

# Il modello dimensionale

Luca Cabibbo  
marzo 2010

## Il modello dimensionale

La progettazione dei dati del data warehouse è la pietra angolare del progetto dell'intero data warehouse

- basandosi sul progetto dei dati è possibile
  - pianificare e progettare le applicazioni
  - pianificare l'estrazione e la trasformazione dei dati
  - stimare l'occupazione di memoria complessiva del data warehouse

La progettazione dei dati in un data warehouse dimensionale

- basata sulla modellazione dimensionale
- resa coerente dall'adozione di un'architettura a bus del data warehouse

# Schemi dimensionali

La modellazione dimensionale è una tecnica di progettazione logica dei dati nel data warehouse

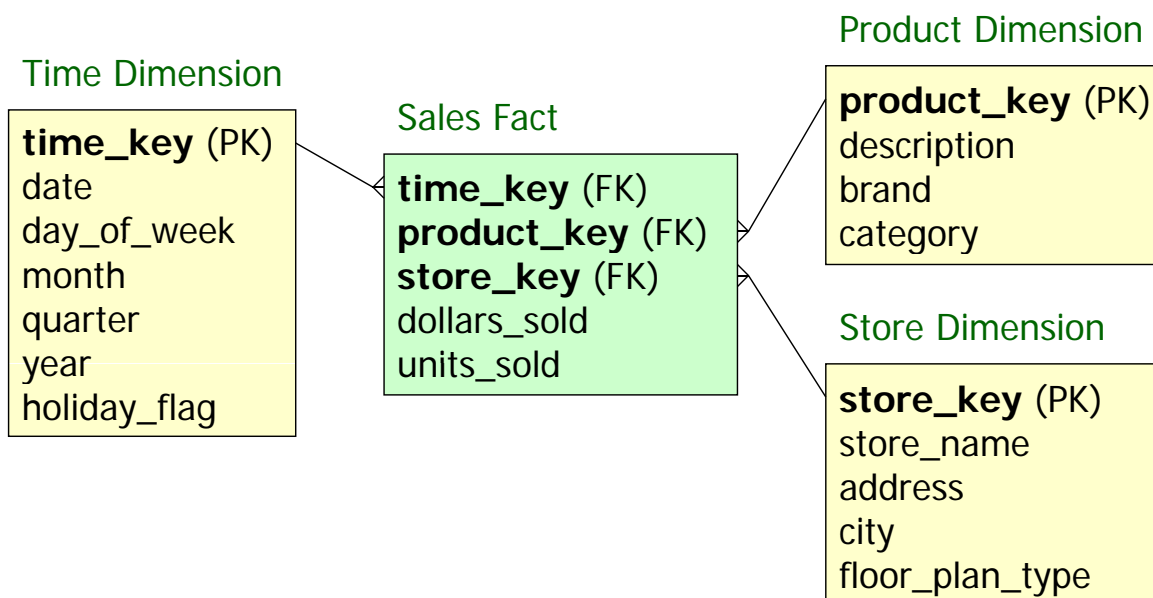
- è orientata alla definizione di schemi relazionali di tipo “dimensionale”
- uno **schema dimensionale** (chiamato anche **star schema** o **schema a stella**) è composto da
  - una tabella principale – chiamata **tabella fatti**
  - un insieme di tabelle ausiliarie – chiamate **tabelle dimensione**

3

Il modello dimensionale

Luca Cabibbo

## Esempio di schema dimensionale



- questo schema modella i dati delle vendite di prodotti in un certo numero di negozi nel corso del tempo
  - memorizza i totali giornalieri delle vendite dei prodotti per negozio

4

Il modello dimensionale

Luca Cabibbo

## Scopo di uno schema dimensionale

In uno schema dimensionale

- una **tabella dimensione** serve a rappresentare un insieme di elementi (un insieme in senso matematico), chiamati membri
- una **tabella fatti** serve a memorizzare un insieme di funzioni numeriche (funzioni in senso matematico)

Nell'esempio, lo schema rappresenta

- una dimensione **Product** di tipi di prodotti in vendita
- una dimensione **Time** di giorni in un intervallo di interesse
- una dimensione **Store** dei negozi di una catena di negozi
- una funzione **dollars\_sold: Product × Time × Store → R**
- una funzione **units\_sold: Product × Time × Store → N**

