DW: evita join, denormalizzazione, contiene dati storici, no istantanee come db operazionali

OLTP (db operazionali) vs OLAP (dw)

Albero degli attributi (da database)

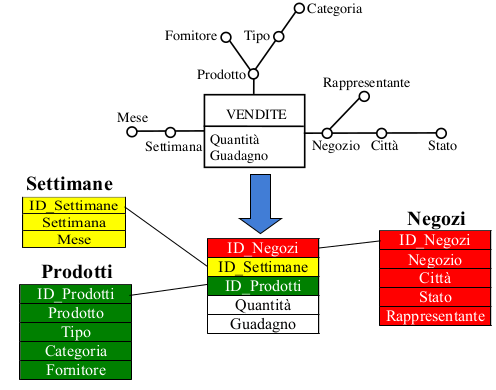
Schema di fatto (molto dinamici)

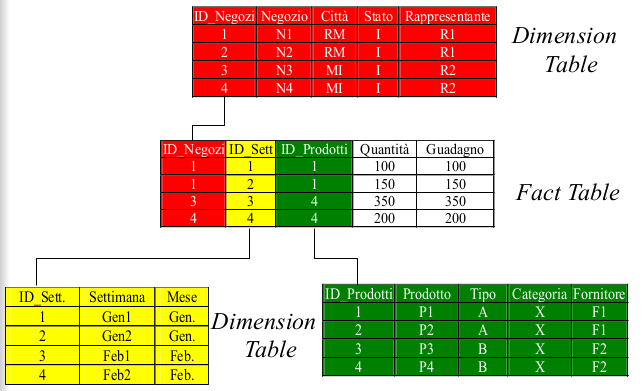
Progettazione logica (ROLAP, MOLAP, HYBRID)

ROLAP: schema a stella

schema a stella:

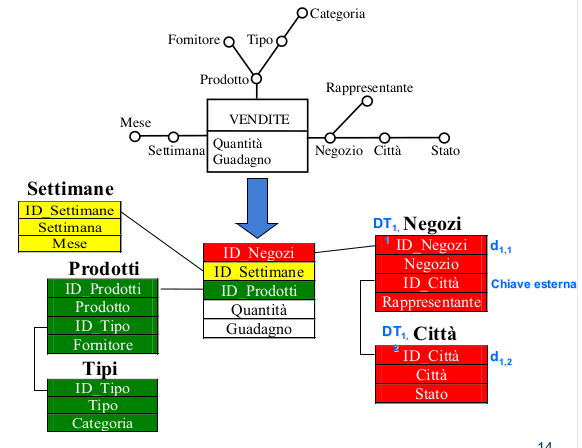
- basta solo un join per prendere tutti i dati da una dimensione

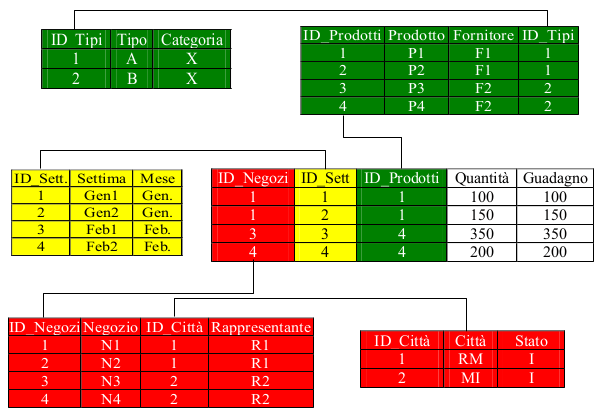




**Il portafoglio direzionale**

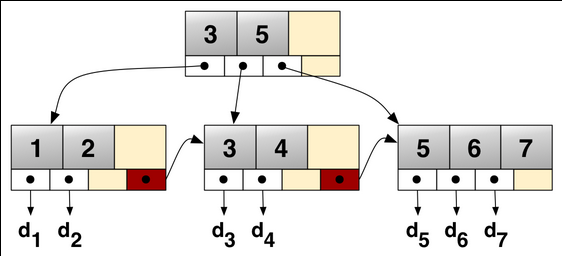
E l insieme delle applicazioni utilizzate dai manager aziendali per:



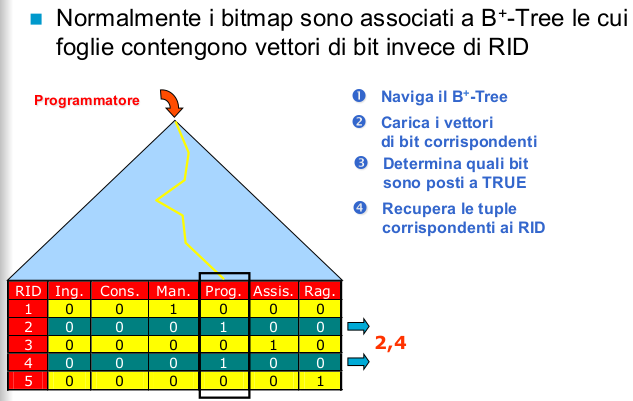


Indici fisici:

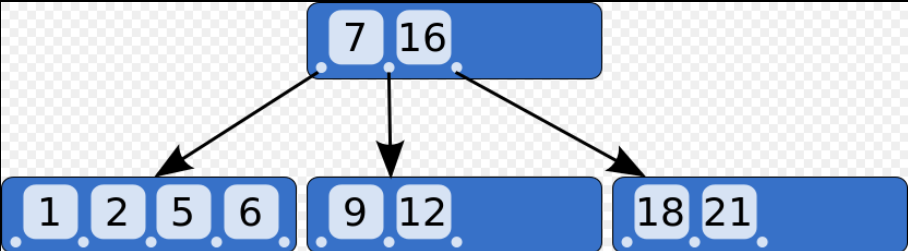
- B+ Tree



-Bitmap



-B Tree (ripasso, no dw)



" Analizzare lo stato dell azienda

" Prendere decisioni rapide

" Prendere le decisioni migliori

**Business intelligence**

Una definizione:

" La business intelligence è un insieme di strumenti

e procedure che consentono a un azienda di

trasformare i propri dati di business in informazioni

utili al processo decisionale, da rendere disponibili

alla persona giusta e nel formato idoneo

" Le informazioni così ottenute sono utilizzate dai

decisori aziendali per definire e supportare le

strategie di business, così da operare decisioni

consapevoli e informate con l'obiettivo di trarre

vantaggi competitivi, migliorare le prestazioni

operative e la profittabilità e, più in generale,

creare valore per l'azienda

**Si parla di piattaforma di BI**

poiché per consentire ai manager analisi potenti e flessibili è necessario

definire un apposita infrastruttura hardware e software di supporto composta da:

