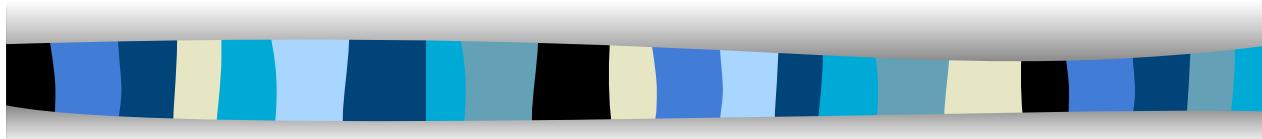


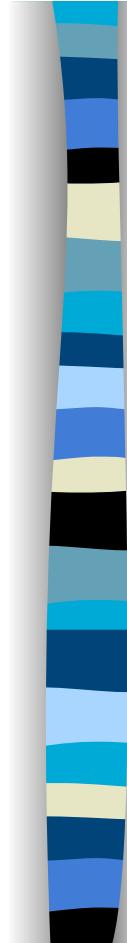
# Progettazione logica



## 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.

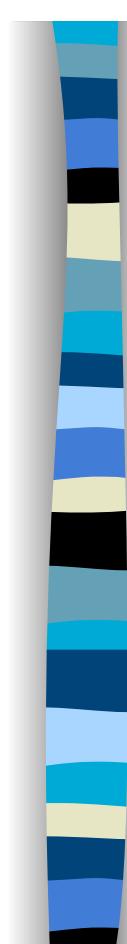




# Sistemi MOLAP

- 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.

3

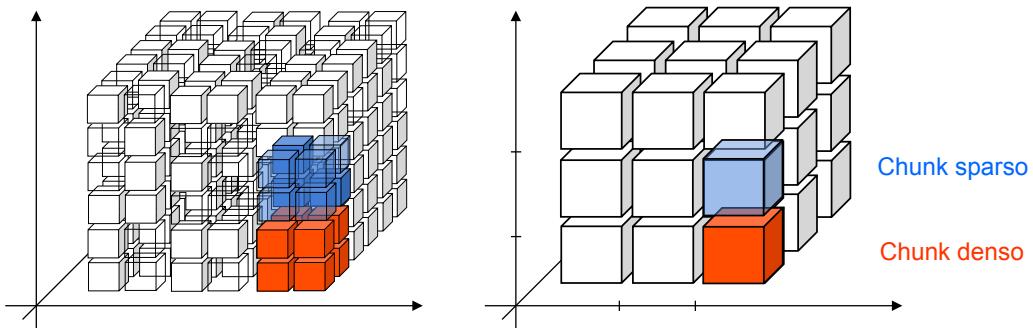


# Sistemi MOLAP e sparsità

- Le tecniche di gestione della sparsità sono basate sui seguenti principi:
  - ✓ **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.

4

# Sistemi MOLAP e sparsità



Una struttura dati comunemente usata per la compressione dei chunk sparsi prevede un indice che riporti il solo offset delle celle che effettivamente contengono informazioni.

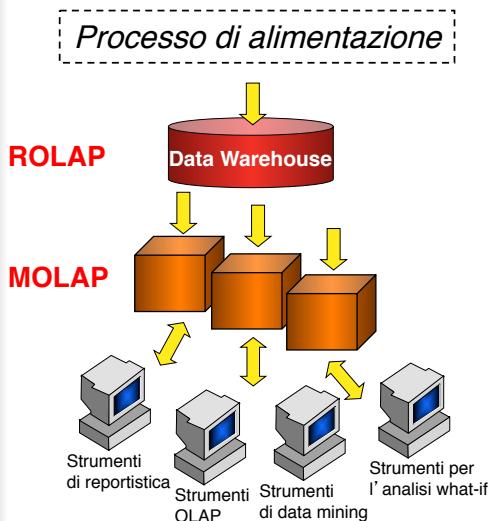
5

## ROLAP, MOLAP e HOLAP

- I sistemi commerciali si differenziano in base al modello logico adottato.
- Sebbene la maggior parte dei sistemi, soprattutto di grandi dimensioni, sia realizzato con soluzioni ROLAP, cominciano ad essere proposte anche alcune soluzioni ibride (Hybrid-OLAP)
- Le soluzioni HOLAP sfruttano le proprietà di entrambi i modelli....

6

# HOLAP



- Il DW ROLAP è ottimale per memorizzare enormi quantità di dati
- I DM MOLAP massimizzano la velocità di accesso ai dati
- I cubi MOLAP possono anche essere creati ‘al volo’ per svolgere specifiche sessioni di analisi (report semi-statici)

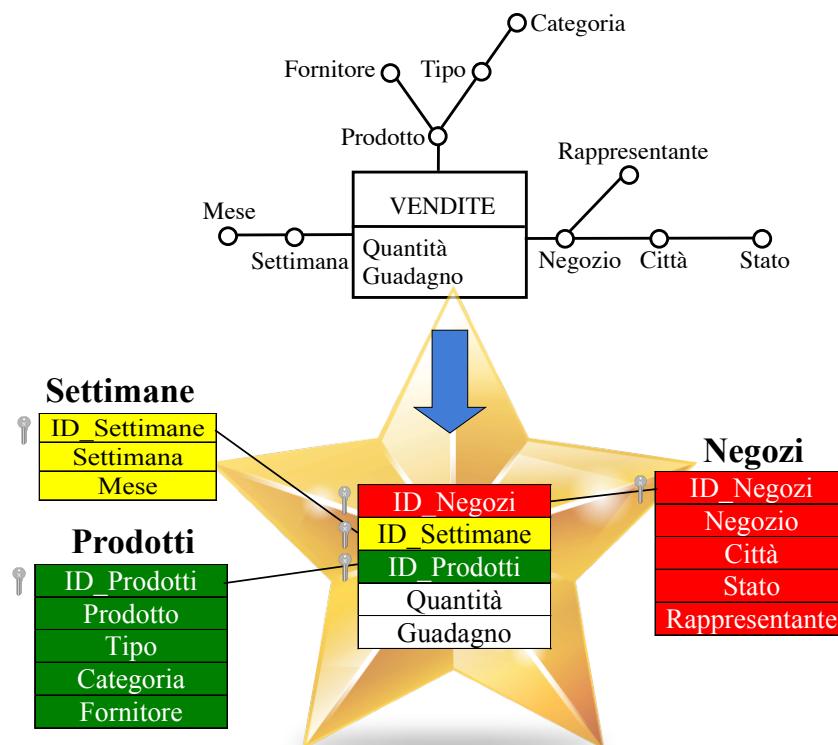
7

## 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  $DT_1, \dots, DT_n$ , 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,  $d_1, \dots, d_n$ ;  $FT$  contiene inoltre un attributo per ogni misura.

8

# Lo schema a stella



9

# Lo schema a stella

ID_Negozi	Negozio	Città	Stato	Rappresentante
1	DiTutto	Roma	I	Rossi
2	DiPiù	Roma	I	Rossi
3	NonSolo	Milano	I	Verdi
4	MaAnche	Milano	I	Verdi

*Dimension Table*

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

*Fact Table*

ID_Sett.	Settimana	Mese
1	1-2019	Gen19
2	2-2019	Gen19
3	3-2019	Gen19
4	4-2019	Gen19

*Dimension Table*

ID_Prodotti	Prodotto	Tipo	Categoria	Fornitore
1	Pecorino	Latticini	Alimentari	Bianchi
2	Emmenthal	Latticini	Alimentari	Bianchi
3	Cola	Bibite	Alimentari	Carli
4	Aranciata	Bibite	Alimentari	Carli

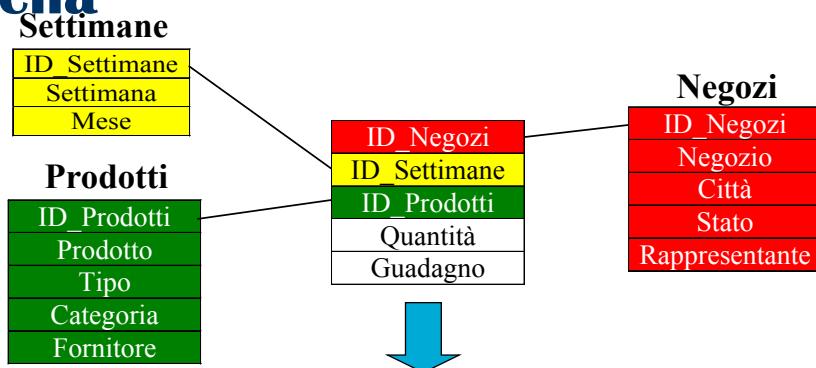
10

# 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
  - ↳ 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

11

## Interrogazioni OLAP su schemi a stella



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

```
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;
```

