Progettazione logica

Prof. Stefano Rizzi

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.

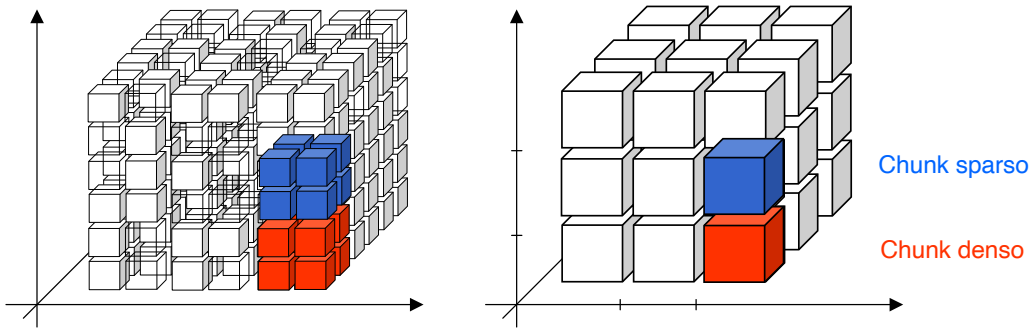
Sistemi MOLAP

- 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.

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.

Sistemi MOLAP e sparsità

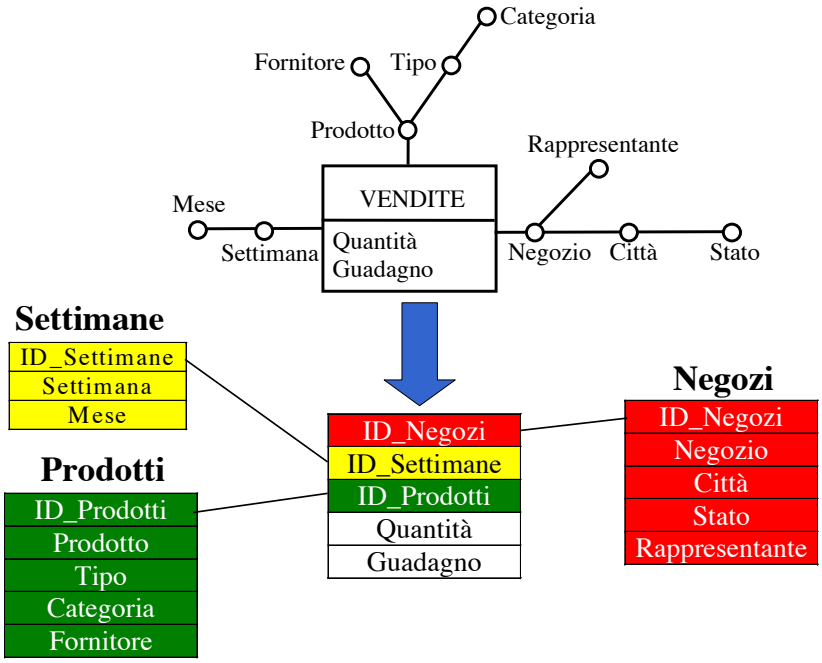


Una struttura dati comunemente usata per la compressione dei chunk sparsi prevede un indice che riporti il solo offset delle celle che effettivamente contengono informazioni.

ROLAP: lo schema a stella

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

Lo schema a stella



Lo schema a stella

ID_Negozi	Negozio	Città	Stato	Rappresentante
1	N1	RM	I	R1
2	N2	RM	I	R1
3	N3	MI	I	R2
4	N4	MI	I	R2

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	Gen1	Gen.
2	Gen2	Gen.
3	Feb1	Feb.
4	Feb2	Feb.

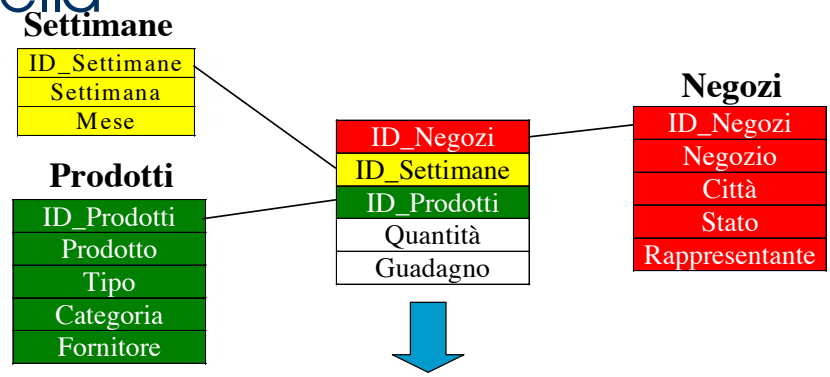
Dimension Table

ID_Prodotti	Prodotto	Tipo	Categoria	Fornitore
1	P1	A	X	F1
2	P2	A	X	F1
3	P3	B	X	F2
4	P4	B	X	F2