## Tabelle dimensione

Una **tabella dimensione** (**dimension table**) memorizza una dimensione rispetto a cui è di interesse analizzare un processo

- una **dimensione** è un dominio (insieme) di elementi, chiamati membri
  - ad es., un insieme di prodotti, un insieme di negozi o un insieme di giorni in un intervallo di tempo di interesse
- ciascuna riga di una tabella dimensione rappresenta un **membro** della dimensione
  - ad es., ciascuna riga della tabella **Product Dimension** descrive uno dei prodotti in vendita nella catena di negozi
- la chiave è semplice ed artificiale – di solito numerica
- gli altri campi (non chiave) di una tabella dimensione memorizzano gli **attributi** dei membri
  - gli attributi sono le proprietà dei membri – che sono solitamente testuali, discrete e descrittive

# Tabelle dimensione

## Time Dimension

time_key	date	...
1	1/1/2005	...
2	2/1/2005	...
3	3/1/2005	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
1461	31/12/2008	...

## Product Dimension

product_key	description	...
1	Lattina Coca Cola	...
2	Lattina Coca Cola Diet	...
3	Tubo Pringles Original	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
9827	Spaghetti De Cecco	...

store_key	description	...
1	MegaStore Marconi (RM)	...
2	MegaStore EUR (RM)	...
...	...	...
...	...	...
...	...	...
...	...	...
49	HyperStore Duomo (MI)	...

## Store Dimension

7

Il modello dimensionale

Luca Cabibbo

# Tabella fatti

Una **tabella fatti** (**fact table**) memorizza le misure numeriche (fatti) di un processo di business

- per **fatto** si intende una misura relativa ad un processo
- la chiave è normalmente composta da riferimenti alle chiavi delle varie tabelle dimensione
- gli altri campi rappresentano i fatti
  - questi fatti sono solitamente numerici, continui e additivi
- ciascuna riga della tabella fatti memorizza un insieme di misure (fatti) associati ad una particolare combinazione di membri, presa all'intersezione di tutte le dimensioni

8

Il modello dimensionale

Luca Cabibbo

# Tabella fatti

## Time Dimension

time_key	date	...
1	1/1/2005	...
2	2/1/2005	...
3	3/1/2005	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
1461	31/12/2008	...

## Sales Fact

time	prd	store	dollars sold	units sold
...	...	...	...	...
3	2	1	9.48	12
3	3	49	9.45	7
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

## Product Dimension

product_key	description	...
1	Lattina Coca Cola	...
2	Lattina Coca Cola Diet	...
3	Tubo Pringles Original	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
9827	Spaghetti De Cecco	...

store_key	description	...
1	MegaStore Marconi (RM)	...
2	MegaStore EUR (RM)	...
...	...	...
...	...	...
...	...	...
...	...	...
...	...	...
49	HyperStore Duomo (MI)	...

## Store Dimension

9

Il modello dimensionale

Luca Cabibbo

# Tabella fatti

Una tabella fatti serve

- a memorizzare un insieme di funzioni numeriche (in senso matematico)
  - una funzione per ciascuno dei fatti
  - il cui dominio è dato dall'insieme delle dimensioni
  - ciascuna di queste funzioni associa un valore a ciascuna possibile combinazione dei membri delle dimensioni
- in un modo adeguato per l'analisi dimensionale

Nell'esempio

- una funzione **dollars\_sold**: **Product** × **Time** × **Store** → **R**
- una funzione **units\_sold**: **Product** × **Time** × **Store** → **N**

10

Il modello dimensionale

Luca Cabibbo

## Schemi dimensioni, processi e grana

Ciascuno schema dimensionale serve a rappresentare i fatti di interesse per un certo processo di business, ad una certa granularità

- nell'esempio, il processo è la vendita di prodotti nei negozi di una catena di negozi
- i fatti sono
  - l'incasso in dollari (**dollars\_sold**)
  - la quantità venduta (**units\_sold**)
- la granularità a cui sono rappresentati di dati sono il totale giornaliero per prodotto e negozio

## Additività dei fatti

Un fatto è **additivo** se ha senso sommarlo rispetto ad ogni possibile combinazione delle dimensioni da cui dipende

- nell'esempio, l'incasso in dollari è additivo perché ha senso calcolare la somma degli incassi per un certo intervallo di tempo, insieme di prodotti e insieme di negozi
  - ad esempio, in un mese, per una categoria di prodotti e per i negozi in un'area geografica