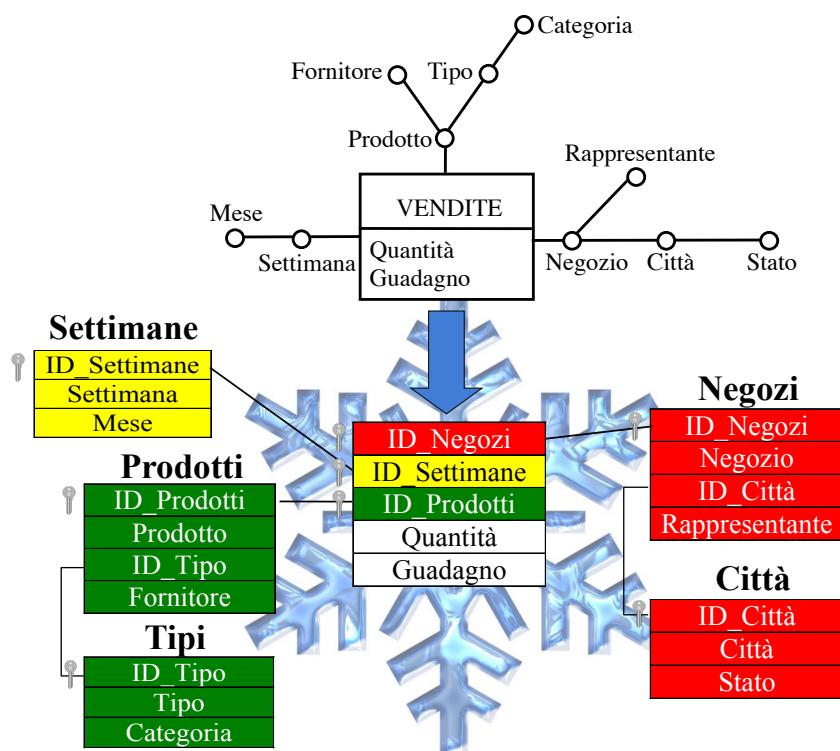
12

# 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 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.

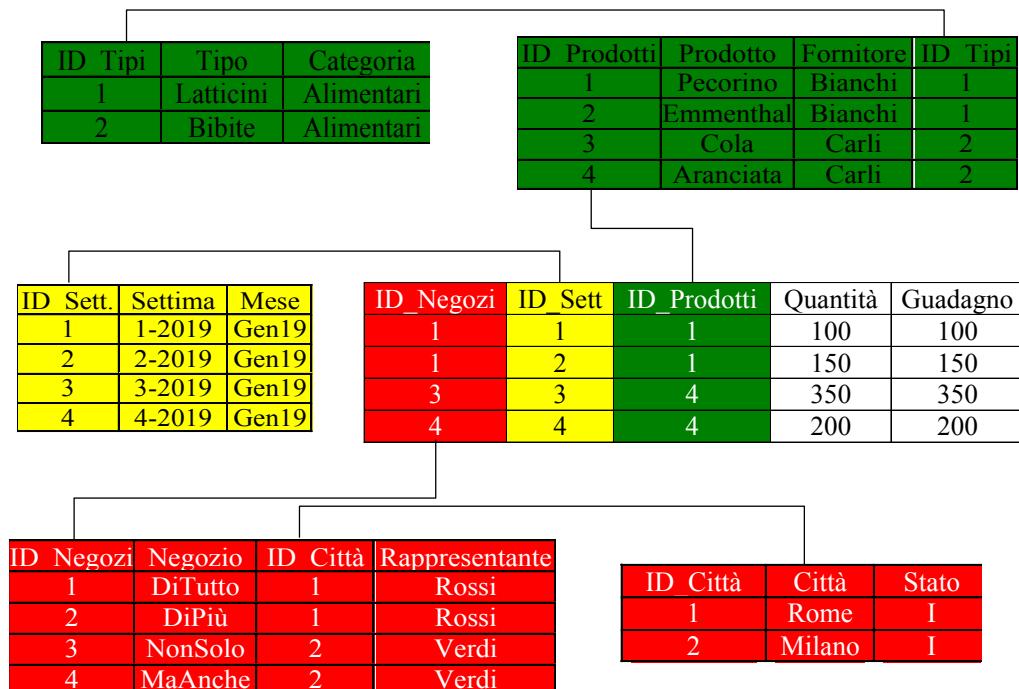
13

# Lo snowflake schema



14

# Lo snowflake schema



15

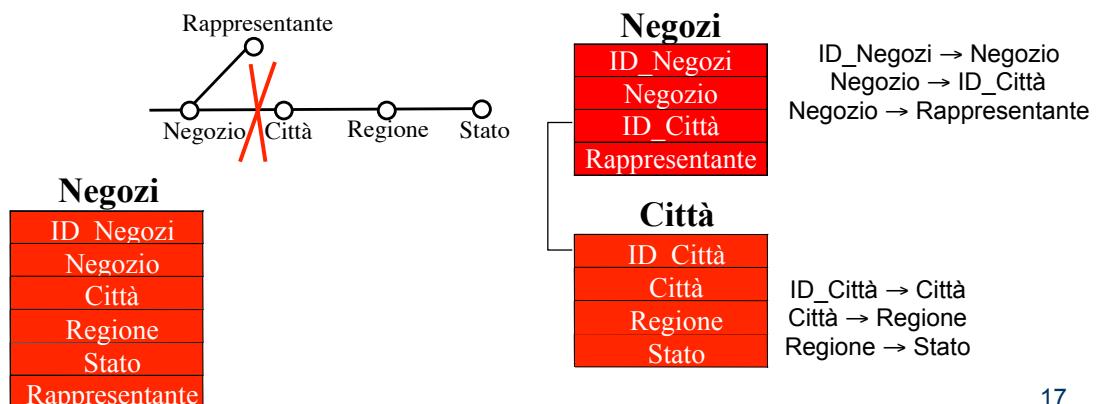
## Lo snowflake schema: considerazioni

- 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

16

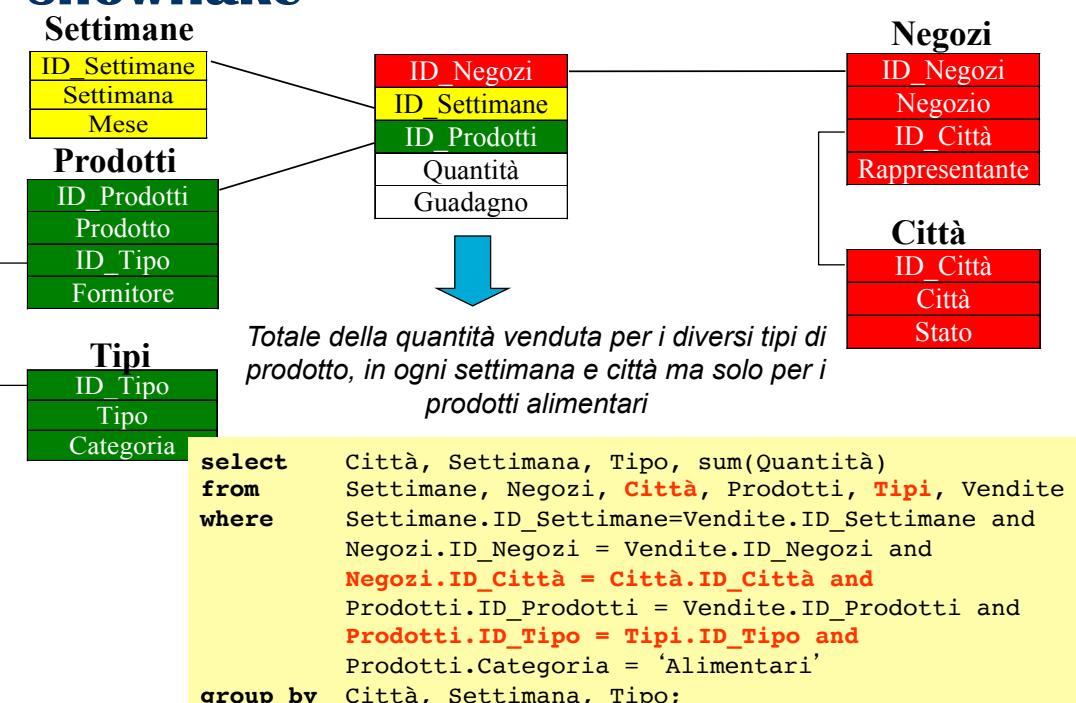
# Normalizzazione con lo snowflake schema

- Le specifiche caratteristiche degli schemi a stella richiedono particolare attenzione affinché nella nuova relazione sia spostato il corretto insieme di attributi
- La presenza di più dipendenze funzionali transitive in cascata fa sì che, affinché la decomposizione sia efficace, tutti gli attributi che dipendono (transitivamente e non) dall'attributo che ha determinato lo snowflaking siano posti nella nuova relazione



17

## Interrogazioni OLAP su schemi snowflake



18

# Le viste

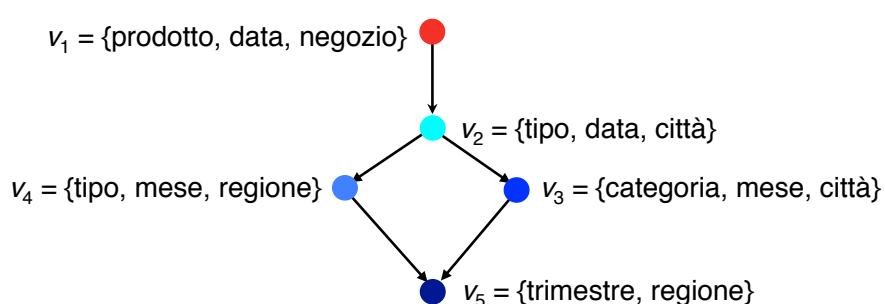
- L' analisi dei dati al massimo livello di dettaglio è spesso troppo complessa e non interessante per gli utenti che richiedono dati di sintesi
- L' aggregazione rappresenta il principale strumento per ottenere informazioni di sintesi
- L' elevato costo computazionale connesso con l' aggregazione induce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

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

19

# Le viste

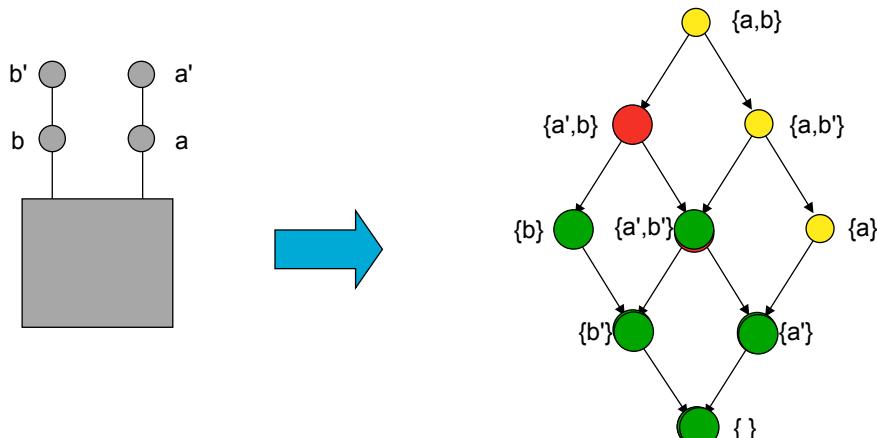
- Le viste possono essere identificate in base al livello (*group-by set*) di aggregazione che le caratterizza



20

# Risolvibilità delle interrogazioni

- Una vista  $v$  sul group-by set  $p$  non serve solo per le interrogazioni con group-by set  $p$  ma anche per tutte quelle che richiedono i dati a group-by set  $p'$  più aggregati di  $p$  ( $p \leq p'$ )



**Reticolo multidimensionale**

21

## Aggregazioni parziali

- Per la corretta gestione dei dati aggregati può essere necessario introdurre nuove misure
  - Misure derivate:** ottenute applicando operatori matematici a due o più valori appartenenti alla stessa tupla

Tipo	Prodotto	Quantità	Prezzo	Incasso
T1	P1	5	1,00	5,00
T1	P2	7	1,50	10,50
T2	P3	9	0,80	7,20

Sum      AVG

Tipo	Quantità	Prezzo	Incasso
T1	12	1,25	15,00
T2	9	0,80	7,20

22,70      ?      22,20

La soluzione corretta è sempre quella che si ottiene aggregando i dati direttamente dalla vista primaria

22

# Aggregazioni parziali

- ✓ **Misure di supporto:** sono necessarie in presenza di operatori di aggregazione non distributivi

Data	Livello di inventario
1/1/1999	100
10/2/1999	200
31/4/1999	60
5/6/1999	85
18/7/1999	125
31/12/1999	110

1999

113,33

Trimestre	Livello di inventario	Count	Livello di inventario
4/1999	120	3	360
8/1999	105	2	210
12/1999	110	1	110
1999			111,66
			113,33



La soluzione corretta è sempre quella che si ottiene aggregando i dati direttamente dalla vista primaria

23

# Classificazione degli operatori di aggregazione

- I problemi visti in precedenza derivano dalla natura degli operatori di aggregazione che possono essere così classificati:
  - ✓ **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)

24

# Schemi relazionali e viste

- La soluzione più semplice consiste nell' utilizzare lo schema a stella memorizzando tutti i dati in una sola fact table
  - ✓ La dimensione dell' unica fact table cresce considerevolmente a discapito delle prestazioni
  - ✓ Le dimension table contengono tuple relative a diversi livelli di aggregazione. Il valore NULL viene utilizzato per identificare l' origine delle tuple

25

# Schemi relazionali e viste

Sono relative al group-by set:  
{Negozi, Settimane, **Prodotti**}

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
2	1	5	3600	3600
1	3	6	2400	2400
1	1	7	1000	1000

ID_Prodotti	Prodotto	Tipo	Categoria	Fornitori
1	Pecorino	Latticini	Alimentari	Bianchi
2	Emmenthal	Latticini	Alimentari	Bianchi
3	Cola	Bibite	Alimentari	Carli
4	Aranciata	Bibite	Alimentari	Carli
5	-	Latticini	Alimentari	Bianchi
6	-	Bibite	Alimentari	Carli
7	-	-	-	Bianchi
8	-	-	-	Carli

26

# Schemi relazionali e viste

Sono relative al group-by set:  
{Negozi, Settimane, **Tipo**}

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
2	1	5	3600	3600
1	3	6	2400	2400
1	1	7	1000	1000



ID_Prodotti	Prodotto	Tipo	Categoria	Fornitore
1	Pecorino	Latticini	Alimentari	Bianchi
2	Emmenthal	Latticini	Alimentari	Bianchi
3	Cola	Bibite	Alimentari	Carli
4	Aranciata	Bibite	Alimentari	Carli
5	-	Latticini	Alimentari	Bianchi
6	-	Bibite	Alimentari	Carli
7	-	-	-	Bianchi
8	-	-	-	Carli

27

# Schemi relazionali e viste

È relativa al group-by set:  
{Negozi, Settimane, **Fornitore**}

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
2	1	5	3600	3600
1	3	6	2400	2400
1	1	7	1000	1000

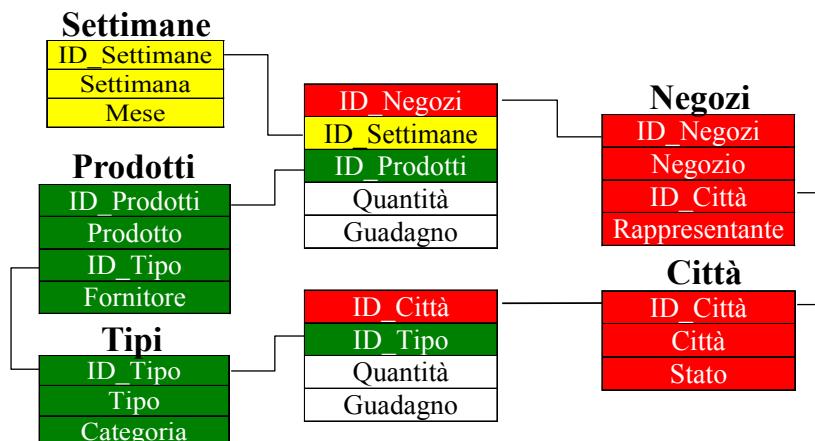


ID_Prodotti	Prodotto	Tipo	Categoria	Fornitore
1	Pecorino	Latticini	Alimentari	Bianchi
2	Emmenthal	Latticini	Alimentari	Bianchi
3	Cola	Bibite	Alimentari	Carli
4	Aranciata	Bibite	Alimentari	Carli
5	-	Latticini	Alimentari	Bianchi
6	-	Bibite	Alimentari	Carli
7	-	-	-	Bianchi
8	-	-	-	Carli