Lo schema a stella: considerazioni

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

Interrogazioni OLAP su schemi a stella



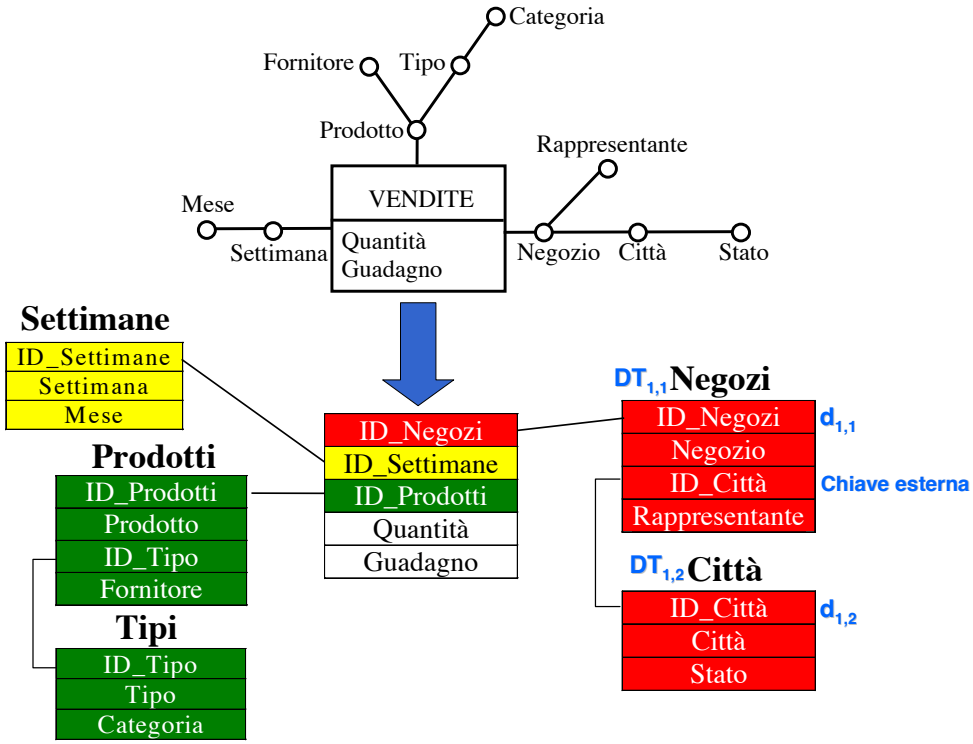
VENDITE(Negozi.Città, Settimane, Prodotti.Tipo;
Prodotto.Categoria='Alimentari').Quantità