" Hardware dedicato

" Infrastrutture di rete

" DBMS

" Software di back-end

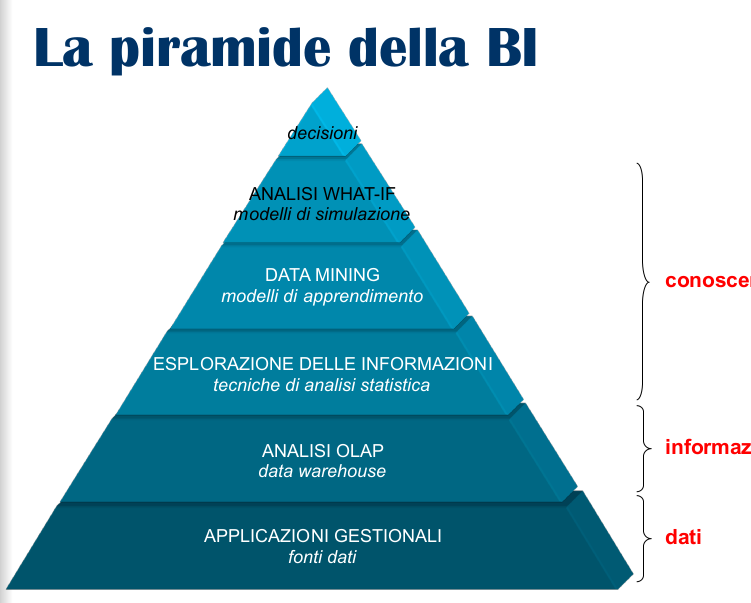
" Software di front-end

Il ruolo chiave di una piattaforma di business

intelligence è la trasformazione dei dati aziendali in

informazioni fruibili a diversi livelli di dettaglio e,

quindi, in conoscenza



**OLTP e OLAP**

**OLTP** (On-Line Transactional Processing):

" Le interrogazioni eseguono transazioni che leggono e

scrivono un ridotto numero di record da diverse tabelle

legate da semplici relazioni

" Il nucleo sostanziale del carico di lavoro è congelato

all interno dei programmi applicativi

**OLAP** (On-Line Analytical Processing):

" Le interrogazioni effettuano un analisi multidimensionale

che richiede la scansione di un enorme quantità di record

per calcolare un insieme di dati numerici di sintesi che

quantifichino le prestazioni dell azienda

" L interattività è una caratteristica irrinunciabile delle sessioni

di analisi e fa sì che il carico di lavoro effettivo vari

continuamente nel tempo

**Le lamentele**

abbiamo montagne di dati ma non possiamo accedervi!

# come è possibile che persone che svolgono lo stesso ruolo presentino risultati sostanzialmente diversi?

# vogliamo selezionare, raggruppare e manipolare i dati in ogni modo possibile!

# mostratemi solo ciò che è importante!

# tutti sanno che alcuni dati non sono corretti!

**Il Data Warehouse**

Un Data Warehouse è una collezione di dati

di supporto per il processo decisionale che

presenta le seguenti caratteristiche:

" è orientata ai soggetti di interesse;

(in OLTP era orientato alle applicazioni)

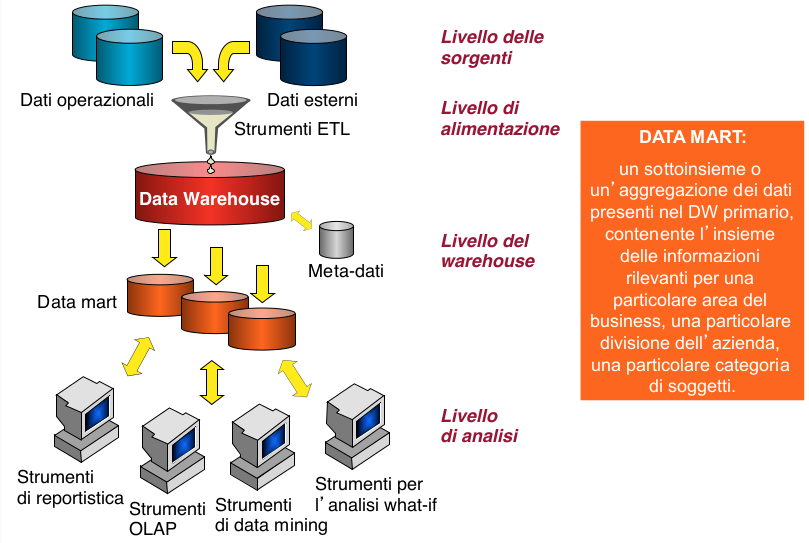
" è integrata e consistente;

" è rappresentativa dell evoluzione temporale;

" non volatile

(si mantiene la storia, non si fa delete, ma load)

**architettura a 2 livelli:**



I data mart alimentati dal DW primario sono detti dipendenti. Per

i sistemi collocati all interno di realtà aziendali medio-grandi

essi sono utili:

il DW è più grande, un data-mart è più piccolo

In alcuni contesti si preferisce adottare data mart alimentati

direttamente dalle sorgenti, detti **indipendenti**.

Venivano alimentati direttamente dai DB operazionali, ma c'era il problema della inconsistenza dei dati fra 2 DB diversi.

Il **data mart bus** è consigliato da Kimball, è come i data-mart indipendenti ma si hanno le dimensioni conformi, così non c'è l'inconsistenza.

L assenza di un DW primario snellisce le fasi progettuali, ma

determina uno schema complesso di accessi ai dati e ingenera il

rischio di inconsistenze tra i data mart

**Vantaggi:**

A livello del warehouse è continuamente disponibile informazione di

buona qualità anche quando, per motivi tecnici oppure

organizzativi, è temporaneamente precluso l accesso alle sorgenti

L interrogazione analitica effettuata sul DW non interferisce con la

gestione delle transazioni a livello operazionale, la cui affidabilità è

essenziale per il funzionamento dell azienda

L organizzazione logica del DW è basata sul modello

multidimensionale, mentre le sorgenti offrono in genere modelli

relazionali o semi-strutturati

C è una discordanza temporale e di granularità tra sistemi OLTP,

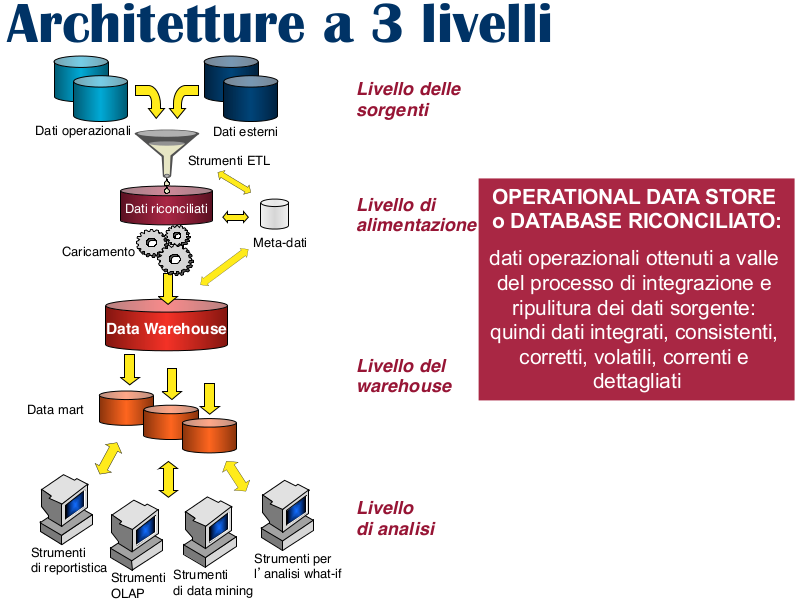
che trattano dati correnti e al massimo livello di dettaglio, e sistemi

OLAP che operano su dati storici e di sintesi

A livello del warehouse è possibile impiegare tecniche specifiche

per ottimizzare le prestazioni per applicazioni di analisi e

reportistica



Il vantaggio principale del livello dei dati

riconciliati è che esso crea un modello di dati

comune e di riferimento per l intera azienda,

introducendo al contempo una separazione

netta tra le problematiche legate

all estrazione e integrazione dei dati dalle

sorgenti e quelle inerenti l alimentazione del

DW

**Hub and spoke**

Come l'alrchitettura a 3 livelli, ma manca il DW, i datamart sono alimentati direttamente dal db riconciliato

**ETL**

Il ruolo degli strumenti di Extraction, Transformation and Loading è quello di alimentare una sorgente dati singola,

dettagliata, esauriente e di alta qualità che possa a sua volta

alimentare il DW (riconciliazione)

Durante il processo di alimentazione del DW, la riconciliazione

avviene in due occasioni:

**quando il DW viene popolato per la prima volta,**

**e periodicamente quando il DW viene aggiornato.**

" estrazione

" pulitura

" trasformazione

" caricamento

(**ECTL** = Extracion, Cleaning, Transormation and Loading)