28

# Schemi relazionali e viste

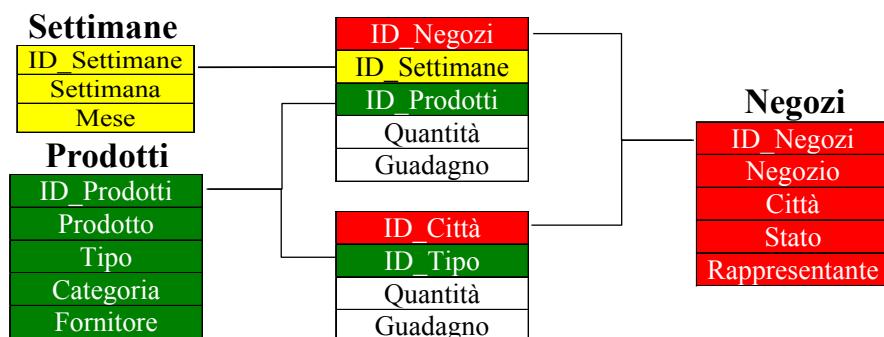
- Adottando lo snowflake schema è possibile memorizzare in fact table separate dati appartenenti a diversi group-by set
  - ✓ Lo snowflaking deve essere applicato in corrispondenza dei livelli di aggregazione a cui sono presenti viste



29

# Schemi relazionali e viste

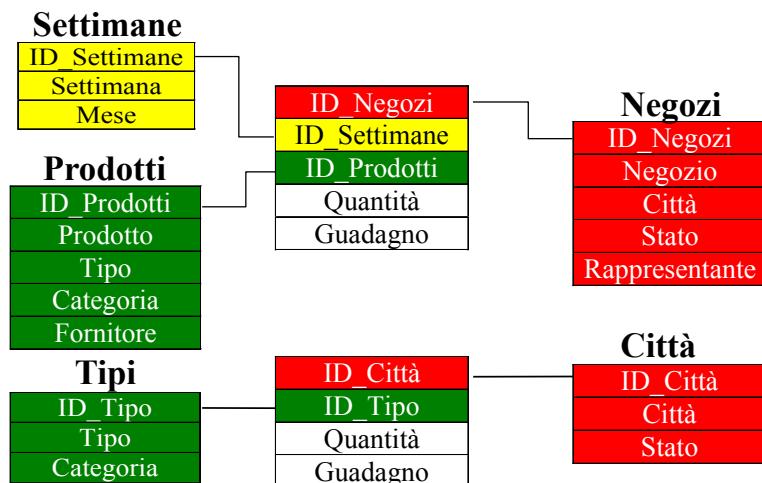
- Una soluzione intermedia rispetto alle due presentate prevede di memorizzare in fact table separate dati relativi a group-by set diversi senza però ricorrere alla normalizzazione delle dimension table (*constellation schema*)
  - ✓ L'accesso alle fact table è ottimizzato, quello alle dimension table no
  - ✓ La dimensione delle fact table è di molto superiore a quella delle dimension table e conseguentemente la loro ottimizzazione gioca un ruolo fondamentale



30

# Schemi relazionali e viste

- 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



31

# 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

32

# Scenari temporali

- Il modello multidimensionale assume che gli eventi che istanziano un fatto siano **dinamici**, e che i valori degli attributi che popolano le gerarchie siano **statici**
- Questa visione non è realistica poiché anche i valori presenti nelle gerarchie variano nel tempo dando vita alle gerarchie dinamiche (**slowly-changing dimension**)
- L'adozione di gerarchie dinamiche implica un sovraccosto in termini di spazio e può comportare una forte riduzione delle prestazioni

33

# Scenari temporali

.... Sono possibili diverse soluzioni

- Oggi per ieri (*attualizzazione*)
  - ✓ I dati vengono interpretati in base all'attuale configurazione della gerarchia
  - ✓ Implementabile sullo schema a stella
- Oggi o ieri (*verità storica*)
  - ✓ I dati vengono interpretati in base alla configurazione valida al momento in cui sono stati registrati
  - ✓ Implementabile sullo schema a stella
- Ieri per oggi (*retrodatazione*)
  - ✓ I dati vengono interpretati in base alla configurazione della gerarchia valida in un particolare istante
  - ✓ Richiede la storizziazione dei dati

34

# Un esempio

Situazione al 1/1/2011

negozio	responsabile
DiTutto	Rossi
NonSoloPile	Bianchi
NonSoloPane	Bianchi

Situazione al 1/11/2011

negozio	responsabile
DiTutto	Rossi
NonSoloPile	Bianchi
PaneEPizza	Rossi

Situazione al 1/7/2011

negozio	responsabile
DiTutto	Rossi
NonSoloPile	Bianchi
NonSoloPane	Rossi

Situazione al 1/1/2012

negozio	responsabile
DiTutto	Bianchi
NonSoloPile	Bianchi
PaneEPizza	Rossi
DiTuttoDiPiù	Rossi

35

# Un esempio

negozio	responsabile
DiTutto	Rossi
NonSoloPile	Bianchi
NonSoloPane	Bianchi

negozio	data	incasso
DiTutto	20/6/2011	10
NonSoloPile	20/6/2011	20
NonSoloPile	30/6/2011	15
NonSoloPile	2/7/2011	10
DiTutto	2/7/2011	30
NonSoloPile	10/7/2011	15
NonSoloPile	12/7/2011	10
NonSoloPile	15/7/2011	20

1/7/2011

tempo

36

# Un esempio

negozi	responsabile
DiTutto	Rossi
NonSoloPile	Bianchi
NonSoloPane	Bianchi

negozi	data	incasso
DiTutto	20/6/2011	10
NonSoloPane	20/6/2011	20
NonSoloPane	30/6/2011	15
NonSoloPile	2/7/2011	10
DiTutto	2/7/2011	30
NonSoloPile	10/7/2011	15
NonSoloPile	12/7/2011	10
NonSoloPile	15/7/2011	20

- Incassi totali per responsabile (16/7/2011)

✓ attualizzazione

responsabile	incasso
Rossi	100
Bianchi	30

# Un esempio

negozi	responsabile
DiTutto	Rossi
NonSoloPile	Bianchi
NonSoloPane	Bianchi

negozi	data	incasso
DiTutto	20/6/2011	10
NonSoloPane	20/6/2011	20
NonSoloPane	30/6/2011	15
NonSoloPile	2/7/2011	10
DiTutto	2/7/2011	30
NonSoloPile	10/7/2011	15
NonSoloPile	12/7/2011	10
NonSoloPile	15/7/2011	20

- Incassi totali per responsabile (16/7/2011)

✓ attualizzazione

✓ verità storica

responsabile	incasso
Rossi	100
Bianchi	30

responsabile	incasso
Rossi	65
Bianchi	65

# Un esempio

negozi	responsabile
DiTutto	Rossi
NonSoloPile	Bianchi
NonSoloPane	Bianchi

negozi	responsabile
DiTutto	Rossi
NonSoloPile	Bianchi
NonSoloPane	Rossi

negozi	data	incasso
DiTutto	20/6/2011	10
NonSoloPane	20/6/2011	20
NonSoloPane	30/6/2011	15
NonSoloPane	2/7/2011	10
DiTutto	2/7/2011	30
NonSoloPane	10/7/2011	15
NonSoloPile	12/7/2011	10
NonSoloPile	15/7/2011	20

- Incassi totali per responsabile (16/7/2011)

✓ attualizzazione

✓ verità storica

✓ retrodatazione al 25/6/2016

responsabile	incasso
Rossi	100
Bianchi	30

responsabile	incasso
Rossi	65
Bianchi	65

responsabile	incasso
Rossi	40
Bianchi	90

39

## Gerarchie dinamiche: tipo I

- Supportano solo lo scenario oggi per ieri, pertanto tutti gli eventi, anche quelli passati, vengono interpretati in base all'attuale configurazione delle gerarchie senza tenere traccia del passato
- Questa soluzione è realizzabile sullo schema a stella sovrascrivendo il vecchio valore con quello nuovo ogni volta che si verifica un cambiamento

40

## Gerarchie dinamiche: tipo I

Situazione al 1/1/2011

chiaveN	negozi	responsabile	...
1	DiTutto	Rossi	...
2	NonSoloPile	Bianchi	...
3	NonSoloPane	Bianchi	...

Situazione al 1/7/2011

chiaveN	negozi	responsabile	...
1	DiTutto	Rossi	...
2	NonSoloPile	Bianchi	...
3	NonSoloPane	Rossi	...

Tutte le vendite di NonSoloPane vengono attribuite a Rossi anche se erano state effettuate durante la gestione di Bianchi

41

## Gerarchie dinamiche: tipo II

- Supportano solo lo scenario oggi o ieri, e consentono di registrare la verità storica
- Gli eventi memorizzati nella fact table vengono associati ai dati dimensionali che erano validi quando si è verificato l' evento
- Questa soluzione è realizzabile sullo schema a stella: ogni modifica a una gerarchia comporta l' inserimento di un nuovo record che codifichi le nuove caratteristiche nella dimension table corrispondente
- È possibile adottare strategie diverse per attributi appartenenti alla stessa gerarchia

42

## Gerarchie dinamiche: tipo II

Situazione al 1/1/2011

chiaveN	negozi	responsabile	...
1	DiTutto	Rossi	...
2	NonSoloPile	Bianchi	...
3	NonSoloPane	Bianchi	...

Situazione al 1/7/2011

chiaveN	negozi	responsabile	...
1	DiTutto	Rossi	...
2	NonSoloPile	Bianchi	...
3	NonSoloPane	Bianchi	...
4	NonSoloPane	Rossi	...

Dopo l' 1/7 i record della fact table relativi a NonSoloPane importeranno il valore di chiaveN = 4

N.B. Solo le selezioni su campi che hanno subito modifiche sono sensibili alle modifiche stesse!!

43

## Gerarchie dinamiche: tipo III

- Supportano tutti gli scenari temporali. La loro adozione richiede la storicizzazione dell' attributo e non può pertanto essere basata sul classico schema a stella
- 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*)
- In uno schema così modificato la dinamicità viene gestita aggiungendo, per ogni modifica, un nuovo record nella dimension table e aggiornando di conseguenza i valori dei time-stamp e dell' attributo master

44

# Gerarchie dinamiche: tipo III

Situazione al 1/1/2011

chiaveN	negozi	responsabile	...	da	a	Master
1	DiTutto	Rossi	...	1/1/2011	—	1
2	NonSoloPile	Bianchi	...	1/1/2011	—	2
3	NonSoloPane	Bianchi	...	1/1/2011	—	3

Situazione al 1/1/2012

chiaveN	negozi	responsabile	...	da	a	Master
1	DiTutto	Rossi	...	1/1/2011	31/12/2011	1
2	NonSoloPile	Bianchi	...	1/1/2011	—	2
3	NonSoloPane	Bianchi	...	1/1/2011	30/6/2011	3
4	NonSoloPane	Rossi	...	1/7/2011	31/10/2011	3
5	PaneEPizza	Rossi	...	1/11/2011	—	3
6	DiTuttoDiPiù	Rossi	...	1/1/2012	—	6
7	DiTutto	Bianchi	...	1/1/2012	—	1

45

# Gerarchie dinamiche: tipo III

- Avendo a disposizione lo schema descritto in precedenza è facile realizzare i differenti scenari temporali:
  - ✓ **Oggi per ieri:** si identificano dapprima le tuple della dimension table attualmente valide (in base ai time-stamp) e per ciascuna si individuano eventuali altre tuple da cui esse hanno avuto origine
  - ✓ **Ieri per oggi:** fissata una particolare data si individuano le tuple valide in quel particolare momento, quindi si procede come nel caso precedente
  - ✓ **Oggi o ieri:** non richiede l' analisi delle marche temporali poiché l' aggiornamento delle tuple nelle dimension table avviene come per le gerarchie di tipo 2

46

# Gerarchie dinamiche: tipo III

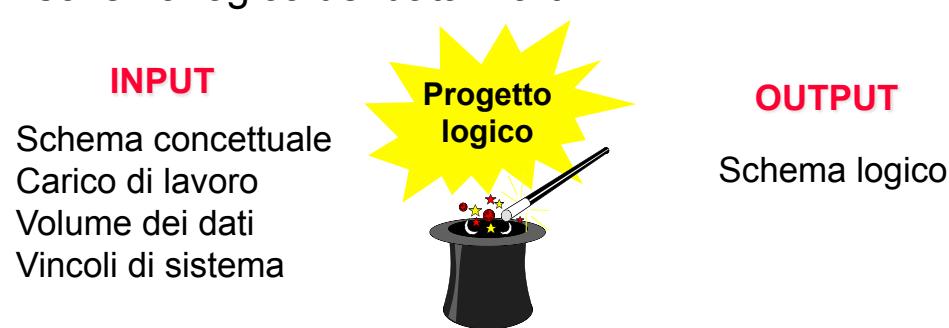
- Utilizzando la soluzione ieri per oggi, l' interrogazione SQL che richiede: “*La quantità totale venduta dai diversi responsabili se si considera l'assegnamento ai negozi vero il 1/10/2011*” è la seguente

```
select      N1.Responsabile, sum(Quantità)
from        Negozi N1, Negozi N2, Vendite
where       N1.Da <= 1/10/11
           AND N1.A > 1/10/11
           AND N1.Master=N2.Master
           AND Vendite.ChiaveN=N2.ChiaveN
group by    N1.Responsabile;
```