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

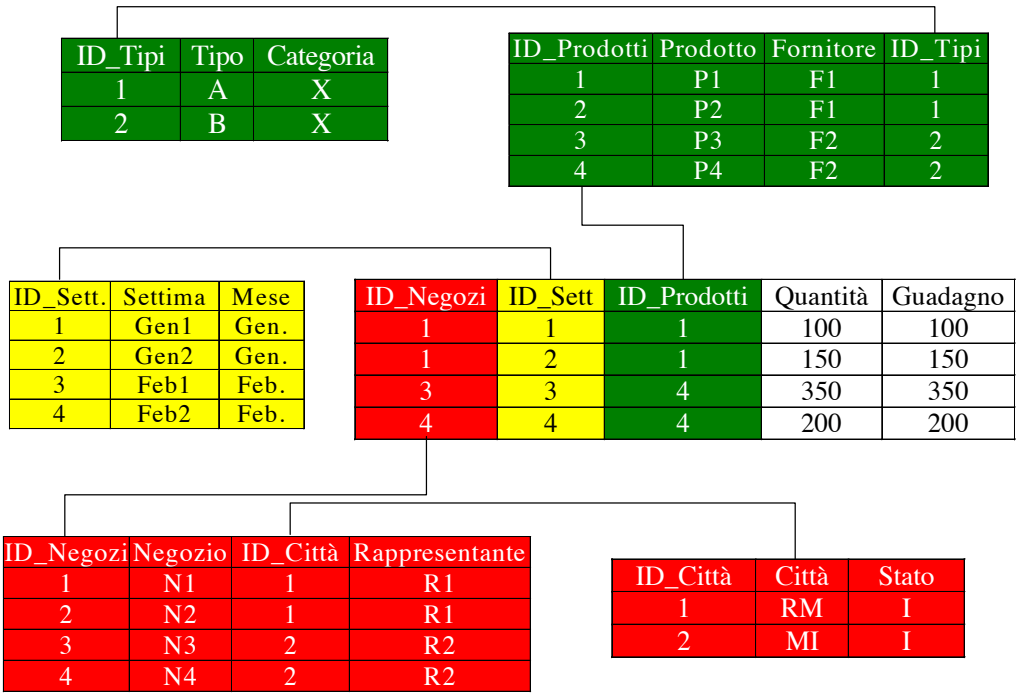
Lo snowflake schema

- Lo schema a fiocco di neve (*snowflake schema*) riduce la denormalizzazione delle dimension table DT_i degli schemi a stella eliminando alcune delle dipendenze transitive che le caratterizzano.
- Le dimension table $DT_{i,j}$ di questo schema sono caratterizzate da:
  - ✓ una chiave primaria (tipicamente surrogata) $d_{i,j}$
  - ✓ il sottoinsieme degli attributi di DT_i che dipendono funzionalmente da $d_{i,j}$.
  - ✓ zero o più chiavi esterne 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.

Lo snowflake schema



Lo snowflake schema



ID_Tipi	Tipo	Categoria
1	A	X
2	B	X

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

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

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

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

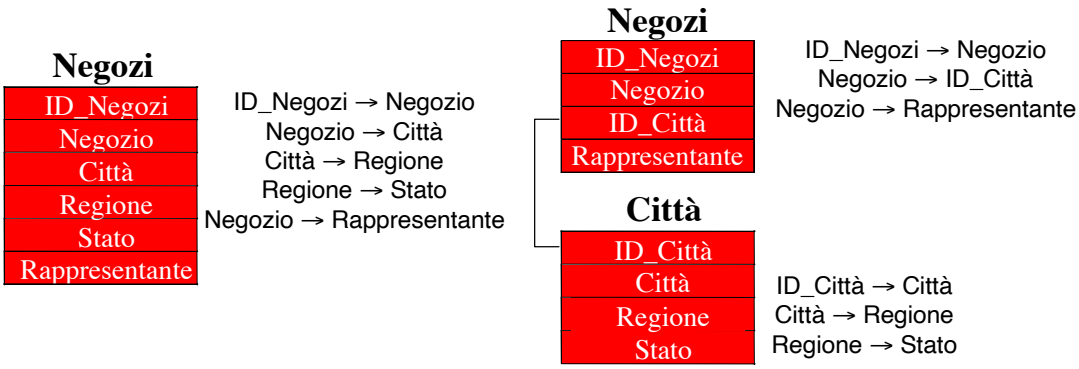
ID_Città	Città	Stato
1	RM	I
2	MI	I

Lo snowflake schema: considerazioni

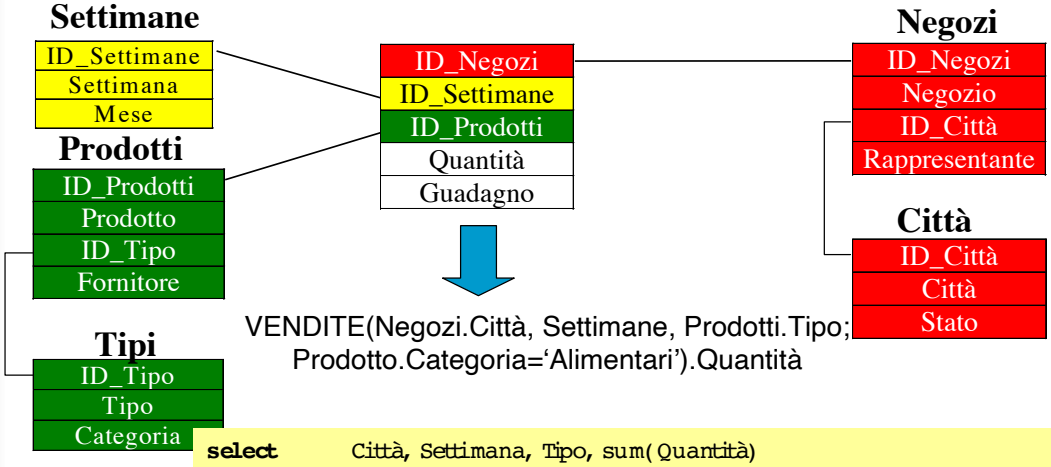
- Lo spazio richiesto per la memorizzazione dei dati si riduce grazie alla normalizzazione
- È necessario inserire nuove chiavi surrogate che permettano di determinare le corrispondenze tra dimension table primarie e secondarie
- L'esecuzione di interrogazioni che coinvolgono solo gli attributi contenuti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