**Estrazione (legge):**

L **estrazione** **statica** viene effettuata

quando il DW deve essere popolato per

la prima volta e consiste

concettualmente in una fotografia dei

dati operazionali

" L **estrazione** **incrementale** viene usata

per l aggiornamento periodico del DW,

e cattura solamente i cambiamenti

avvenuti nelle sorgenti dall ultima

estrazione

• basata sul log mantenuto dal DBMS operazionale

• basata su time-stamp

• guidata dalle sorgenti

applicazioni, trigger

**Pulitura**

Si incarica di migliorare la

qualità dei dati delle sorgenti

" dati duplicati

" inconsistenza tra valori logicamente associati

dati mancanti

uso non previsto di un campo

valori impossibili o errati

valori inconsistenti per la stessa entità dovuti a errori di battitura

**Trasformazione**

Converte i dati dal formato operazionale

sorgente a quello del DW. La corrispondenza

con il livello sorgente è complicata dalla

presenza di fonti distinte eterogenee, che

richiede una complessa fase di integrazione:

" presenza di testi liberi che nascondono informazioni importanti

" utilizzo di formati e convenzioni differenti per lo stesso dato

es: piazza, p.zza, …

**Per l alimentazione dei dati riconciliati:**

" conversione e normalizzazione (operano a livello di formato di memorizzazione e di unità di misura per uniformare i dati)

" matching (stabilisce corrispondenze tra campi equivalenti in sorgenti diverse)

" selezione (riduce il numero di campi e di record rispetto alle sorgenti)

**Per l alimentazione del DW:**

" la normalizzazione è sostituita dalla denormalizzazione

" si introduce l aggregazione, che realizza le opportune sintesi dei dati

**Caricamento**

" **Refresh**: i dati del DW vengono riscritti integralmente, sostituendo

quelli precedenti (tecnica utilizzata per popolare inizialmente il DW)

" **Update**: i soli cambiamenti

occorsi nei dati sorgente vengono

aggiunti nel DW (tecnica utilizzata

per l aggiornamento periodico del

DW)

**Progettazione dell'alimentazione:**

**Dalle sorgenti operazionali al livello riconciliato**:

realizzano a livello estensionale le trasformazioni definite nella fase di

integrazione



**Estrazione:**

**Statica**: il livello riconciliato viene ricreato ex-novo

**" Incrementale**: vengono aggiunti solo i dati prodotti dal sistema

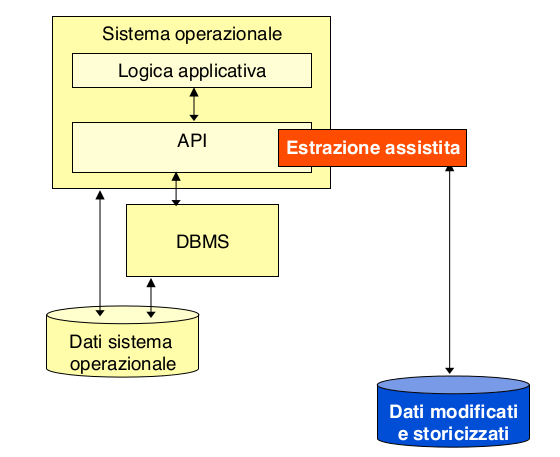
operazionale nell intervallo di tempo intercorso dall ultimo

caricamento

**• Immediata**

**• Ritardata**

**Estrazione guidata dalle applicazioni:**

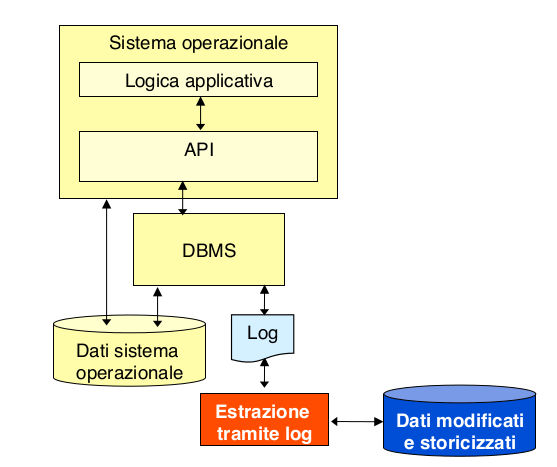


Tecnica di estrazione immediata

Le modifiche vengono rilevate da specifiche funzioni implementate direttamente all interno delle applicazioni

Fattibile anche nei sistemi moderni con le API in comune a tutte le applicazioni

**Estrazione basata sui log:**

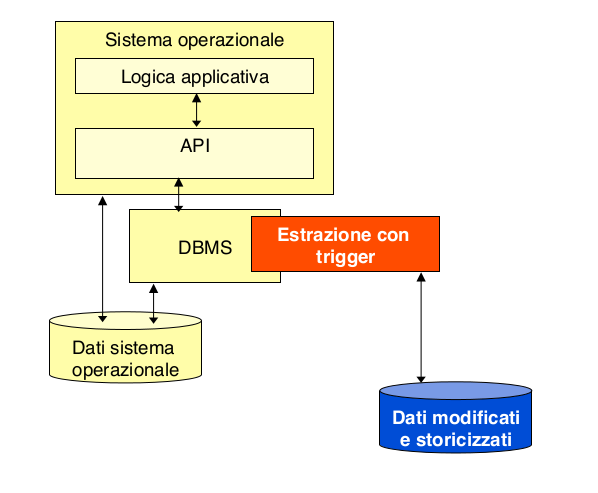


Tecnica di estrazione immediata

" Le modifiche vengono memorizzate in appositi file prodotti dal DBMS

È consigliabile solo quando il modulo di estrazione è fornito direttamente dal produttore del DBMS

**Estrazione basata su trigger:**



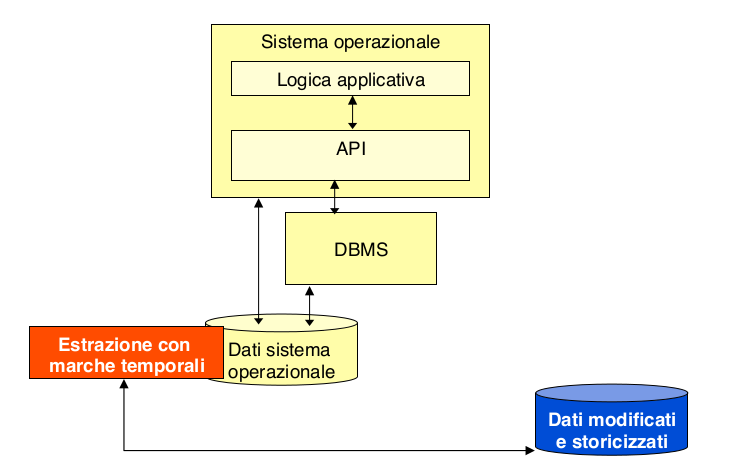
(lo stesso DBMS ha sia i database operazionali che il db riconciliato che il datawarehouse)

Tecnica di estrazione immediata

" Le modifiche vengono individuate mediante funzioni basate su eventi implementate e controllate direttamente nel DBMS

" Per motivi prestazionali non è possibile adottare in modo estensivo questa tecnica che richiederebbe al DBMS di monitorare continuamente tutte le transazioni potenzialmente in grado di innescare un trigger

**Estrazione con marche temporali:**



Tecnica di estrazione ritardata

" Prevede la modifica dello schema del database che dovrà contenere uno o

più campi necessari a contrassegnare i record modificati

" Il modulo di estrazione opera a posteriori individuando il tipo di modifica

subita dai dati

**NB**

se i dati sono **transitori** o **semi-storicizzati** l estrazione basata su marche temporali non può identificare gli stati intermedi di quei record modificati più volte durante l intervallo di aggiornamento

**Risultato dell'estrazione:**

Qualunque tecnica incrementale si utilizzi, il risultato della fase di estrazione consiste nell insieme di record della sorgente modificati, aggiunti o cancellati rispetto alla precedente esecuzione della procedura di estrazione