47

# Progettazione logica

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



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

48

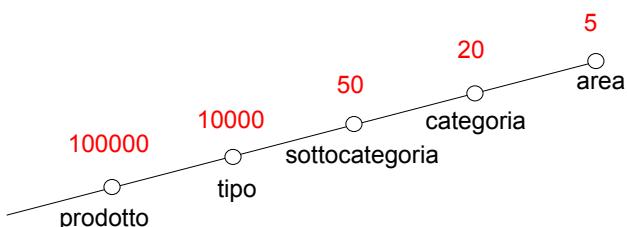
# Progettazione logica

- 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)

49

## Star VS Snowflake

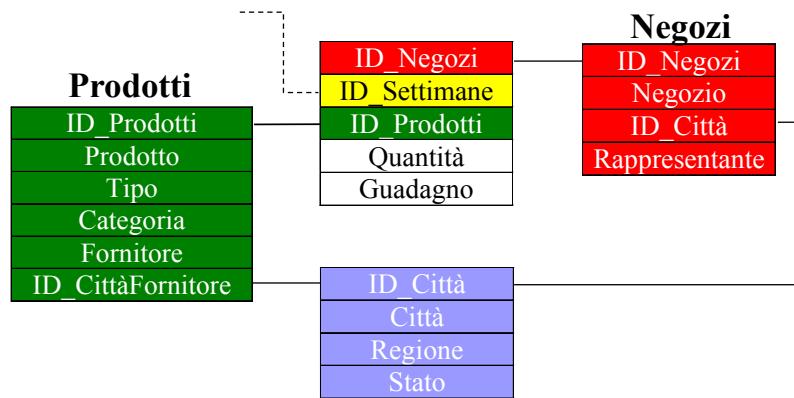
- Esistono pareri contrastanti sull' utilità dello snowflaking:
  - ✓ Contrasta con la filosofia del data warehousing
  - ✓ Rappresenta un inutile “abbellimento” dello schema
- Può essere utile
  1. Quando il rapporto tra le cardinalità della dimension table primaria e secondaria è elevato, poiché determina un forte risparmio di spazio



50

# Star VS Snowflake

- Può essere utile
  - 2. Quando una porzione di una gerarchia è comune a più dimensioni

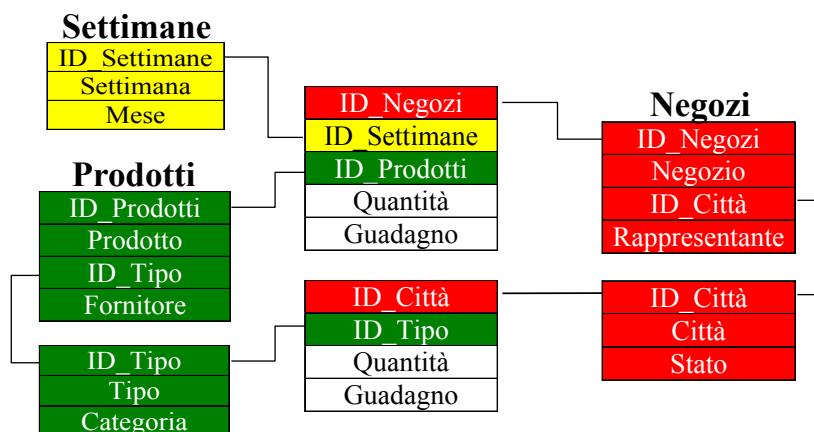


La dimension table secondaria è riutilizzata per più gerarchie

51

# Star VS Snowflake

- Può essere utile
  - 3. In presenza di viste aggregate

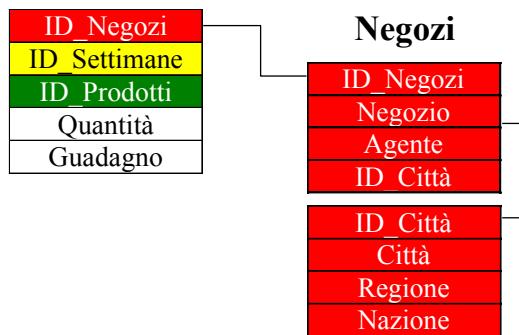


La dimension table secondaria della vista primaria coincide con la dimension table primaria della vista secondaria

52

# Star VS Snowflake

- Può essere utile
  - 4. Quando una parte della gerarchia è soggetta a frequenti aggiornamenti



L'agente del negozio varia frequentemente, mentre la regione e nazione della città del negozio sono statici

53

# 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.*

- In aggiunta a questa semplice regola, la corretta traduzione di uno schema di fatto richiede una trattazione approfondita dei costrutti avanzati del DFM

54

# Attributi descrittivi

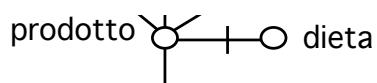
- Contiene informazioni non utilizzabili per effettuare aggregazioni ma che si ritiene comunque utile mantenere
  - ✓ Se collegato a un attributo dimensionale, va incluso nella dimension table che contiene l' attributo
  - ✓ Se collegato direttamente al fatto deve essere incluso nella fact table



55

# Archi opzionali

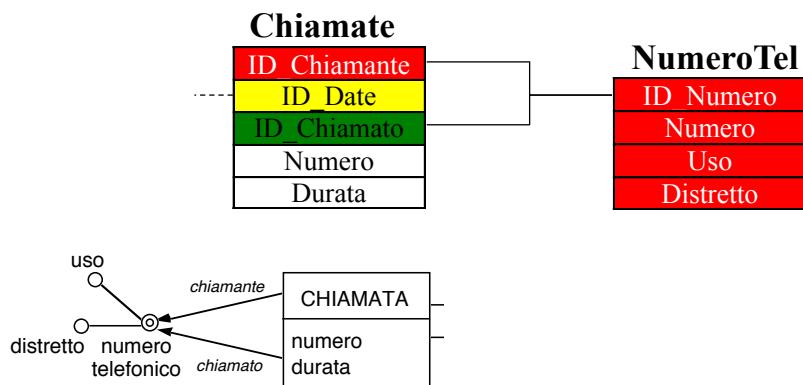
- Alcune porzioni delle gerarchie possono essere opzionali
  - ✓ Nella dimension table, nelle righe per cui non è definito un valore viene inserito un valore fittizio (NULL oppure NON APPLICABILE)
- A causa dei vincoli di integrità, l' opzionalità di un' intera gerarchia NON può essere gestita introducendo un valore nullo nella chiave esterna della fact table, occorre invece inserire un' intera tupla fittizia nella dimension table



56

# Gerarchie condivise

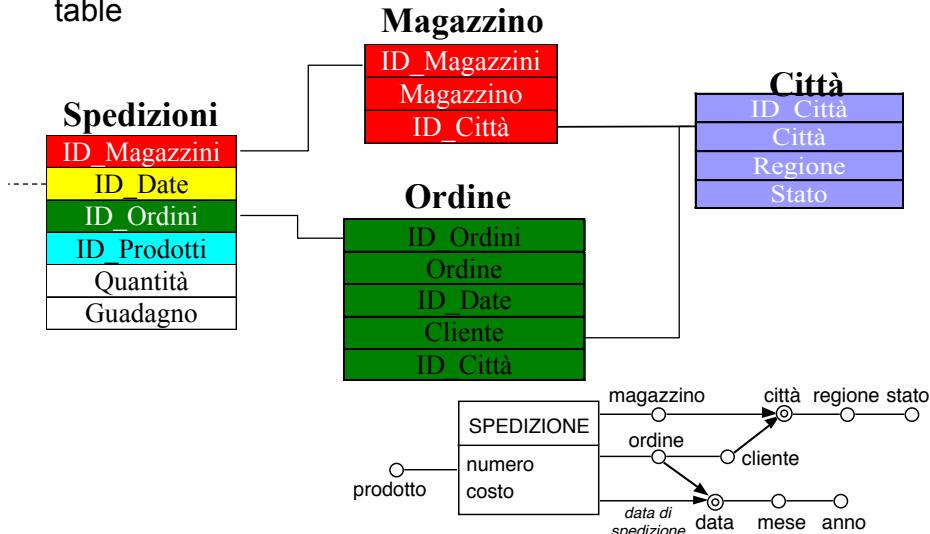
- Se una gerarchia si presenta più volte nello stesso fatto (o in due fatti diversi) non conviene introdurre copie ridondanti delle relative dimension table
- Se le due gerarchie contengono esattamente gli stessi attributi sarà sufficiente importare due volte la chiave della medesima dimension table



57

# Gerarchie condivise

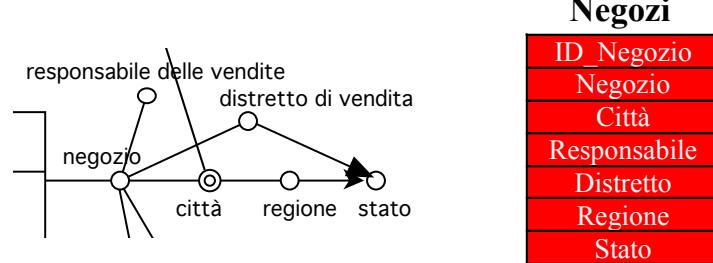
- Se le due gerarchie condividono solo una parte degli attributi è necessario decidere se:
  - Introdurre ulteriore ridondanza nello schema duplicando le gerarchie e replicando i campi comuni
  - Eseguire uno snowflake sul primo attributo condiviso introducendo una terza tabella comune a entrambe le dimension table



58

# Convergenza

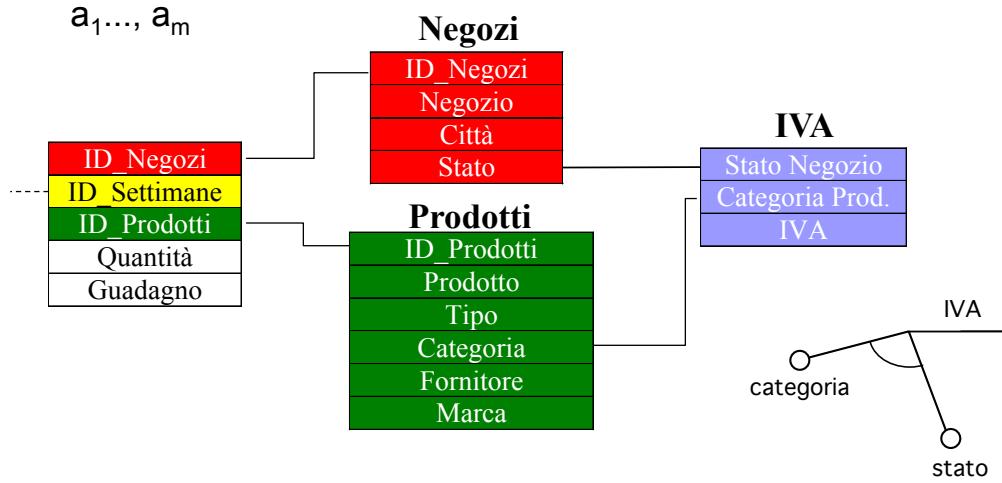
- Gli attributi di convergenza si includono nella stessa dimension table dei loro attributi padri, senza particolari accorgimenti



59

# Attributi cross-dimensional

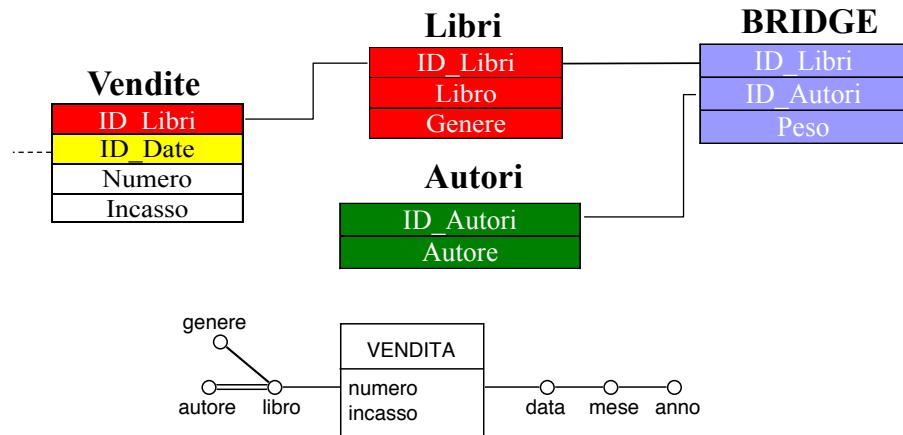
- Dal punto di vista concettuale, un attributo cross-dimensionale b definisce un' associazione molti-a-molti tra due o più attributi dimensionali  $a_1, \dots, a_m$
- La sua traduzione a livello logico richiede l' inserimento di una nuova tabella che includa b e abbia come chiave gli attributi  $a_1, \dots, a_m$