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



Interrogazioni OLAP su schemi snowflake



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

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

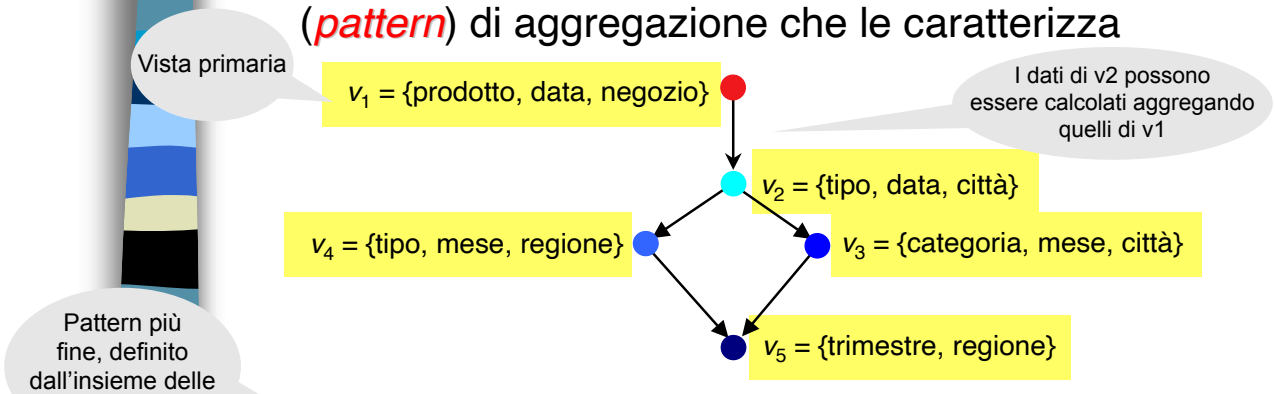
Le viste

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

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

Le viste

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

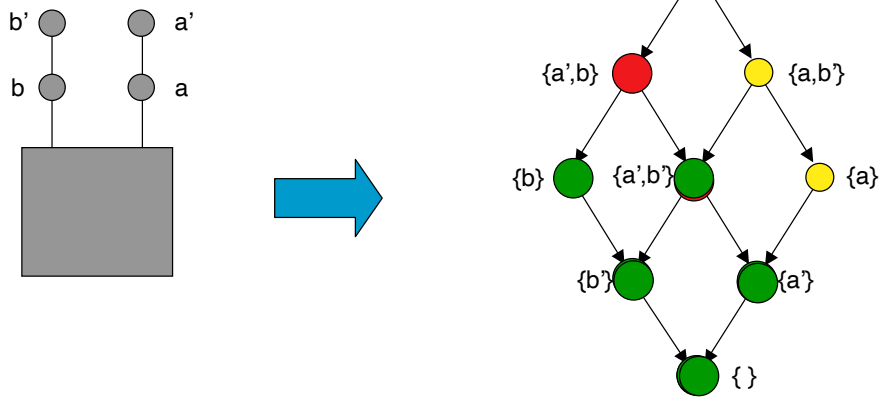


- **Viste primarie:** corrispondono al pattern di aggregazione primario (non aggregato)
- **Viste secondarie:** corrispondono ai pattern di aggregazione secondari (aggregati)

Una interrogazione che richieda i dati aggregati per tipo prodotto, data di vendita e città del negozio risulterà meno costosa se eseguita sulla vista v2 (piuttosto che su v1) poiché insisterà su una fact table con un numero ridotto di tuple e non richiederà ulteriori operazioni di aggregazione.

Risolvibilità delle interrogazioni

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



Reticolo multidimensionale

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.

I record delle dimension table corrispondenti a dati aggregati presenteranno dei valori NULL in tutti i campi il cui livello di aggregazione è più fine di quello su cui si sta operando.