L'additività è una proprietà importante, perché le applicazioni del data warehouse devono solitamente combinare (aggregare) i fatti descritti da molte righe di una tabella fatti

- il modo più comune di combinare un insieme di fatti è di sommarli (se questo ha senso)
- è possibile anche l'uso di altre operazioni – ad esempio, min, max, avg

## Semi additività e non additività

I fatti possono essere anche

- **semi additivi**
  - se ha senso sommarli solo rispetto ad alcune dimensioni
    - ad esempio, il numero di pezzi in deposito di un prodotto è sommabile rispetto alle categorie di prodotto e ai magazzini, ma non rispetto al tempo
- **non additivi**
  - se non ha senso sommarli
- può avere senso combinare fatti anche non completamente additivi mediante operazioni diverse dalla somma
  - ad esempio, min, max

## Sulla scelta delle chiavi

Alcuni commenti circa l'uso delle chiavi nei vari tipi di tabelle in uno schema dimensionale

- la chiave primaria di ciascuna tabella dimensione deve essere semplice e artificiale (“surrogata”)
  - tipicamente, un semplice numero intero
  - non deve avere nessun significato “naturale”
  - le chiavi originali “di produzione” non vanno assolutamente usate
  - perché? riuso di chiavi naturali, cambiamento delle chiavi naturali, dimensioni che cambiano lentamente, ...
  - corrispondenza con tra chiavi naturali e chiavi artificiali gestita nell'area di preparazione dei dati
- le chiavi delle tabelle fatti sono composte da chiavi esterne delle tabelle dimensione coinvolte

## Attributi e interrogazioni

Gli attributi delle tabelle dimensione sono il principale strumento per l'interrogazione del data warehouse

- gli attributi delle dimensioni vengono usati principalmente con due finalità
  - per selezionare un sottoinsieme dei dati di interesse
    - vincolando il valore di uno o più attributi
    - ad esempio, le vendite nel corso dell'anno 2007
  - per raggruppare i dati di interesse
    - usando gli attributi come intestazioni della tabella risultato
    - ad esempio, per mostrare le vendite per ciascuna categoria di prodotto in ciascun mese

## Attributi e interrogazioni

Un esempio di interrogazione

- somma degli incassi in dollari e delle quantità vendute
- per ciascuna categoria di prodotto in ciascun mese
- nel corso dell'anno 2007

(product) category	(time) month	(sum of) dollars_sold	(sum of) units_sold
Drinks	gennaio 2007	21.509,05	23.293
Drinks	febbraio 2007	19.486,93	22.216
Drinks	marzo 2007	21.986,43	23.532
Food	gennaio 2007	86.937,77	55.135
Supplies	gennaio 2007	21.554,17	13.541

## Formato delle interrogazioni

Le interrogazioni assumono solitamente la seguente forma standard



- sono comuni anche interrogazioni che effettuano confronti e/o rapporti

17

Il modello dimensionale

Luca Cabibbo

## Drill down

L'operazione di drill down aggiunge dettaglio ai dati restituiti da una interrogazione

- il drill down avviene aggiungendo un nuovo attributo nell'intestazione di una interrogazione
- diminuisce la grana dell'aggregazione – aumenta il dettaglio dei dati

(product) category	(time) month	(sum of) dollars_sold	(sum of) units_sold
-----------------------	-----------------	--------------------------	------------------------

↓  
drill down

(product) category	(time) month	(store) city	(sum of) dollars_sold	(sum of) units_sold
-----------------------	-----------------	-----------------	--------------------------	------------------------

18

Il modello dimensionale

Luca Cabibbo

## Drill up

L'operazione di drill up riduce il dettaglio dei dati restituiti da una interrogazione

- il drill up avviene rimuovendo un attributo dall'intestazione di una interrogazione
- aumenta la grana dell'aggregazione – diminuisce il dettaglio dei dati

(product) category	(time) month	(sum of) dollars_sold	(sum of) units_sold
-----------------------	-----------------	--------------------------	------------------------



drill up

(product) category	(sum of) dollars_sold	(sum of) units_sold
-----------------------	--------------------------	------------------------

## Discussione

Per il data warehouse, la modellazione dimensionale presenta dei vantaggi rispetto alla modellazione tradizionale (ER-BCNF) adottata nei sistemi operazionali