60

# Archi multipli

- La soluzione progettuale più ovvia è quella di inserire una tabella aggiuntiva (*bridge table*) che modelli l' arco multiplo:
  - ✓ La chiave della bridge table è composta dalla combinazione degli attributi collegati all' arco multiplo
  - ✓ Un eventuale attributo *peso* può permettere di attribuire importanza diversa alle tuple partecipanti



61

# Archi multipli

LIBRO

<u>chiaveL</u>	libro	genere
1	Il DFM	tecnico
2	Mi Sembra Logico	tecnico
3	La Giusta Misura	attualità
4	Un Fatto Come e Perchè	attualità
5	La Quarta Dimensione	fantascienza

AUTORE

<u>chiaveA</u>	autore
1	Matteo Golfarelli
2	Stefano Rizzi

BRIDGE\_AUTORE

<u>chiavel</u>	<u>chiaveA</u>	<u>peso</u>
1	1	0,5
1	2	0,5
2	1	1,0
3	2	1,0
4	1	0,5
4	2	0,5
5	1	1,0

VENDITE

<u>chiavel</u>	<u>chiaveD</u>	numero	incasso
1	1	3	150
2	1	5	250
3	1	10	300
4	1	4	80
5	1	8	400

62

# Archi multipli

- Possono essere necessari sino a 3 join per recuperare tutte le informazioni contenute nella gerarchia
- La soluzione con bridge table rende possibili due tipi di interrogazioni:
  - ✓ **Interrogazioni pesate:** considerano il peso dell' arco multiplo e forniscono pertanto l' effettivo totale

## *Incasso di ciascun autore*

```
SELECT AUTORI.Autore, sum(VENDITE.Incasso * BRIDGE.Peso)
FROM AUTORI, BRIDGE, LIBRI, VENDITE
WHERE AUTORI.ID_Autori = BRIDGE.ID_Autori
AND BRIDGE.ID_Libri = LIBRI.ID_Libri
AND LIBRI.ID_Libri = VENDITE.ID_Libri
GROUP BY AUTORI.Autore
```

63

# Archi multipli

- Possono essere necessari sino a 3 join per recuperare tutte le informazioni contenute nella gerarchia
- La soluzione con bridge table rende possibili due tipi di interrogazioni:
  - ✓ **Interrogazioni pesate:** considerano il peso dell' arco multiplo e forniscono pertanto l' effettivo totale
  - ✓ **Interrogazioni di impatto:** non considerano il peso e perciò forniscono valori più elevati

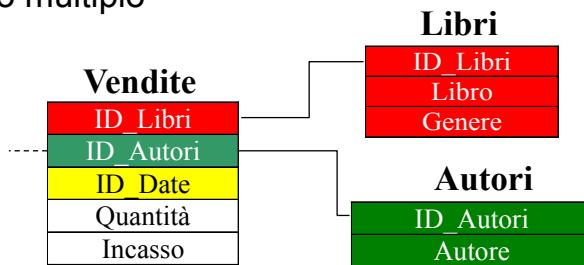
## *Copie vendute per ogni autore*

```
SELECT AUTORI.Autore, sum(VENDITE.Quantità)
FROM AUTORI, BRIDGE, LIBRI, VENDITE
WHERE AUTORI.ID_Autori = BRIDGE.ID_Autori
AND BRIDGE.ID_Libri = LIBRI.ID_Libri
AND LIBRI.ID_Libri = VENDITE.ID_Libri
GROUP BY AUTORI.Autore
```

64

# Archi multipli

- Nel caso si voglia continuare a utilizzare lo schema a stella è necessario rendere più fine la granularità del fatto modellando così l' arco multiplo direttamente nella fact table (*push-down*)
- Questa soluzione richiede l' aggiunta alla fact table di una nuova dimensione corrispondente all' attributo terminale dell' arco multiplo



65

# Archi multipli: comparazione

- Il potere informativo delle due soluzioni è identico
- 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

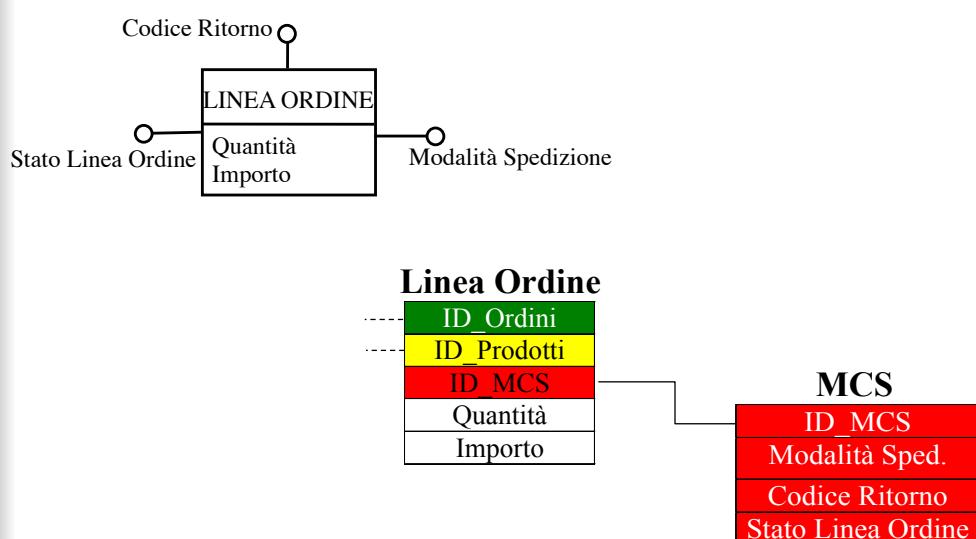
66

# Dimensioni degeneri

- Questo termine indica una dimensione la cui gerarchia contiene un solo attributo
- 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

67

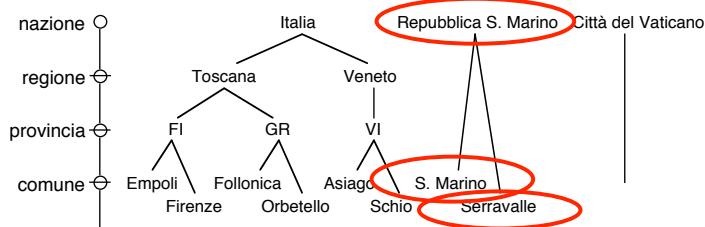
# Dimensioni degeneri



68

# 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
- Il problema è più complesso rispetto al caso degli attributi opzionali poiché la mancanza di un valore di un attributo non implica la mancanza dei successivi nella gerarchia di aggregazione



- È necessario mantenere la consistenza rispetto all' operatore di roll-up
- Sono possibili più soluzioni che si differenziano per il tipo di segnaposto inserito

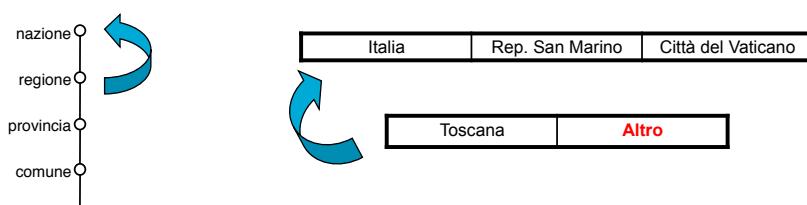
69

# Gerarchie incomplete

- Bilanciamento per esclusione:** in tutte le tuple viene inserito un segnaposto generico (es. "altro")

nazione	Italia	Rep. San Marino	Rep. San Marino	Città del Vaticano
regione	Toscana	Altro	Altro	Altro
provincia	Firenze	Altro	Altro	Altro
comune	Empoli	San Marino	Serravalle	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



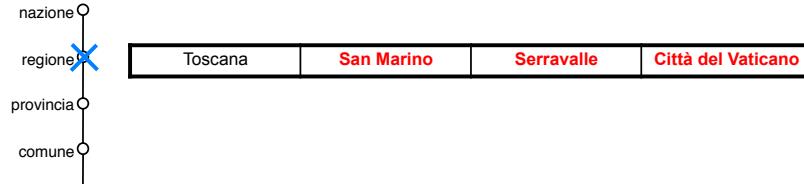
70

# Gerarchie incomplete

- **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



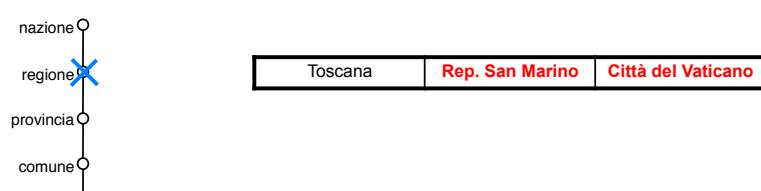
71

# Gerarchie incomplete

- **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



72

# 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

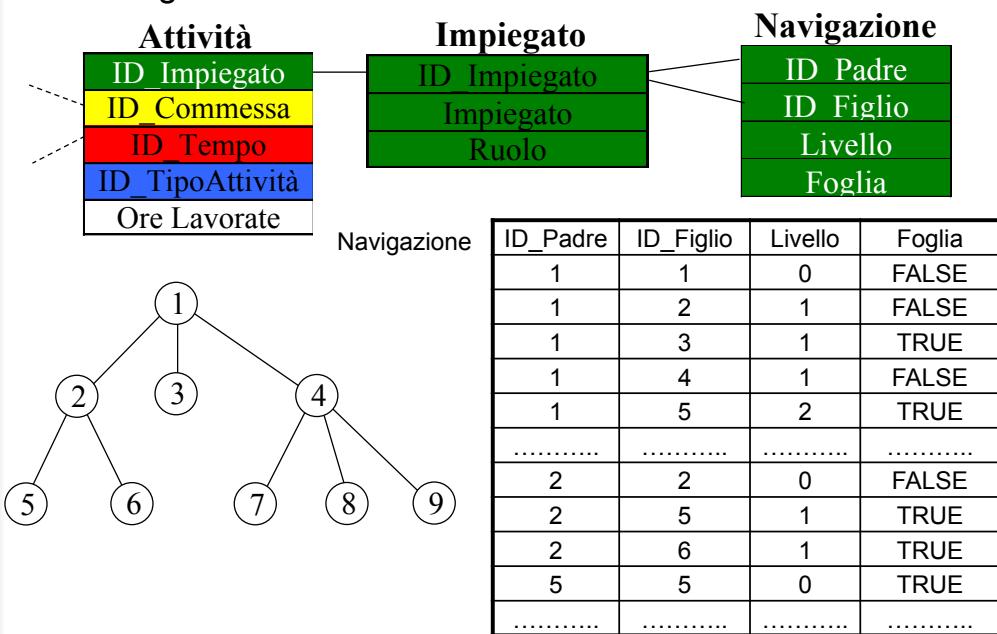


- Non sempre è gestibile in modo ottimale con DBMS commerciali
- SQL non è un linguaggio ricorsivo

73

# Gerarchie ricorsive

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



74

# Gerarchie ricorsive

- 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
- Per **descendere** la gerarchia:

*Il totale delle ore lavorate dal gruppo di cui è responsabile il sig. Rossi*

```
SELECT sum(ore lavorate)
FROM ATTIVITA A, IMPIEGATO I, NAVIGAZIONE N
WHERE I.Nome= 'Rossi' AND I.ID_Impiegato=N.ID_Padre
AND N.ID_Figlio = A.ID_Impiegato;
```

*Il totale delle ore lavorate dai subordinati diretti dal sig. Rossi*

```
SELECT sum(ore lavorate)
FROM ATTIVITA A, IMPIEGATO I, NAVIGAZIONE N
WHERE I.Nome= 'Rossi' AND I.ID_Impiegato=N.ID_Padre
AND N.ID_Figlio = A.ID_Impiegato AND N.Livello=1;
```

75

# Gerarchie ricorsive

- 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.
- Per **risalire** la gerarchia:

*Il totale delle ore lavorate dai responsabili del sig. Rossi*

```
SELECT sum(ore lavorate)
FROM ATTIVITA A, IMPIEGATO I, NAVIGAZIONE N
WHERE I.Nome= 'Rossi' AND I.ID_Impiegato=N.ID_Figlio
AND N.ID_Padre = A.ID_Impiegato;
```