Schemi relazionali e viste

Sono relative al pattern:
{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	Fornitore
1	P1	A	X	F1
2	P2	A	X	F1
3	P3	B	X	F2
4	P4	B	X	F2
5	-	A	X	F1
6	-	B	X	F2
7	-	-	-	F1

Schemi relazionali e viste

Sono relative al pattern:
{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	P1	A	X	F1
2	P2	A	X	F1
3	P3	B	X	F2
4	P4	B	X	F2
5	-	A	X	F1
6	-	B	X	F2
7	-	-	-	F1

Schemi relazionali e viste

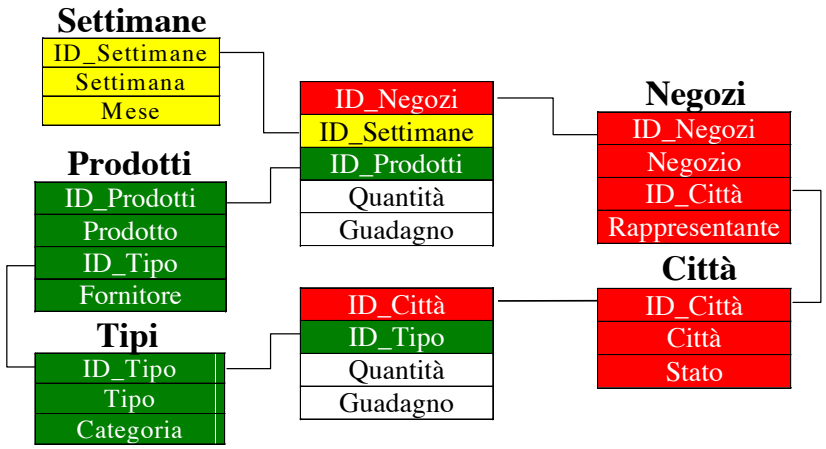
È relativa al pattern:
{Negozi,Settimane,**Fornitore**}

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	P1	A	X	F1
2	P2	A	X	F1
3	P3	B	X	F2
4	P4	B	X	F2
5	-	A	X	F1
6	-	B	X	F2
7	-	-	-	F1

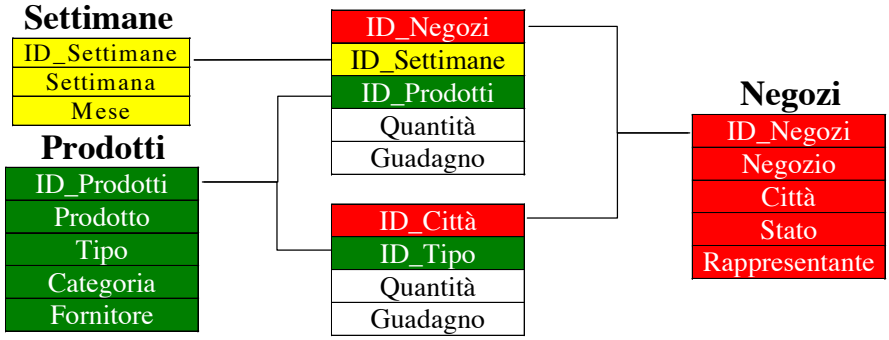
Schemi relazionali e viste

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



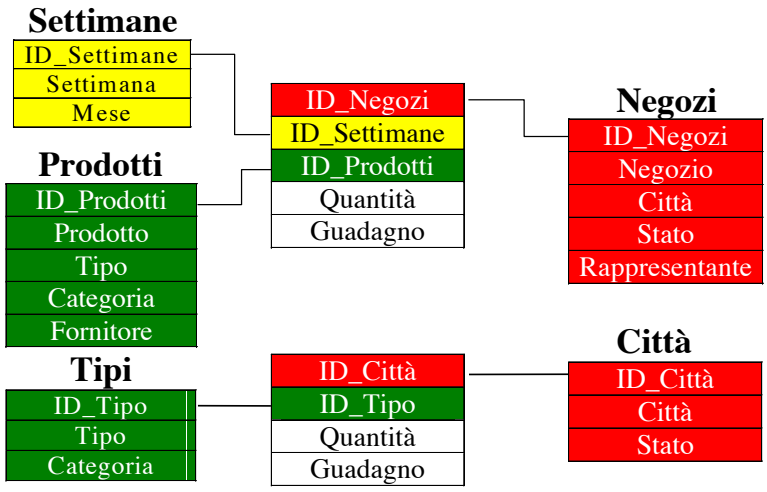
Schemi relazionali e viste

- Una soluzione intermedia rispetto alle due presentate prevede di memorizzare in fact table separate dati relativi a pattern di aggregazione 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.



