fact table contenenti i dati necessari a risolvere una data interrogazione pone il problema di determinare la vista che determinerà il minimo costo di esecuzione.
- Questo ruolo è svolto dagli *aggregate navigator*, ossia i moduli preposti a riformulare le interrogazioni OLAP sulla "migliore" vista a disposizione.
- Gli aggregate navigator dei sistemi commerciali gestiscono attualmente solo gli operatori distributivi riducendo così l'utilità delle misure di supporto.

# Progettazione logica

- Include l'insieme dei passi che, a partire dallo schema concettuale, permettono di determinare lo schema logico del data mart

**INPUT**

Schema concettuale
Carico di lavoro
Volume dei dati
Vincoli di sistema


**OUTPUT**

Schema logico

- È basata su principi diversi e spesso in contrasto con quelli utilizzati nei sistemi operazionali
  - ✓ Ridondanza dei dati
  - ✓ Denormalizzazione delle relazioni



# Progettazione logica

- Le principali operazioni da svolgere durante la progettazione logica sono:

1. Scelta dello schema logico da utilizzare (es. star/snowflake schema)
2. Traduzione degli schemi concettuali
3. Scelta delle viste da materializzare
4. Applicazione di altre forme di ottimizzazione (es. frammentazione verticale/orizzontale)

## Dagli schemi di fatto agli schemi a stella

- La regola di base per la traduzione di uno schema di fatto in schema a stella prevede di:

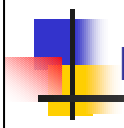
*Creare una fact table contenente tutte le misure e gli attributi descrittivi direttamente collegati con il fatto e, per ogni gerarchia, creare una dimension table che ne contiene tutti gli attributi.*



## Scelta delle viste

- La scelta delle viste da materializzare è un compito complesso, la soluzione rappresenta un trade-off tra numerosi requisiti in contrasto:

1. Minimizzazione di funzioni di costo
2. Vincoli di sistema
   - ✓ Spazio su disco
   - ✓ Tempo a disposizione per l'aggiornamento dei dati
3. Vincoli utente
   - ✓ Tempo massimo di risposta
   - ✓ Freschezza dei dati



# Progettazione dell'alimentazione

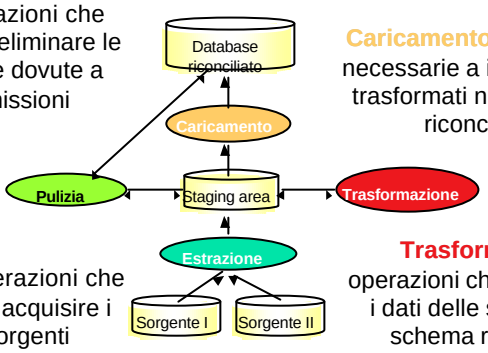
# Progettazione dell'alimentazione

- Durante la fase di progettazione dell'alimentazione vengono definite le procedure necessarie a caricare all'interno del data mart i dati provenienti dalle sorgenti operazionali.
  - Dalle sorgenti operazionali al livello riconciliato: realizzano a livello estensionale le trasformazioni definite nella fase di integrazione
  - Dal livello riconciliato al livello del data mart: si definiscono le procedure che permettono di conformare la struttura dei dati del livello riconciliato agli schemi a stella utilizzati in ambito multidimensionale

# Alimentazione dello schema riconciliato

**Staging area:** spazio utilizzato per memorizzare in via transitoria le informazioni necessarie all'esecuzione delle procedure

**Pulizia:** operazioni che permettono di eliminare le incongruenze dovute a errori e omissioni




**Caricamento:** operazioni necessarie a inserire i dati trasformati nel database riconciliato

**Estrazione:** operazioni che permettono di acquisire i dati dalle sorgenti

**Trasformazione:** operazioni che conformano i dati delle sorgenti allo schema riconciliato



# Caricamento dei dati

- La modalità di caricamento dei dati dalla staging area al database riconciliato dipende dalla tecnica utilizzata in fase di estrazione e dal livello di storicizzazione del livello riconciliato.
  - Estrazione statica → Riscrittura completa
  - Estrazione incrementale

# Pulizia dei dati

- Con questo termine si intende l'insieme delle operazioni atte a garantire la correttezza e la consistenza dei dati presenti nel livello riconciliato rispetto a:
  - ✓ Errori di battitura
  - ✓ Differenza di formato dei dati nello stesso campo
  - ✓ Inconsistenza tra valori e descrizione dei campi
    - Evoluzione del modo di operare dell'azienda
    - Evoluzioni della società
    - Convenzioni interne ai reparti e diverse da quelle generali del sistema informativo
  - ✓ Inconsistenza tra valori di campi correlati
    - Città='Bologna' Regione='Lazio'

La maggior parte delle inconsistenze può essere prevenuta rendendo più rigorose le regole di inserimento dei dati nelle applicazioni del sistema operazionale