- Escludendo dai join la tabella di navigazione si continua ad avere uno schema a stella

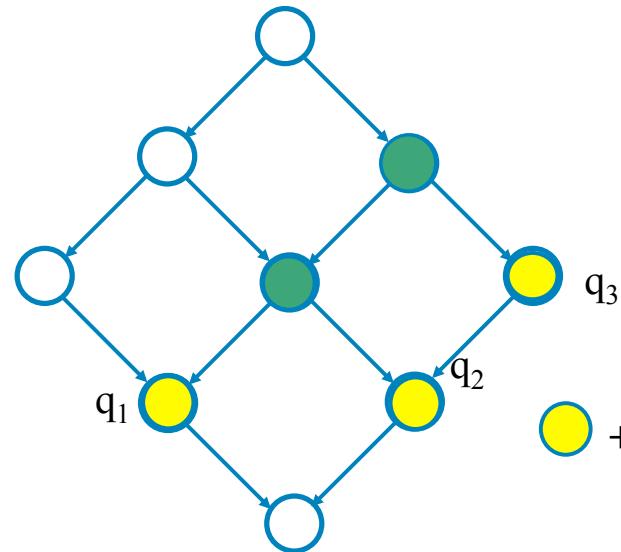
76

# 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

77

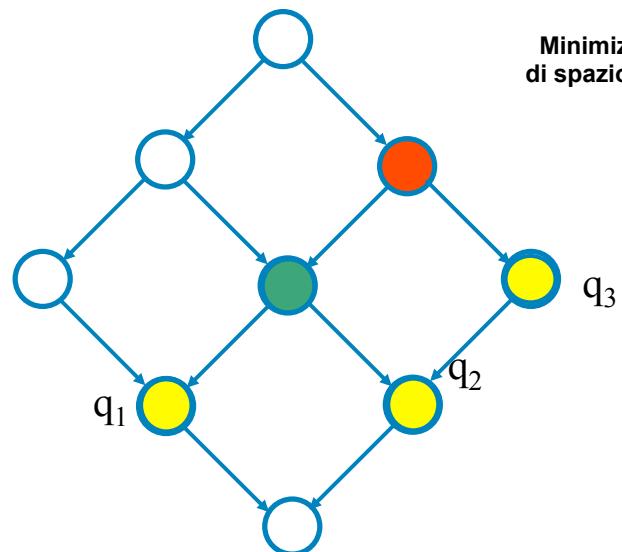
# Scelta delle viste



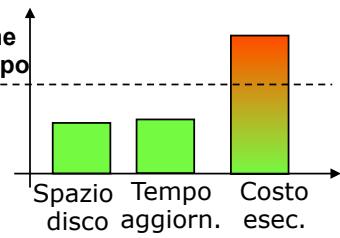
**viste candidate**,  
ossia potenzialmente  
utili a ridurre il costo  
di esecuzione del  
carico di lavoro

78

# Scelta delle viste

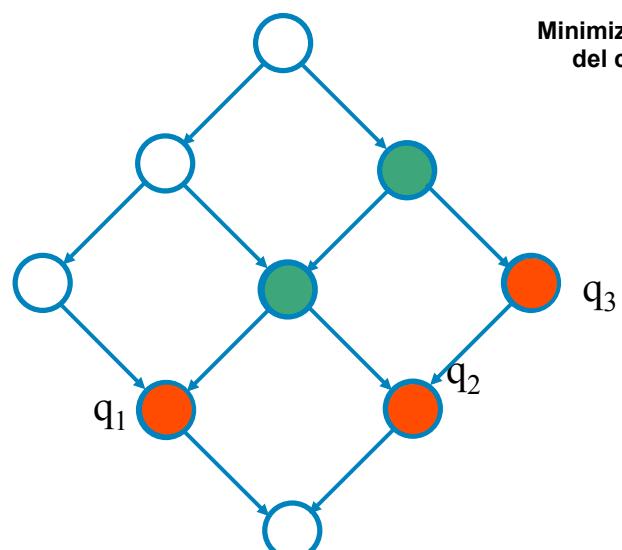


Minimizzazione  
di spazio e tempo

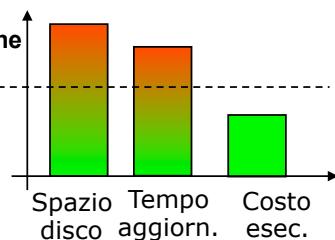


79

# Scelta delle viste

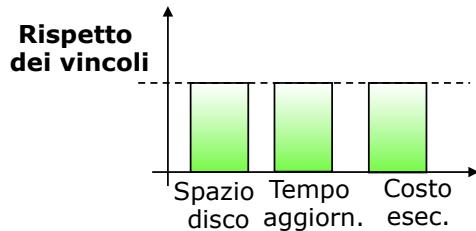
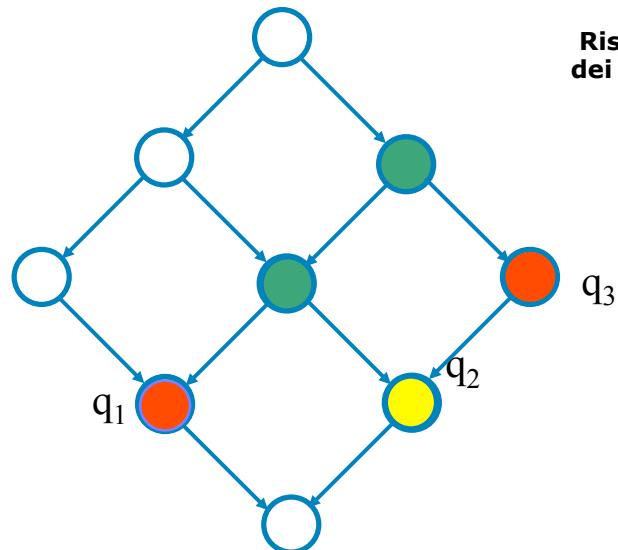


Minimizzazione  
del costo



80

# Scelta delle viste

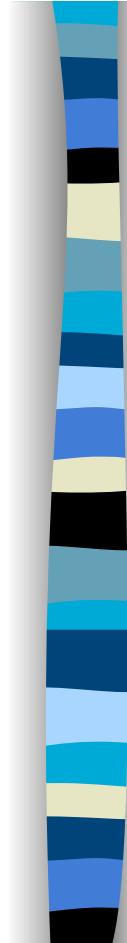


81

# Scelta delle viste

- 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

82



# Frammentazione delle viste

- Con il termine frammentazione si intende la suddivisione delle fact table (primarie e secondarie) in più frammenti al fine di aumentare le prestazioni del sistema.
- Le specifiche caratteristiche dei DW (ridondanza dei dati, cubi correlati, ecc.) rendono particolarmente utile questa forma di ottimizzazione.
  - ✓ **Frammentazione orizzontale:** la relazione viene suddivisa in più parti, ognuna delle quali contiene tutti gli attributi ma solo una parte delle tuple di quella di origine.
  - ✓ **Frammentazione verticale:** la relazione viene suddivisa in più parti, ognuna delle quali contiene tutte le tuple ma solo una parte degli attributi di quella di origine.

83

## 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.

84

# 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

- La frammentazione verticale:
  - ✓ 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 ognqualvolta si evita di materializzare una misura

85

# Progettazione dell'ETL

# Progettazione dell'ETL

- Durante la fase di progettazione dell'ETL 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

87

# Alimentazione dello schema riconciliato

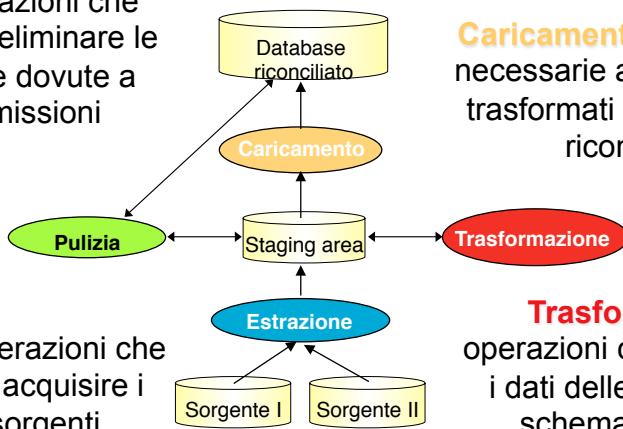
**Staging area:** spazio utilizzato per memorizzare in via transitoria le informazioni necessarie all'esecuzione delle procedure

**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



88

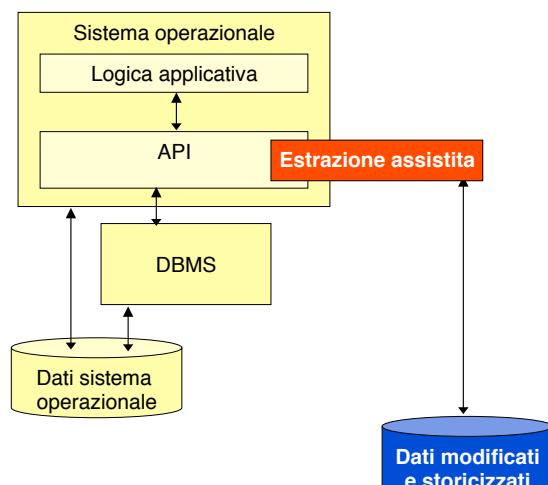
# Estrazione dei dati

- Le operazioni di estrazione dipendono dalla natura dei dati presenti nelle sorgenti operazionali
  - ✓ **Transitoria:** il sistema mantiene solo l' immagine corrente sovrascrivendo i dati che non sono più validi (es. dati di inventario, scorte di magazzino)
  - ✓ **Semi-storicizzata:** il sistema mantiene un limitato numero degli stati precedenti e non è possibile determinare per quanto tempo ciascun dato verrà conservato nel sistema
  - ✓ **Storicizzata:** tutte le modifiche intervenute nei dati vengono mantenute in un intervallo di tempo ben definito (es. dati bancari e assicurativi)
- L' estrazione può essere
  - ✓ **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

89

# Estrazione dei dati

- Estrazione assistita dall' applicazione
  - ✓ Tecnica di estrazione immediata
  - ✓ Le modifiche vengono rilevate da specifiche funzioni implementate direttamente all' interno delle applicazioni
  - ✓ Richiedono la modifica delle applicazioni OLTP
  - ✓ È utile quando si lavora con sistemi legacy che non forniscono sistemi di triggering, log, ecc.
  - ✓ Trova applicazione anche nei sistemi moderni quando è disponibile un livello di API comuni a tutte le applicazioni: una sola modifica per tutti gli accessi di uno stesso tipo

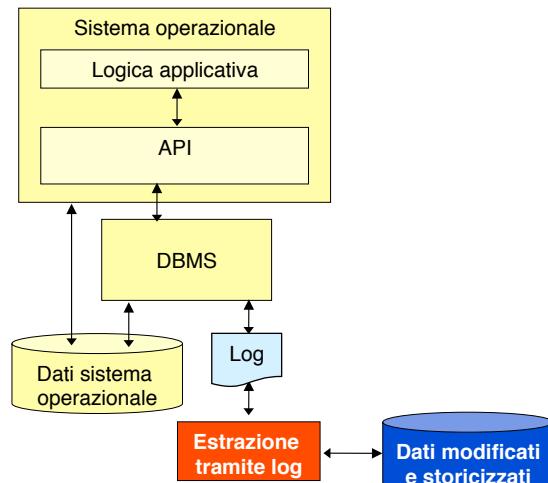


90

# Estrazione dei dati

## ■ Estrazione basata su log

- ✓ Tecnica di estrazione ritardata
- ✓ Le modifiche vengono memorizzate in appositi file prodotti dal DBMS
- ✓ Può risultare molto complesso interpretare il contenuto dei file il cui formato è normalmente proprietario dello specifico DBMS
- ✓ È consigliabile solo quando il modulo di estrazione è fornito direttamente dal produttore del DBMS