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



Aggregate navigator

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

Progettazione logica

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

INPUT

Schema concettuale
Carico di lavoro
Volume dei dati
Vincoli di sistema

OUTPUT

Schema logico

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

Progettazione logica

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

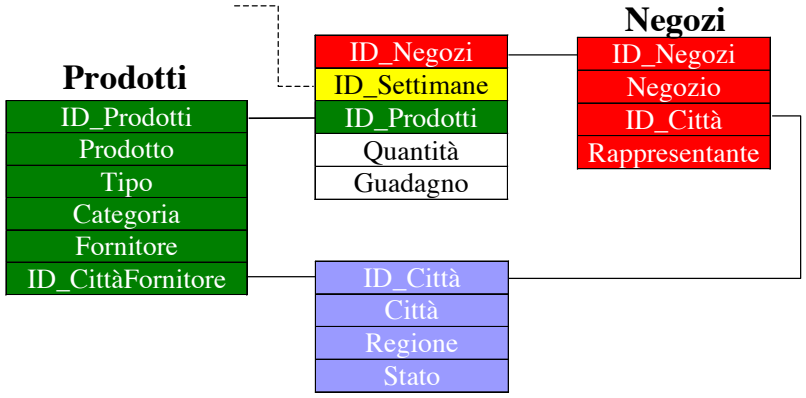
1. Scelta dello schema logico da utilizzare (es. star/snowflake schema)
2. Traduzione degli schemi concettuali
3. Scelta delle viste da materializzare
4. Applicazione di altre forme di ottimizzazione (es. frammentazione verticale/orizzontale)

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
  - ✓ Quando il rapporto tra le cardinalità della dimension table primaria e secondaria è elevato, poiché determina un forte risparmio di spazio

Star VS Snowflake

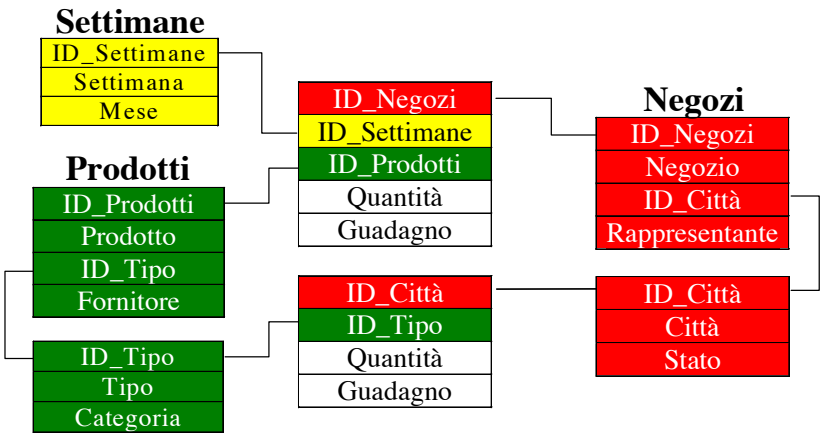
- Può essere utile
  - ✓ Quando una porzione di una gerarchia è comune a più dimensioni



La dimension table secondaria è riutilizzata per più gerarchie

Star VS Snowflake

- Può essere utile
  - ✓ In presenza di viste aggregate



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

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.

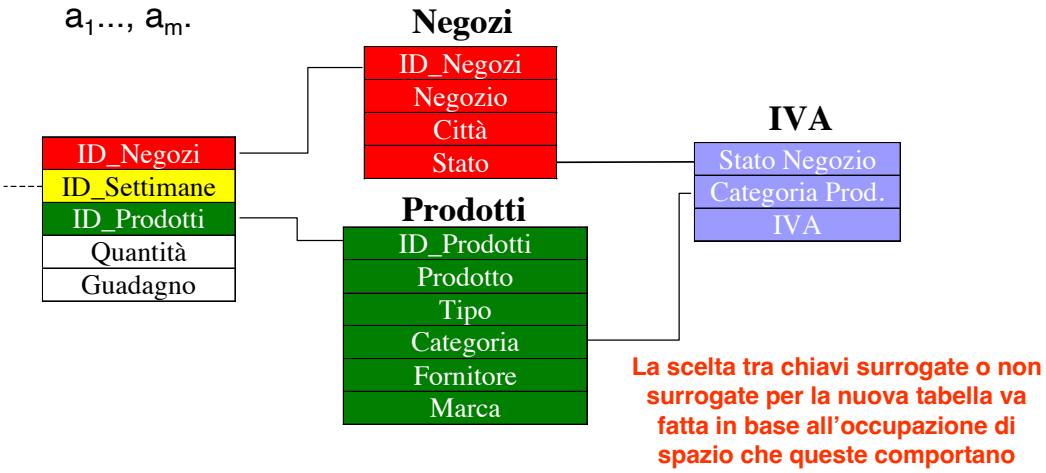
- In aggiunta a questa semplice regola, la corretta traduzione di uno schema di fatto richiede una trattazione approfondita dei costrutti avanzati del DFM.