I dati risiedono nella staging area

Per facilitare le fasi successive è opportuno associare

a ogni record estratto il tipo di operazione

(**Inserimento, Modifica, Cancellazione**) che ne ha

generato la variazione

**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

• Livello riconciliato non storicizzato: memorizzo solo il tipo di

operazione che ha determinato la variazione

• Livello riconciliato storicizzato: memorizzo anche una coppia di

marche temporali che indicano l intervallo di validità della tupla

L architettura a 3 livelli presuppone che il livello di

storicizzazione dello schema riconciliato sia maggiore

o uguale a quello delle sorgenti operazionali

**Pulizia dei dati:**

La maggior parte delle inconsistenze può essere

prevenuta rendendo più rigorose le regole di

inserimento dei dati nelle applicazioni del

sistema operazionale

" 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

**Tecniche basate su dizionari**:

utilizzano tabelle di look-up per identificare ed eliminare sinonimi e abbreviazioni

• Utilizzabili solo quando il dominio dell attributo è conosciuto e limitato

• Utili per errori di battitura e discrepanze di formato

**" Tecniche ad hoc**:

ogni dominio applicativo ha regole proprie, troppo specifiche per essere verificate tramite strumenti standard

• Equazioni: profitto = guadagno - spese

• Outliers: variazione di prezzo di oltre il 20%

**" Tecniche di fusione approssimata:**

permettono di identificare record corrispondenti in assenza di identificatori comuni

• Join approssimati

• Purge/merge problem

**" 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

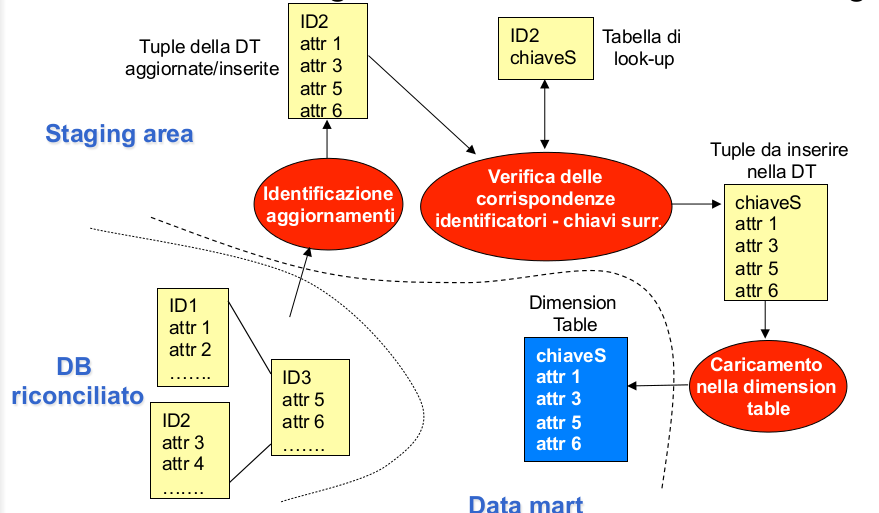
multidimensionale

**Alimentazione delle dimension table:**

Vengono identificati i dati dal db riconciliato da caricare.

Vengono sostituite le chiavi delle tuple con il nuovo surrogato se le tuple vengono aggiunte, altrimenti si trova la corrispondente tupla e la si aggiorna con una delle 3 gerarchie di storicizzazione.

Nella staging area rimane un collegamento fra il nuovo surrogato e l'id della tupla nel db riconciliato, così si possono gestire gli aggiornamenti delle tuple.



**Alimentazione delle Fact Table:**

Dopo la prima volta, ha sempre carattere incrementale, ci vorrebbe molto tempo per caricare tutta la FT ad ogni aggiornamento.

Segue l alimentazione delle dimension table per poter rispettare i vincoli di integrità referenziale

Identificazione dei dati da caricare

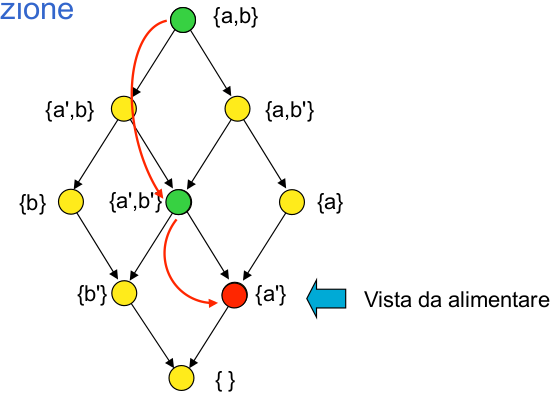
Sostituzione degli identificatori con chiavi surrogate

**Alimentazione delle viste:**

L'alimentazione delle viste segue quella della fact table.

Se è piccola, la si può cancellare e ricreare.

Altrimenti si aggiorna.



Per l'aggiornamento delle viste secondarie si segue l'ordinamento di roll-up tra pattern di aggregazione, si sceglie così una vista con meno tuple rispetto alla fact table.

**Slicing e Dicing:**

**Slicing**: prende una sola dimensione (una riga del cubo)

**Dicing**: filtra su un attributo (un sotto cubo)

**Aggregazione:** le celle del cubo diventano più grandi quando sono aggegate (sono meno dettagliati i dati)

**Tecniche di analisi dei dati**

**reportistica**: non richiede conoscenze informatiche

" **OLAP**: richiede all utente di ragionare in modo multidimensionale e

di conoscere l interfaccia dello strumento grafico utilizzato

Mentre gli utenti degli strumenti di reportistica svolgono un ruolo

essenzialmente passivo, gli utenti OLAP sono in grado di costruire

attivamente una sessione di analisi complessa in cui ciascun passo

effettuato è conseguenza dei risultati ottenuti al passo precedente

" estemporaneità delle sessioni di lavoro

" richiesta approfondita conoscenza dei dati

" complessità delle interrogazioni formulabili

" orientamento verso utenti non esperti di informatica

**Reportistica semi-statica:**

Un rapporto semi-statico, pur essendo focalizzato su un insieme di

informazioni predefinite, permette all analista alcuni gradi di libertà,

che si concretizzano nella possibilità di eseguire un ristretto

insieme di percorsi di navigazione

Vantaggi:

" agli utenti è richiesta una minor competenza sul modello dei dati e

sullo strumento di analisi rispetto al caso dell OLAP

" si elimina il rischio di creare risultati d analisi inconsistenti o

scorretti a causa dell uso improprio dei meccanismi di

aggregazione

" vincolando i tipi di analisi permessi si evita che l utente possa

involontariamente rallentare il sistema formulando interrogazioni

eccessivamente pesant

**Operatori OLAP:**

**Roll-Up:** aggrego, rendo i dati meno dettagliati

**Drill-down:** il contrario, rendo i dati più dettagliati

**Pivoting:** si ribalta il cubo, si invertono le dimensioni

**Drill-Across:** prende informazioni da 2 cubi diversi (frammentazione veriticale delle viste, si prendono un po' di attributi da una relazione e un po' da un'altra)

**ROLAP**

I dati sono memorizzati su un DBMS relazionale, in forma

dettagliata e pre-aggregata

" Occorre elaborare tipologie specifiche di schemi che permettano di

traslare il modello multidimensionale sul modello relazionale:

schema a stella

" Il problema delle prestazioni porta a denormalizzazione per evitare

costosi join

Il DW ROLAP è ottimale per memorizzare **enormi quantità di dati**

**MOLAP**

Basato su un modello logico ad hoc sul quale i dati e

le operazioni multidimensionali possono essere

direttamente rappresentati

! I dati vengono fisicamente memorizzati in vettori e l accesso è di tipo posizionale

c'è il problema della grande sparsità delle matrici

non c'è uno standard come per i RDBMS

I DM MOLAP massimizzano la **velocità di accesso ai dati**

**gestione della sparsità:**

**Suddivisione delle dimensioni**: consiste nel partizionare un

cubo n-dimensionale in più sottocubi n-dimensionali (chunk).