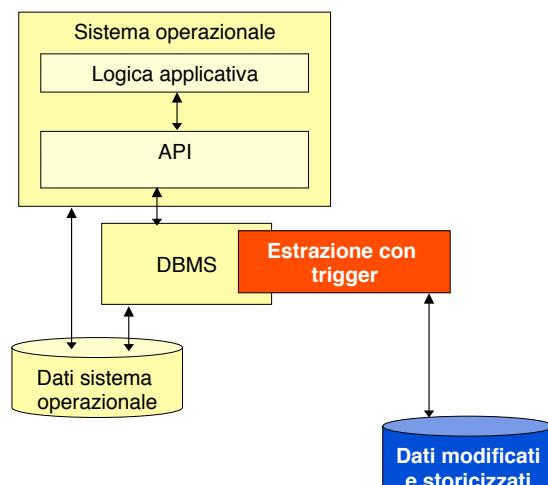


91

# Estrazione dei dati

## ■ Estrazione basata su trigger

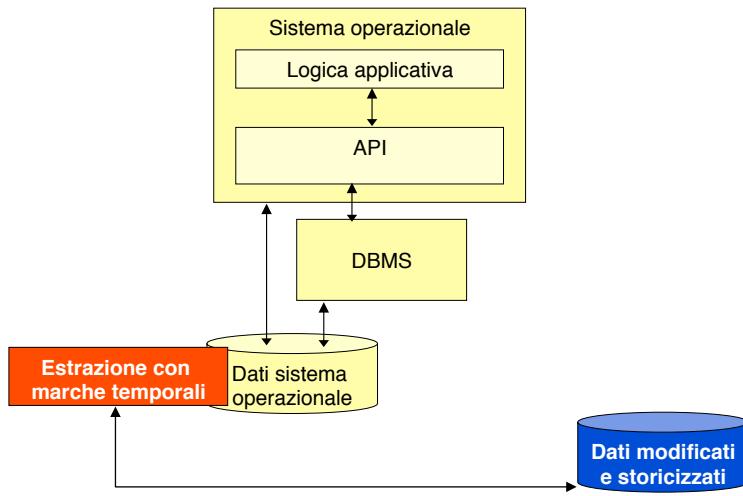
- ✓ Tecnica di estrazione immediata
- ✓ Le modifiche vengono individuate mediante funzioni basate su eventi implementate e controllate direttamente nel DBMS
- ✓ Per motivi prestazionali non è possibile adottare in modo estensivo questa tecnica che richiederebbe al DBMS di monitorare continuamente tutte le transazioni potenzialmente in grado di innescare un trigger



92

# Estrazione dei dati

- Estrazione basata su 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



93

# Estrazione dei dati

- L'efficacia della tecnica basata su marche temporali dipende dalla struttura stessa del sistema operazionale:

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

Situazione al 1/4/2002

Cod	prodotto	cliente	qtà	Data
1	Greco di tufo	Malavasi	50	15/3/2002
2	Barolo	Maio	100	1/4/2002
...	...	...	...	...



Estratto 1/4/2002

Situazione al 2/4/2002

Cod	prodotto	cliente	qtà	Data
1	Greco di tufo	Malavasi	50	15/3/2002
2	Barolo	Maio	200	2/4/2002
...	...	...	...	...

94

# Estrazione dei dati

- L'efficacia della tecnica basata su marche temporali dipende dalla struttura stessa del sistema operazionale:

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

Situazione al 3/4/2002

Cod	prodotto	cliente	qtà	Data
1	Greco di tufo	Malavasi	50	15/3/2002
2	Barolo	Maio	150	3/4/2002
...	...	...	...	...

→ Estratto 3/4/2002

Situazione al 2/4/2002

Cod	prodotto	cliente	qtà	Data
1	Greco di tufo	Malavasi	50	15/3/2002
2	Barolo	Maio	200	2/4/2002
...	...	...	...	...

← Modifica persa

95

# Il 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

96

# Il risultato dell' estrazione

Situazione al 4/4/2002

cod	prodotto	cliente	qtà
1	Greco di tufo	Malavasi	50
2	Barolo	Maio	150
3	Barbera	Lumini	75
4	Sangiovese	Cappelli	45

Situazione al 6/4/2002

cod	prodotto	cliente	qtà
1	Greco di tufo	Malavasi	50
2	Barolo	Maio	150
4	Sangiovese	Cappelli	145
5	Vermentino	Maltoni	25
6	Trebbiano	Maltoni	150

Differenza incrementale

cod	prodotto	cliente	qtà	oper
3	Barbera	Lumini	75	C
4	Sangiovese	Cappelli	145	M
5	Vermentino	Maltoni	25	I
6	Trebbiano	Maltoni	150	I

97

# 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 storizziazione del livello riconciliato
  - ✓ Estrazione statica → Riscrittura completa
  - ✓ Estrazione incrementale
    - Livello riconciliato non storizzato: memorizzo solo il tipo di operazione che ha determinato la variazione
    - Livello riconciliato storizzato: memorizzo anche una coppia di marche temporali che indicano l' intervallo di validità della tupla

Il livello di storizziazione dello schema riconciliato dipende da quello delle sorgenti operazionali e dai requisiti utente relativi alla reportistica operativa

98

# Trasformazione e pulizia

- 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 inconsistenti può essere prevenuta rendendo più rigorose le regole di inserimento dei dati nelle applicazioni del sistema operazionale

99

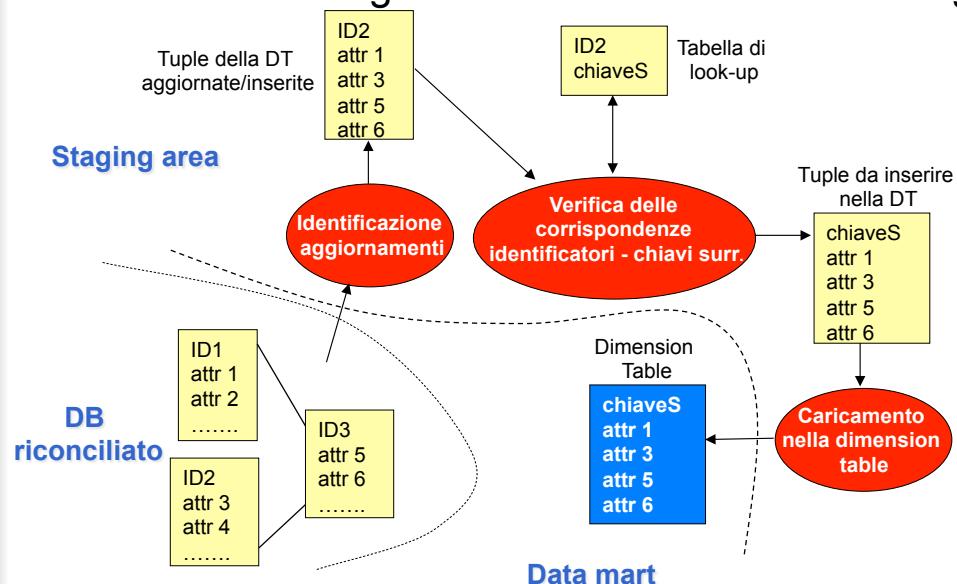
# Trasformazione e pulizia

- Ogni problema richiede una tecnica specifica per la soluzione e molti sistemi commerciali propongono moduli specifici per la pulizia dei dati
  - ✓ 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

100

# Alimentazione delle dimension table

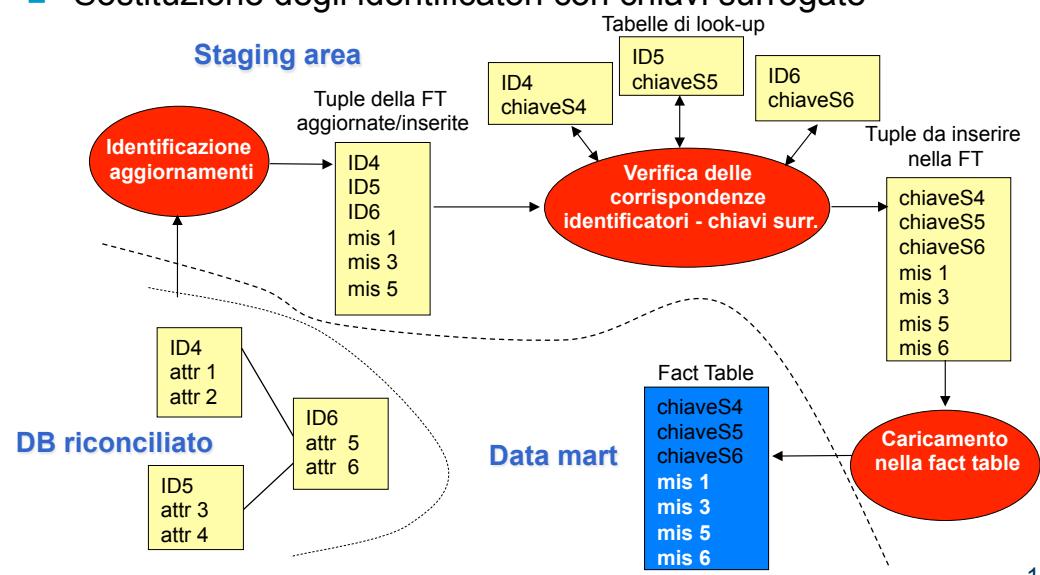
- Identificazione dei dati da caricare
- Sostituzione degli identificatori con chiavi surrogate



101

# Alimentazione delle fact table

- 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

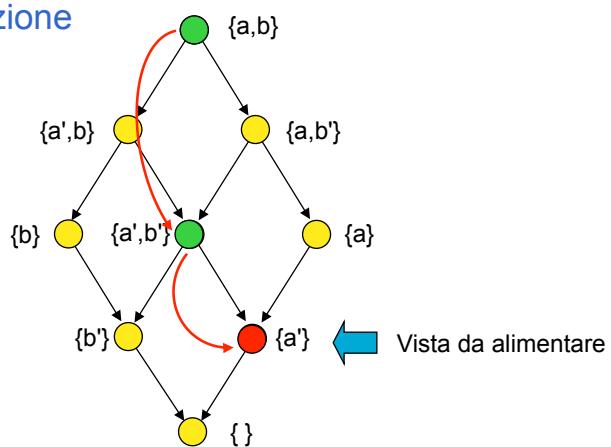


102

# Alimentazione delle viste

- Scelta della vista aggiornata che minimizza il costo di aggiornamento

✓ È la più piccola vista che permette di risolvere l'interrogazione



- Attenzione alla corretta scelta dell'operatore di aggregazione

103

# Progettazione fisica

# Indici per il Data Mart

- Le specifiche caratteristiche dei data mart permettono di utilizzare classi di indici diverse dal ben noto B<sup>+</sup>-tree utilizzato nella maggior parte dei DBMS commerciali.
  - ✓ Accessi in sola lettura
  - ✓ Aggiornamento periodico dei dati con possibilità di riorganizzazione degli indici
  - ✓ Accessi ad ampie porzioni di dati
- Alcuni indici sono nati in conseguenza delle esigenze di data warehousing
- Altri erano preesistenti ma non venivano utilizzati

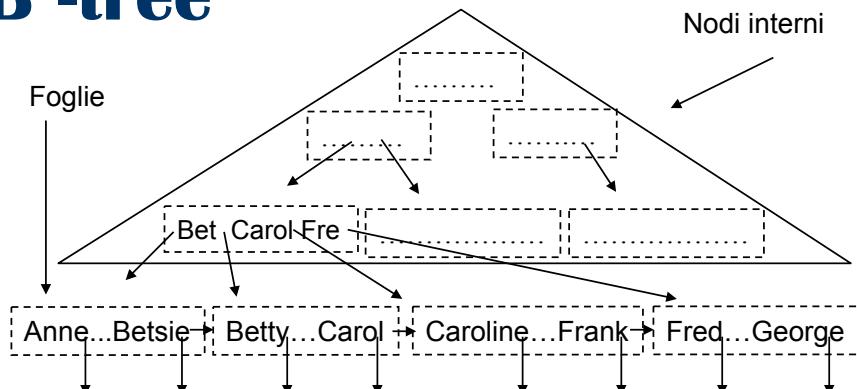
Indici Bitmap

Indici di join

Indici Star

105

## I B<sup>+</sup>-tree



- Le **foglie** contengono tutti i valori di chiave
- I **nodi interni**, organizzati come un B-tree, costituiscono solo una mappa per consentire una rapida localizzazione delle chiavi, e memorizzano dei separatori

106

# Perché i B<sup>+</sup>-tree non sono più sufficienti ?

- Forniscono buone prestazioni quando la selettività dei predicati è molto elevata.
  - ✓ Le interrogazioni OLAP utilizzano spesso predicati a bassa selettività (es. sesso)
- Sono più adatti a interrogazioni semplici
- Possono richiedere molto spazio per la loro memorizzazione

I B<sup>+</sup>-tree non sono più sufficienti ma rimangono ancora molto utili

107

## Gli indici 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
- Il bitmap  $(i,j)$  è posto a TRUE se nella tupla  $i$ -esima è presente il valore  $j$ -esimo

Esempio: Indice sul campo Posizione della tabella impiegati  
Ingegnere – Consulente – Manager – Programmatore  
Assistente – Ragioniere