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.
- Ha senso solo se è compatibile con il livello di granularità dell'evento descritto nella fact table, quindi se connesso direttamente alla fact table dovrà essere omesso nelle viste aggregate.

Attributi cross-dimensionali

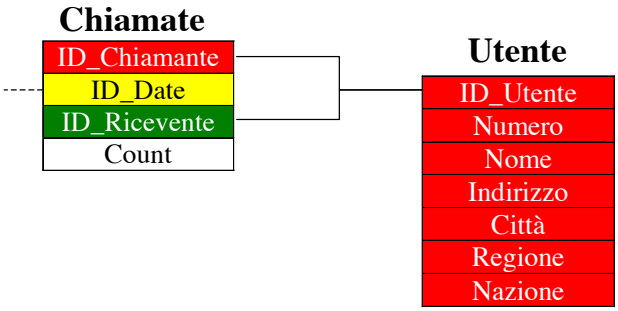
- Dal punto di vista concettuale, un attributo cross-dimensionale b definisce un'associazione molti-a-molti tra due o più attributi dimensionali a_1..., a_m.
- La sua traduzione a livello logico richiede l'inserimento di una nuova tabella che includa b e abbia come chiave gli attributi a_1..., a_m.



La scelta tra chiavi surrogate o non surrogate per la nuova tabella va fatta in base all'occupazione di spazio che queste comportano

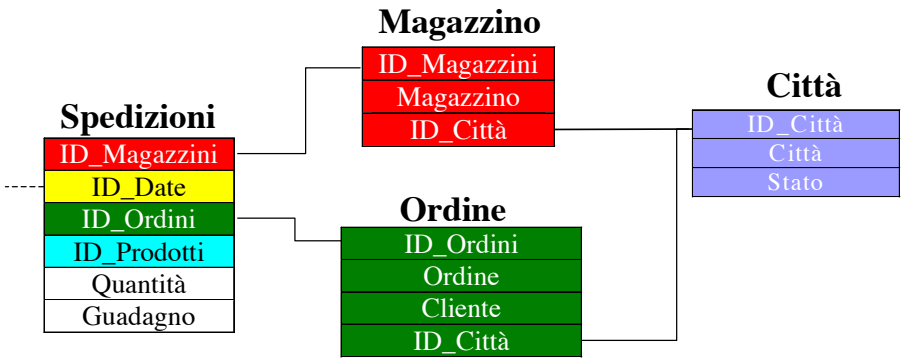
Gerarchie condivise

- 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 dimesion table



Gerarchie condivise

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

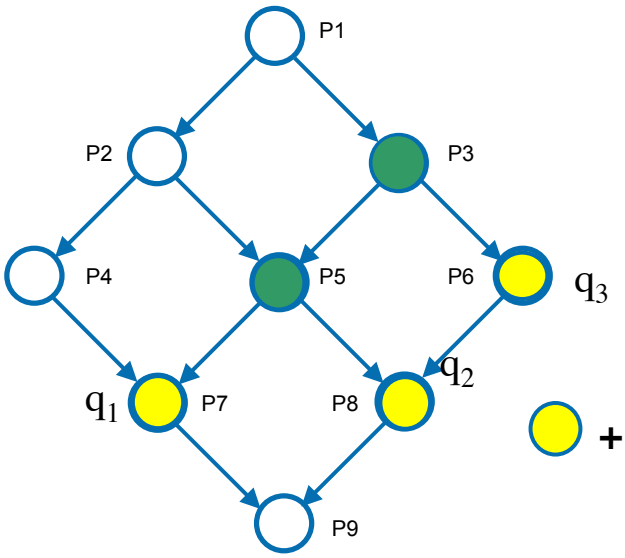


Scelta delle viste

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

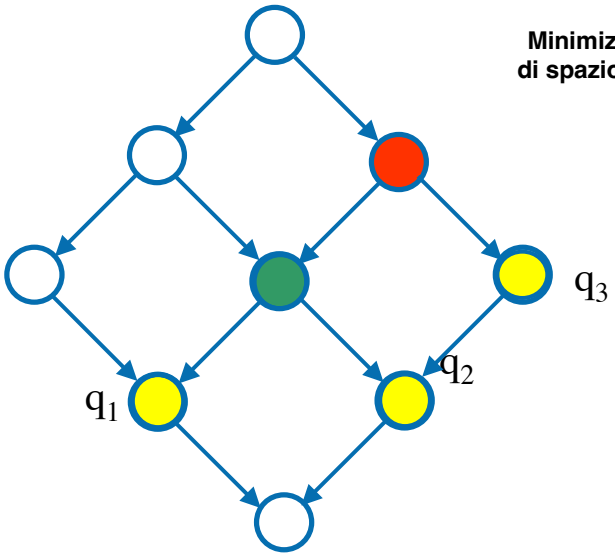
1. Minimizzazione di funzioni di costo
2. Vincoli di sistema
   - ✓ Spazio su disco
   - ✓ Tempo a disposizione per l'aggiornamento dei dati
3. Vincoli utente
   - ✓ Tempo massimo di risposta
   - ✓ Freschezza dei dati