I singoli chunk potranno essere caricati più agevolmente in

memoria e potranno essere gestiti in modo differente a

seconda che siano densi (la maggior parte delle celle

contiene informazioni) oppure sparsi (la maggior parte delle

celle non contiene informazioni).

**Compressione dei chunk**: i chunk sparsi vengono

rappresentati in forma compressa al fine di evitare lo spreco

di spazio dovuto alla rappresentazione di celle che non

contengono informazioni.

**HOLAP (Hybrid OLAP)**

!

Sistemi di questo tipo combinano in un unica

architettura elementi di ROLAP e MOLAP

" Tipicamente i dati di dettaglio sono memorizzati su DBMS

relazionale, i pre-aggregati su strutture multidimensionali

proprietarie

" Oppure, **i sottocubi densi sono memorizzati in forma**

**multidimensionale, quelli sparsi in forma relazionale**

**Modello Multidimensionale:**

Gli oggetti che influenzano il processo decisionale sono i **fatti** del mondo aziendale.

Le occorrenze di un fatto sono gli **eventi** accaduti (es vendita,...)

Per ogni fatto sono interessanti le **misure** che descrivono quantitamente gli eventi

Ogni evento viene collocato in uno spazio multidimensionale a n dimensioni, i cui assi sono le **dimensioni di analisi.**

Ogni dimensione è associata ad una **gerarchia** di attributi dimensionali, che ne raggruppa i valori in diversi modi.

Implementazioni del modello multidimensionale sono ROLAP, MOLAP e HOLAP.

**DFM:**

Il DFM è un modello concettuale grafico per data mart

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

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

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

Dato un insieme di attributi dimensionali (group-by set), 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

**Datawarehouse: Approccio top down**

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

**Approcci guidati dai requisiti (demand-driven)**

" iniziano determinando i requisiti informativi degli utenti del data mart

" il problema di come creare una mappatura tra questi requisiti e le

sorgenti dati disponibili viene affrontato solo in seguito, attraverso

l implementazione di procedure ETL adatte

**Vantaggi**

" i desiderata degli utenti vengono portati in primo piano

**Svantaggi**

" è richiesto al progettista uno sforzo consistente durante il

disegno dell alimentazione

" fatti, misure e gerarchie vengono desunte direttamente dalle

specifiche dettate dagli utenti, e solo a posteriori si verifica

che le informazioni richieste siano effettivamente disponibili

nei database operazionali

la fiducia del cliente verso il progettista e verso l utilità del

data mart può venir meno

**Applicabilità**

" Questo approccio costituisce l unica alternativa nei casi in

cui non sia fattibile a priori un analisi approfondita delle

sorgenti (per esempio quando il data mart viene alimentato

da un sistema ERP), oppure qualora le sorgenti siano

rappresentate da sistemi legacy di tale complessità da

sconsigliarne la ricognizione e la normalizzazione

" E più difficilmente perseguibile dell approccio guidato dai

dati

**Datawarehouse: 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

**Approcci guidati dai dati (supply-driven)**

" progettano il data mart a partire da una dettagliata analisi delle

sorgenti operazionali

" i requisiti utente impattano sul progetto guidando il progettista nella

selezione delle porzioni di dati considerate rilevanti per il processo

decisionale, e determinando la loro strutturazione secondo il

modello multidimensionale

**Vantaggi**

" uno schema concettuale di massima per il data mart può

essere derivato algoritmicamente a partire dal livello dei dati

riconciliati, ossia in funzione della struttura delle sorgenti

" la progettazione dell ETL risulta notevolmente semplificata,

poiché ciascuna informazione nel data mart è direttamente

associata a uno o più attributi delle sorgenti

**Svantaggi**

" ai requisiti utente viene assegnato un ruolo secondario nel

determinare i contenuti informativi per l analisi

" al progettista viene dato un supporto limitato per l identificazione di fatti, dimensioni e misure

**Applicabilità**

" E applicabile quando:

1. è disponibile preliminarmente, oppure ottenibile con costi e

tempi contenuti, una conoscenza approfondita delle sorgenti

da cui il data mart si alimenterà;

2. gli schemi delle sorgenti mostrano un buon grado di

normalizzazione;

3. la complessità degli schemi delle sorgenti non è eccessiva

" Quando l architettura prescelta prevede l adozione di un

livello riconciliato questi requisiti sono soddisfatti: la

normalizzazione e la conoscenza approfondita sono

garantite dalla riconciliazione.

Lo stesso vale nel caso in cui

la sorgente si riduca a un singolo database, ben progettato

e di dimensioni limitate

" L esperienza di progettazione mostra che, qualora

applicabile, l approccio guidato dai dati risulta preferibile

agli altri poiché permette di raggiungere i risultati prefissati

in tempi estremamente contenuti.

**DB Riconciliato**

Viene fatta l'analisi e la riconciliazione delle sorgenti operazionali:

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

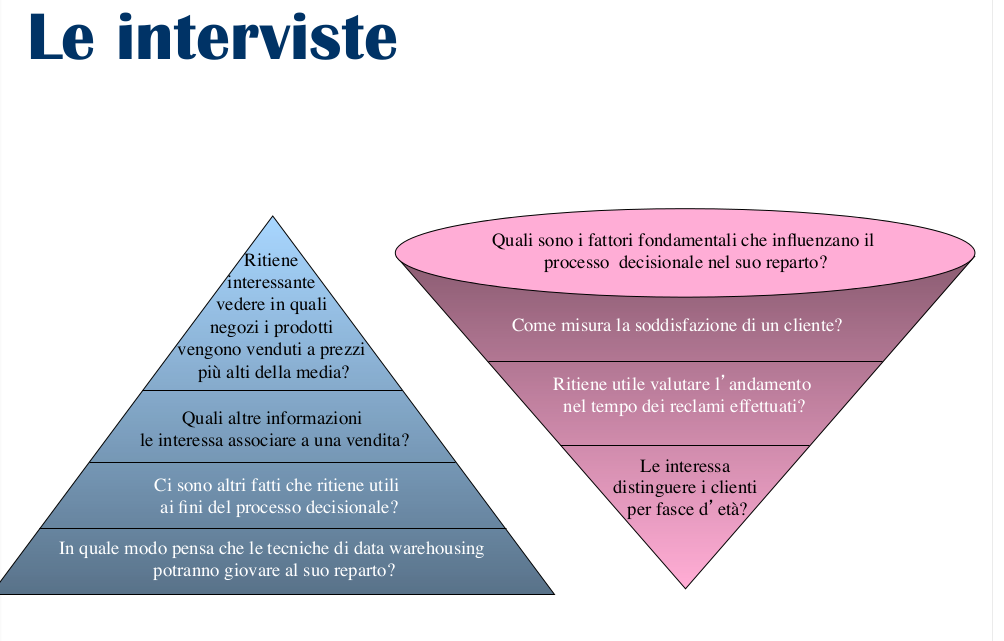
schemi locali mirato alla piena comprensione del dominio

applicativo;

" normalizzazione, il cui obiettivo è correggere gli schemi

locali al fine di modellare in modo più accurato il dominio applicativo

Poi viene fatta l'integrazione delle sorgenti operazionali, con un unico formalismo



**Additività**

**Misure di flusso:** si riferiscono a un periodo, al cui termine vengono

valutate in modo cumulativo (il numero di prodotti venduti in un

giorno, l incasso mensile, il numero di nati in un anno)

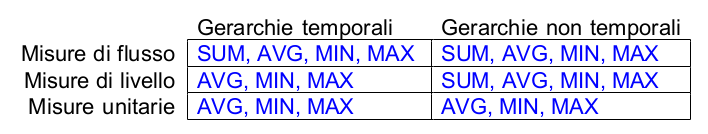
**" Misure di livello:** vengono valutate in particolari istanti di tempo (il

numero di prodotti in inventario, il numero di abitanti di una città)

**" Misure unitarie:** vengono valutate in particolari istanti di tempo, ma

sono espresse in termini relativi (il prezzo unitario di un prodotto, la

percentuale di sconto, il cambio di una valuta)



Uno schema di fatto si dice vuoto se non ha misure

**Transazionale vs. istantaneo**

**Uno schema transazionale...**

" ...è la soluzione migliore se, nel dominio applicativo, gli eventi sono

misurati come flussi entranti e uscenti (delta)

" non può essere adottato se gli eventi sono misurati come livelli, a meno

che non sia possibile decomporli univocamente in flussi

!