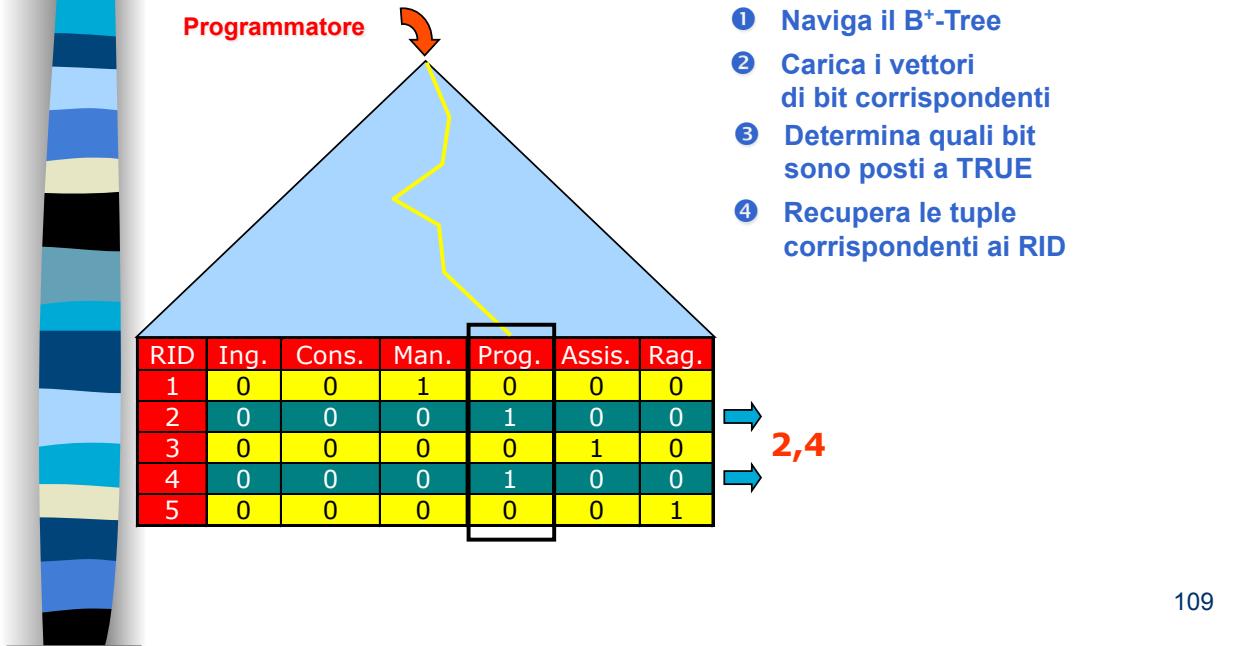
L' impiegato corrispondente al RID 1 è un Manager

RID	Ing.	Cons.	Man.	Prog.	Assis.	Rag.
1	0	0	1	0	0	0
2	0	0	0	1	0	0
3	0	0	0	0	1	0
4	0	0	0	1	0	0
5	0	0	0	0	0	1

108

# Implementazione dei bitmap

- Normalmente i bitmap sono associati a B<sup>+</sup>-Tree le cui foglie contengono vettori di bit invece di RID



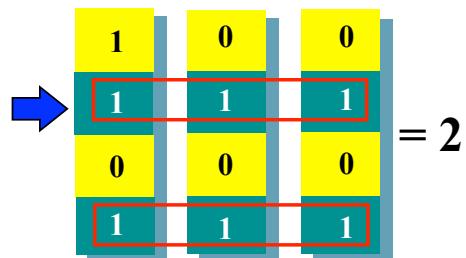
109

# I vantaggi degli indici bitmap

- Lo spazio richiesto su disco può essere molto ridotto
- I/O è molto basso poiché vengono letti solo i vettori di bit necessari
- Ottimi per interrogazioni che non richiedono l'accesso ai dati
- Permettono l'utilizzo di operatori binari per l'elaborazione dei predicati

Esempio: “*Quanti maschi in Emilia-Romagna sono assicurati?*”

RID	Sesso	Assic.	Regione
1	M	No	LO
2	M	Sì	E/R
3	F	No	LA
4	M	Sì	E/R



110

# Occupazione su disco

- Gli indici bitmap sono adatti ad attributi con una ridotta cardinalità poiché ogni nuovo valore distinto di chiave richiede un ulteriore vettore di bit
- All'aumentare del numero di chiavi distinte aumenta la sparsità della matrice

## Esempio:

$NR = 10.000.000$

$Len(Pointer) = 4 \times 8 \text{ bit}$

**B-tree**

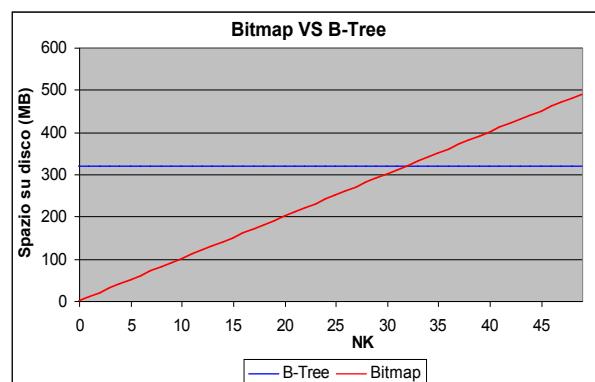
**Bitmap**

$NR \times Len(Pointer)$

$NR \times NK \times 1 \text{ bit}$

Si ha un risparmio di spazio se:

$$\text{Densità media} \geq \frac{1}{Len(RID)}$$

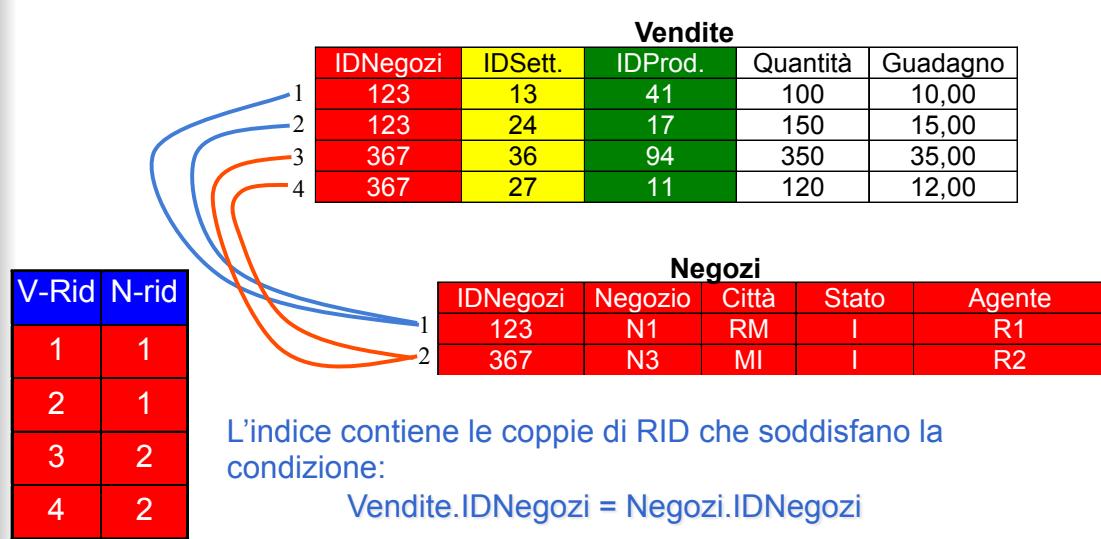


La compressione delle matrici riduce il fattore di crescita della dimensione

111

# Gli indici di join

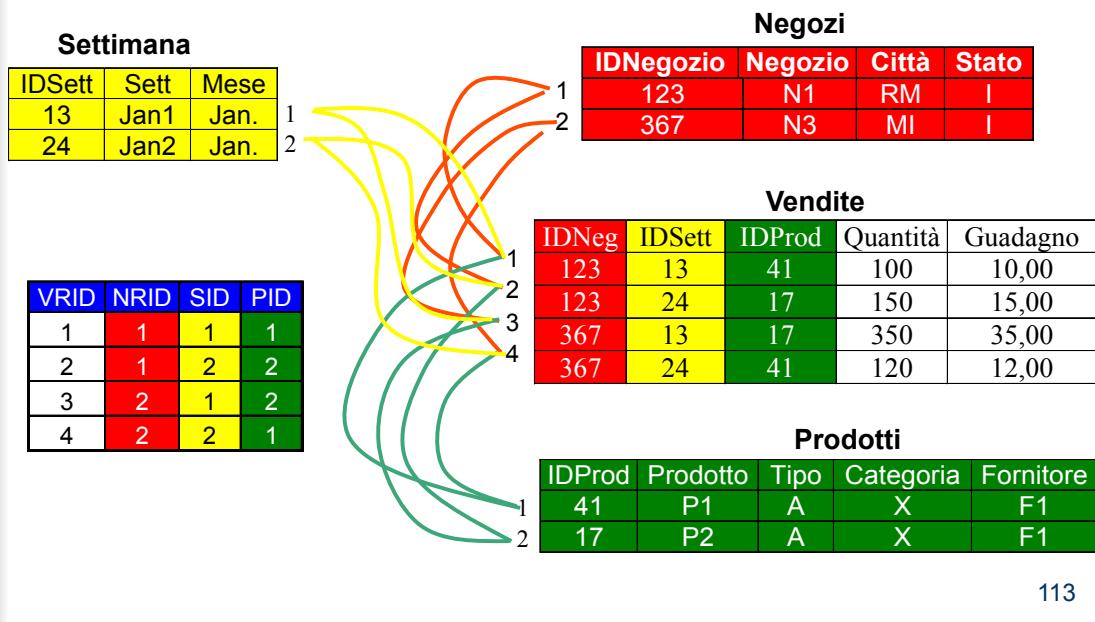
- Le interrogazioni su schemi a stella richiedono sempre uno o più join
- Un indice di join calcola in anticipo le tuple che soddisfano un particolare predicato di join



112

# Gli indici a stella

- Estendono il concetto di indice di join a più tabelle concatenando i valori delle colonne della fact table e di più dimension table



# Gli indici a stella

- 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

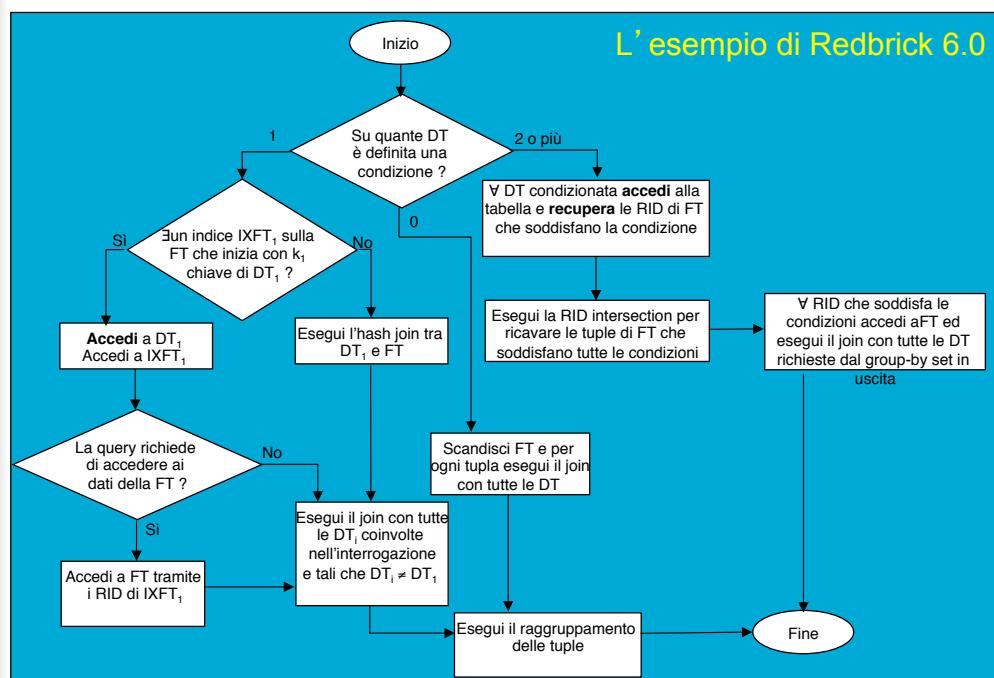
# Si consiglia di creare indici su..

- Chiavi importate sulla fact table per aumentare la velocità di esecuzione dei join (B<sup>+</sup>-Tree o indici di join, indici star, bitmapped join index)
- Attributi dimensionali che sono spesso coinvolti nei criteri di selezione (B<sup>+</sup>-Tree o bitmap)
- Misure che sono spesso coinvolte in clausole di selezione (bitmap evoluti)

**Se il DBMS non utilizza statistiche per definire il piano di accesso la creazione degli indici deve essere valutata con molta attenzione**

115

# Si consiglia di creare indici su..



116