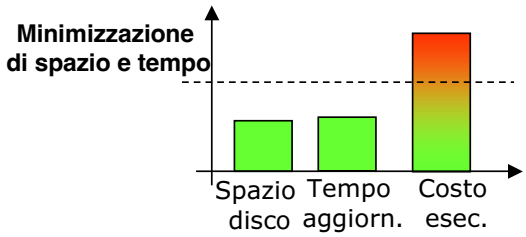
Scelta delle viste



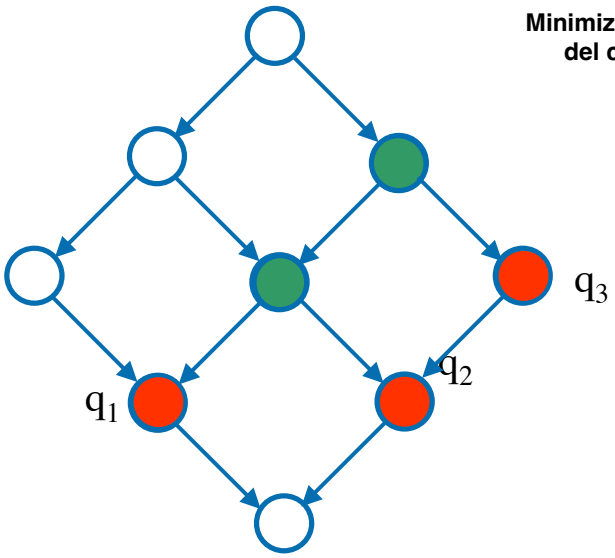
+ = *viste candidate*, **ossia potenzialmente utili a ridurre il costo di esecuzione del carico di lavoro**

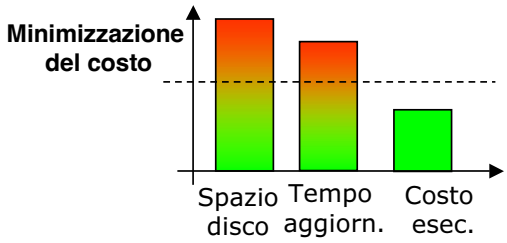
Scelta delle viste



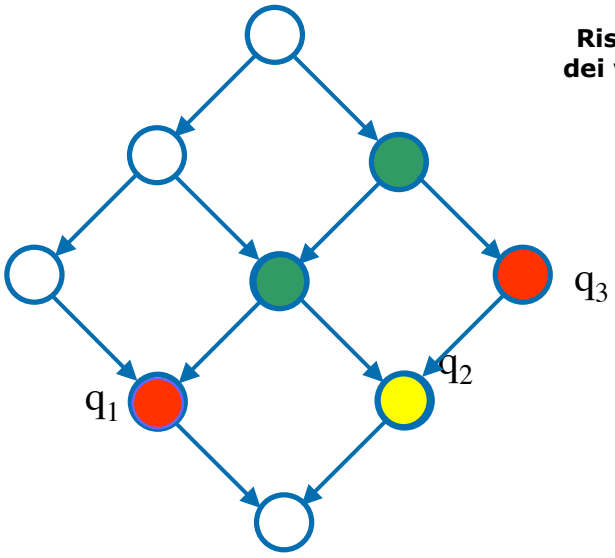


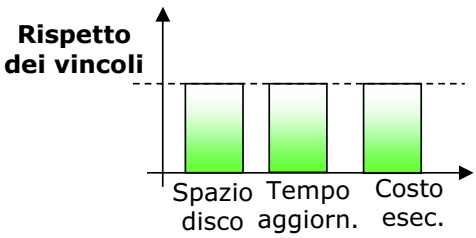
Scelta delle viste





Scelta delle viste





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 pattern di aggregazione è molto simile a quello di una vista già materializzata
  - ✓ Il suo pattern di aggregazione è molto fine
  - ✓ La materializzazione non riduce di almeno un ordine di grandezza il costo delle interrogazioni