**Uno schema istantaneo...**

" ...è la soluzione migliore se, nel dominio applicativo, gli eventi sono

misurati come livelli

" può essere adottato anche quando gli eventi sono misurati come flussi, se

è nota la funzione che compone i flussi per determinare i livelli; in questo

caso, può comportare perdita di informazione

Passi di progettazione:

➊ Definizione dei fatti

➋ 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**

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

**schema a stella:**

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.

Le Dimension Table sono completamente denormalizzate (es. Prodotto → Tipo)

# È sufficiente un join per recuperare tutti i dati relativi a una

dimensione

$ C è una forte ridondanza nei 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

**Snowflake skema:**

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 a 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.

**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

**Viste:**

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

**Misure derivate**: ottenute applicando operatori matematici a

due o più valori appartenenti alla stessa tupla (es guadagno)

**Misure di supporto**: sono necessarie in presenza di

operatori di aggregazione non distributivi

**Operatori di aggregazione:**

**Distributivi**: permettono di calcolare dati aggregati a partire

direttamente da dati parzialmente aggregati (es. somma,

massimo, minimo)

" **Algebrici**: richiedono un numero finito di informazioni

aggiuntive (misure di supporto) per calcolare dati aggregati a

partire da dati parzialmente aggregati (es. media – richiede il

numero dei dati elementari che hanno contribuito a formare

un singolo dato parzialmente aggregato)

" **Olistici**: non permettono di calcolare dati aggregati a partire

da dati parzialmente aggregati utilizzando un numero finito di

informazioni aggiuntive (es. mediana, moda)

**Constellation schema:**

Una soluzione intermedia rispetto alle due presentate prevede di

memorizzare in fact table separate dati relativi a group-by set

diversi senza però ricorrrere alla normalizzazione delle dimension

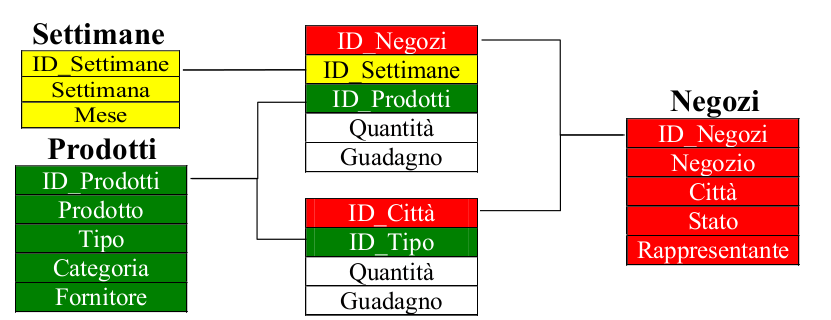
table (constellation schema)

" L accesso alle fact table è ottimizzato, quello alle dimension table no

" La dimensione delle fact table è di molto superiore a quella delle

dimension table e conseguentemente la loro ottimizzazione gioca un

ruolo fondamentale

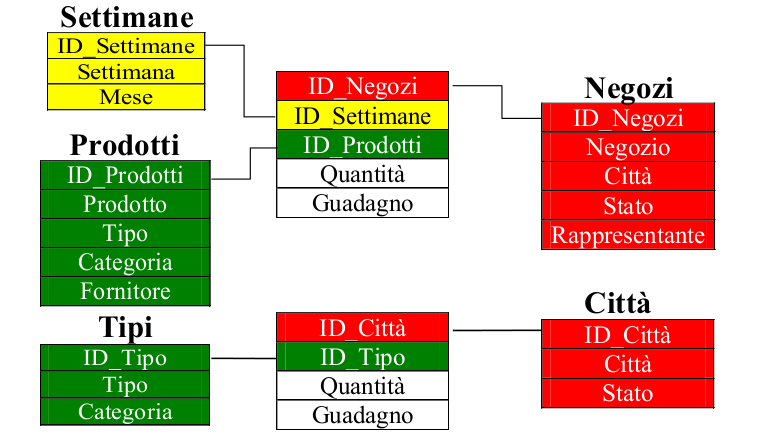


il massimo livello delle prestazioni si ottiene

memorizzando in fact table separate dati a diversi

livelli di aggregazione e replicando completamente

anche le dimension table



**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

**Scenari Temporali:**

**Attualizzazione:**

si fa l'update direttamente

**Verità storica:**

si scrive la nuova tupla aggiornata con un nuovo surrogato, così tutti i join della fact table cambiano automaticamente.

**Storicizzazione:**

Gli elementi necessari per la gestione di una gerarchia di tipo 3

sono:

" Una coppia di marche temporali (time-stamp) che indichino

l intervallo di validità di una tupla

" Un meccanismo per individuare le tuple coinvolte in una serie di

modifiche (tramite per esempio un attributo master)

**da – a – Master**

**Star VS Snowflake**

Può essere utile

1) Quando il **rapporto** tra le **cardinalità** della dimension table

primaria e secondaria è **elevato**, poiché determina un

**forte risparmio di spazio**

2) Quando una porzione di una gerarchia è comune a più

dimensioni

3) In presenza di viste aggregate (**constellation schema**)

4) Quando una parte della gerarchia è soggetta a frequenti

aggiornamenti

**Aggregazione e costrutti DFM:**

**Archi opzionali:**

Alcune porzioni delle gerarchie possono essere opzionali

Nella dimension table, nelle righe per cui non è definito un

valore viene inserito un valore fittizio (NULL oppure NON

APPLICABILE)

A causa dei vincoli di integrità, l opzionalità di

un intera gerarchia NON può essere gestita

introducendo un valore nullo nella chiave esterna

della fact table, occorre invece inserire un intera

tupla fittizia nella dimension table

**Gerarchie convergenti:**

si importano nella stessa dimension table, zero problemi

**Gerarchie condivise:**

se condividono tutti gli attributi, si hano 2 chiavi esterne nella FT

se condividono una parte degli attributi, si fa lo snowflake

**Archi multipli:**

si fa una bridge table con gli attribut coinvolti nella relazione molto a molti.

Si aggiunge l'attributo peso per dare un'importanza diversa alle tuple.

Si possono fare interrogazioni d'impatto (senza considerare i pesi) o pesate.

Se si fa il push down nella fact table (nel caso di un arco multiplo che entra in una dimensione) sono costosi gli aggiornamenti dei pesi in tutte le tuple della FT

Oppure si può eliminare l'arco multiplo rendendo più fine il fatto (es. ricovero per diagnosi)

**Con la soluzione con push-down:**

" Si introduce una forte ridondanza nella fact-table le cui righe

devono essere replicate tante volte quante sono le

corrispondenze dell arco multiplo

" Il peso è codificato permanentemente all interno della fact

table e il suo aggiornamento può risultare molto complesso

" Le interrogazioni di impatto risultano molto complesse

" Il costo di esecuzione delle interrogazioni si riduce grazie al

minor numero di join necessari

" Il calcolo degli eventi pesati avviene durante l alimentazione

**Con la soluzione con bridge-table:**

" Il costo di esecuzione delle interrogazioni si riduce a causa

del minor numero di tuple coinvolte

" Il calcolo degli eventi pesati avviene durante l interrogazione

**Dimensioni degeneri:**

Se la **lunghezza** dell **attributo non è eccessiva** può convenire

**evitare la creazione** di una specifica **dimension table**

importando direttamente i **valori dell attributo nella fact table**

Una soluzione alternativa è quella di utilizzare un unica

dimension table per modellare più dimensioni degeneri (junk

dimension)

" In una junk dimension non esiste alcuna dipendenza

funzionale tra gli attributi per cui risultano valide tutte le

possibili combinazioni di valori

" Questa soluzione risulta attuabile solo quando il **numero di**

**valori** distinti per gli attributi coinvolti è **limitato**

**Gerarchie Incomplete:**

Questo termine indica una gerarchia in cui per alcune istanze

risultano assenti uno o più livelli di aggregazione

Vengono gestite a livello estensionale inserendo opportuni

valori fittizi

È necessario mantenere la consistenza rispetto all operatore di roll-up

Sono possibili più soluzioni che si differenziano per il tipo di segnaposto inserito

**Bilanciamento per esclusione**:

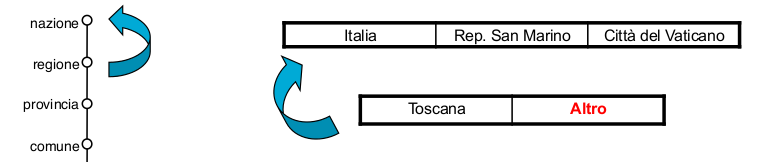
in tutte le tuple viene inserito un segnaposto generico (es. altro )

Preferibile quando il numero di dati mancanti è elevato

Questa soluzione viola la semantica del roll-up poiché

aggregando i dati si avrà un maggior livello di dettaglio delle

informazioni

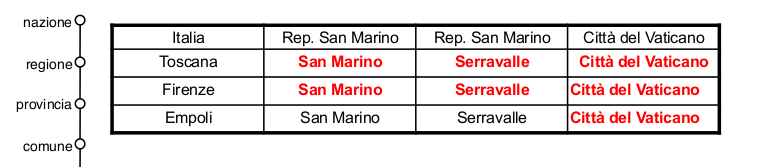


**Bilanciamento verso il basso:**

i valori mancanti vengono rimpiazzati con il valore dell attributo che lo precede nella gerarchia

Preferibile quando il numero di dati mancanti è limitato

L interpretazione dei report è complicata dal fatto che appariranno valori non corrispondenti al livello di aggregazione prescelto



**Bilanciamento verso l'alto:**

i valori mancanti vengono rimpiazzati con i valori dell attributo che lo segue nella gerarchia

Preferibile quando il numero di dati mancanti è elevato

Rispetto alla soluzione precedente i report risultano più

leggibili perché presentano un numero inferiore di valori

**Gerarchie Ricorsive:**

Questo termine indica una gerarchia in cui il numero dei livelli

di aggregazione non è codificabile nello schema e può variare

da istanza a istanza

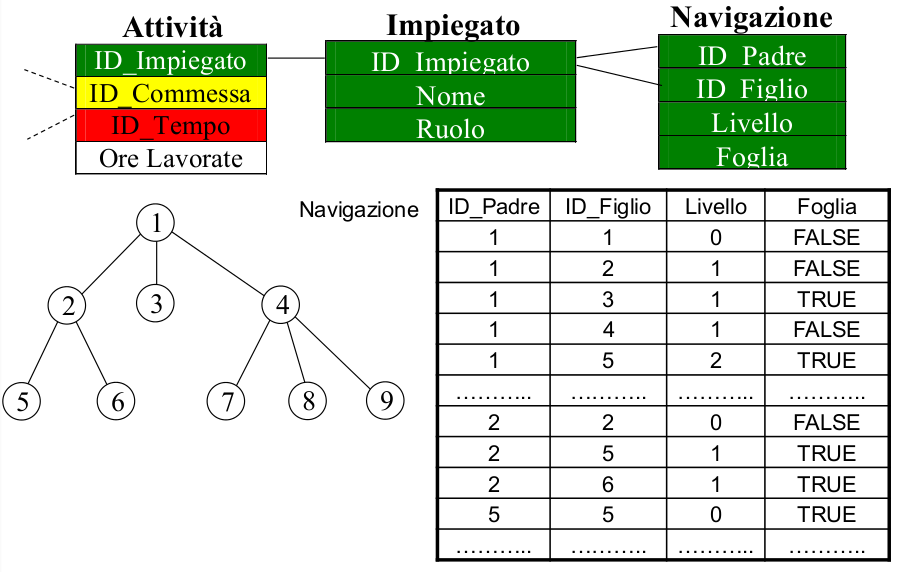
Non può essere modellata tramite schema a stella

Una possibile soluzione prevede l utilizzo di un autoanello

Non è ottimale nei RDBMS perché SQL **non** è ricorsivo.

**Soluzione 2**

Una soluzione più potente prevede di appiattire la gerarchia esplicitando tutti i legami da essa indotti in una tabella di navigazione



La dimensione della tabella di navigazione cresce in modo

esponenziale con la profondità della gerarchia

Se la dimensione della tabella è trattabile questa soluzione

garantisce un maggiore potere espressivo

**Scelta delle viste:**

Una vista è una Fact Table secondaria (vedi constellation schema)

Minimizzazione di funzioni di costo (es costo di esecuzione query)

Vincoli di sistema

" Spazio su disco

" Tempo a disposizione per l aggiornamento dei dati

Vincoli utente

" Tempo massimo di risposta

" Freschezza dei dati

**È utile** materializzare una vista quando:

" Risolve direttamente una interrogazione frequente

" Permette di ridurre il costo di esecuzione di molte interrogazioni

**Non è consigliabile** materializzare una vista quando:

" Il suo group-by set è molto simile a quello di una vista già

materializzata

" Il suo group-by set è molto fine

" La materializzazione non riduce di almeno un ordine di

grandezza il costo delle interrogazioni

**Frammentazione delle viste:**

**Frammentazione orizzontale**

! È la forma di frammentazione maggiormente utilizzata.

! I criteri di selezione delle tuple da inserire nei frammenti sono

determinati in base alle condizioni di selezione maggiormente

utilizzate a uno specifico livello di aggregazione.

! L attributo maggiormente utilizzato a tal fine è **il tempo** che,

oltre a essere largamente coinvolto nelle interrogazioni,

permette una facile gestione degli aggiornamenti.

! La riduzione dei tempi di esecuzione delle interrogazioni è

dovuta alla possibilità di operare su fact table più piccole e su

cui è già stata operata una (parziale) selezione.

! A differenza della frammentazione verticale quella orizzontale

non comporta alcun costo aggiuntivo in termini di spazio

richiesto per la memorizzazione dei dati.

**Frammentazione verticale:**

La frammentazione verticale costituisce una soluzione più

specializzata al problema della materializzazione delle viste

! Per ogni cubo e per ogni livello di aggregazione è possibile

materializzare solo le misure utili per uno specifico carico di

lavoro

Per esempio, sarà molto utile conoscere il valore dell IVA da

versare aggregandola in base al periodo di pagamento (mese o

trimestre), mentre ne sarà richiesto raramente il valore per altri

periodi

" Può richiedere spazio aggiuntivo per la memorizzazione dei dati a

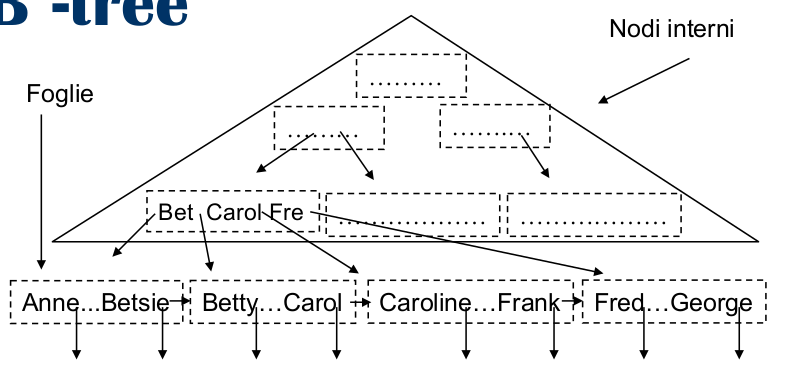
causa delle replicazioni dei campi chiave della fact table

" Determina un risparmio di spazio rispetto alla materializzazione di

viste ogniqualvolta si evita di materializzare una misura

**Indici**

**B+ Tree:**



I B+-Tree forniscono buone prestazioni quando la selettività dei predicati è alta, ma nelle query OLAP la selettività di solito è bassa (es. sesso)

Sono più adatti a interrogazioni semplici.

**Bitmap:**

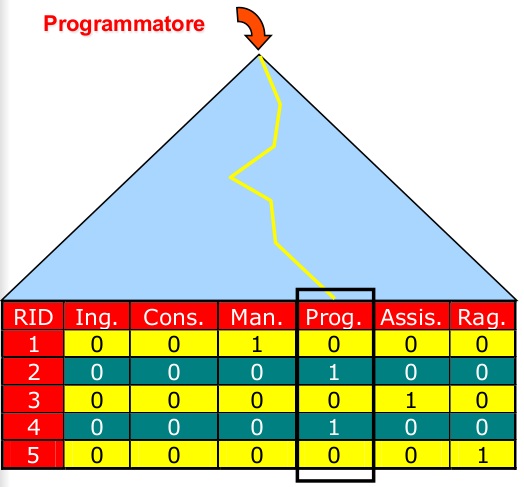
Un indice bitmap su un attributo è composto da una matrice di bit contenente:

" Tante righe quante sono le tuple della relazione

" Tante colonne quanti sono i valori distinti di chiave dell attributo

Sono associati ai B+ Tree, così si possono caricare solo i vettori interessati:

Poi si leggono solo e tuple che hanno TRUE (dal RID)

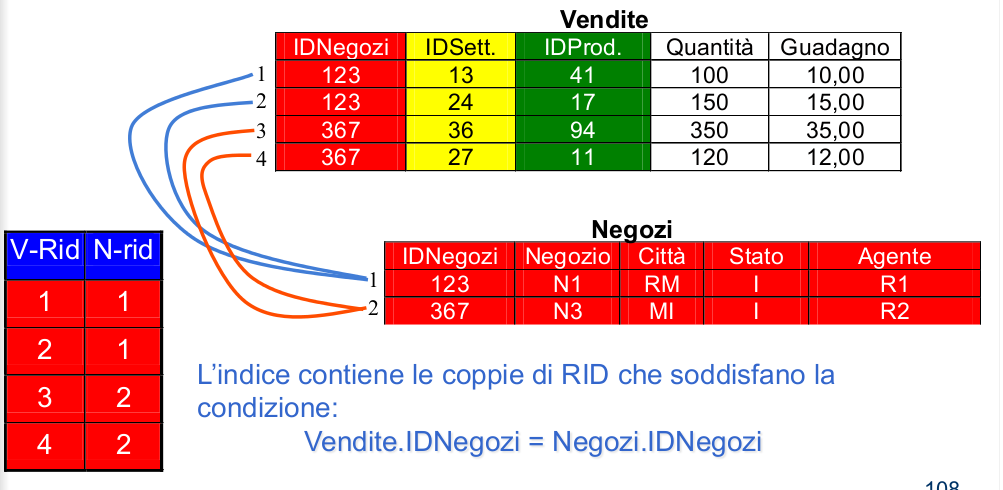


Alcune query possono essere risolte senza accedere ai dati fisici.

Sono adatti per attributi con una ridotta cardinalità, perché ad ogni valore distinto corrisponde un nuovo vettore di bit, e di conseguenza aumenta la sparsità della matrice.

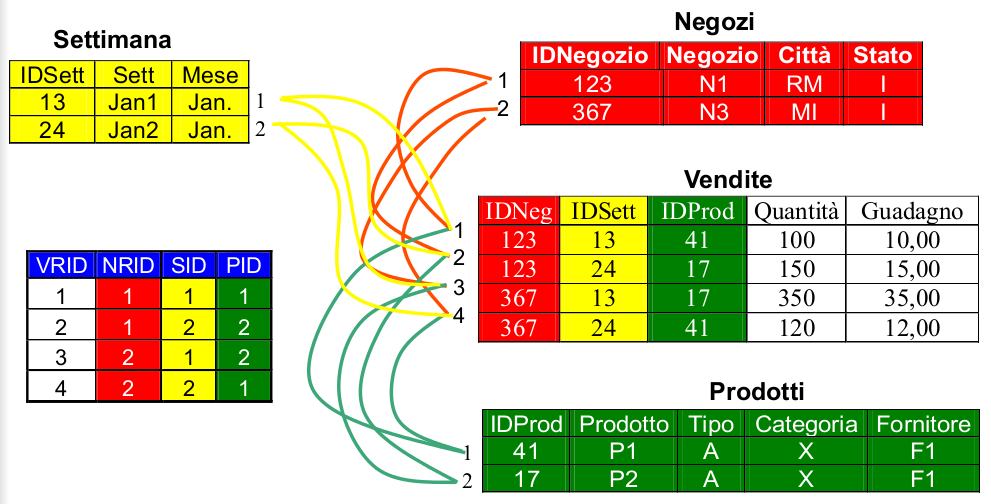
**Indici di Join:**

Calcola in anticipo le tutple che soddisfano un predicato di join (è una tabella con le 2 corrsispondenze di RID delle 2 tuple joinate)



**Indici a stella:**

Come gli indici a join, solo che sono fra più tabelle (cioè fra FT e le sue DT)



Rappresentano esplicitamente l aspetto multidimensionale dei

dati e dipendono fortemente dall ordinamento delle colonne.

! Sono molto efficienti quando utilizzati in interrogazioni che

coinvolgono tutte o le sole colonne iniziali dell indice.

Forniscono prestazioni sub-ottime in caso contrario.

!

Il numero di indici a stella necessari a rispondere efficientemente a

interrogazioni che coinvolgono un insieme arbitrario di

dimensioni è funzione del numero di permutazioni dell insieme

di dimensioni

**Consiglio:**

Fare indici su:

Chiavi importate sulla fact table per aumentare la

velocità di esecuzione dei join (B + -Tree o indici di

join, indici star, bitmapped join index)

! Attributi dimensionali che sono spesso coinvolti nei

criteri di selezione (B + -Tree o bitmap)

**Se** il **DBMS non utilizza statistiche** per definire il

piano di accesso la creazione degli indici deve

essere valutata con **molta attenzione**

**misure calcolate:**

da birth date si calcola la age (diventa una dimensione)

da prezzo unitario e vendite si calcola incasso

Se di una misura prevedi che ti serve spesso sia il totale che la media, fai 2 misure distinte spiegando il perché.

**Possibili soluzioni logiche:**

Per le **dimensioni degeneri:**

1) si possono usare gli attributi interi nella FT

2) per ogni dimensione si fa un surrogato e una DT a parte (un surrogato per ogni dimensione degenere)

3) si usa la junk table, 1 solo surrogato nella FT e la JT contiene tutte le dimensioni degeneri

Per gli **attributi cross-dimensionali:**

si può fare una bridge table in cui non c'è snowflaking, allora la bridge table non ha chiavi esterne, perché non sono chiavi primarie nelle altre DT.

Si può fare il join naturale per recuperare le informazioni corrette.

Oppure si fa snowflaking completo e risparmi più spazio.