- gli schemi dimensionali hanno una forma standardizzata e prevedibile
  - è facilmente comprensibile e rende possibile la navigazione dei dati
  - semplifica la scrittura delle applicazioni
  - ha una strategia di esecuzione efficiente
- gli schemi dimensionali hanno una struttura simmetrica rispetto alle dimensioni
  - la progettazione può essere effettuata in modo indipendente per ciascuna dimensione
  - le interfacce utente e le strategie di esecuzione sono simmetriche

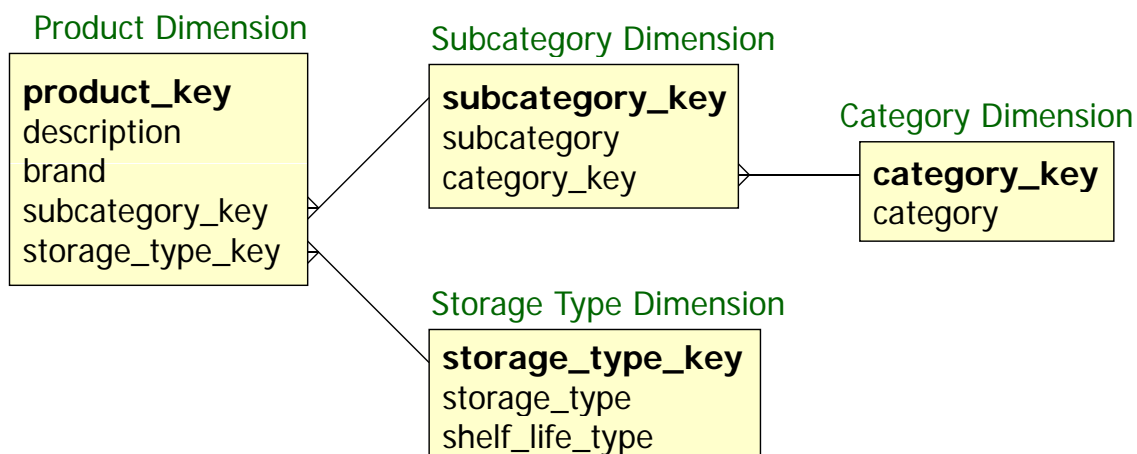
## Vantaggi della modellazione dimensionale

Ulteriori vantaggi, che vedremo successivamente

- gli schemi dimensionali sono facilmente estendibili
  - rispetto all'introduzione di nuovi fatti
  - rispetto all'introduzione di nuovi attributi per le dimensioni
  - rispetto all'introduzione di nuove dimensioni "supplementari"
    - se ogni record della tabella fatti dipende già funzionalmente dai membri della nuova dimensione
- si presta alla gestione e materializzazione di dati aggregati
- sono state già sviluppate numerose tecniche per la descrizione di tipologie fondamentali di fatti e dimensioni
  - ad esempio, dimensioni lentamente variabili, prodotti eterogenei, tabelle fatti senza fatti, ....
  - alcune di queste tecniche saranno presentate nel corso

## Snowflaking

Per snowflaking di una dimensione si intende una rappresentazione "più normalizzata" di una tabella dimensione, che evidenzia delle "gerarchie di attributi"



## Occupazione di memoria

Stima dell'occupazione di memoria della base di dati dimensionale di esempio

- tempo
  - 2 anni di 365 giorni, ovvero 730 giorni
- negozi
  - 300 negozi
- prodotti
  - 30.000 prodotti
- fatti relativi alle vendite
  - ipotizziamo un livello di sparsità del 10% delle vendite giornaliere dei prodotti nei negozi
    - ovvero, che ogni negozio vende giornalmente 3.000 diversi prodotti
  - $730 \times 300 \times 3000 = 630.000.000$  record

## Resistere allo snowflaking

Lo snowflaking è solitamente svantaggioso

- inutile per l'occupazione di memoria
  - ad es., una dimensione prodotto con 30.000 record, di circa 2.000 byte ciascuno -> 60MB di memoria primaria
  - tabella fatti con invece 630.000.000 record, di circa 10 byte ciascuno -> 6.3GB di memoria primaria
  - le tabelle fatti sono sempre molto più grandi delle tabelle dimensione associate
    - anche riducendo l'occupazione di memoria della dimensione prodotto del 100%, l'occupazione di memoria complessiva è ridotta di meno dell'1%
- può peggiorare le prestazioni
- tuttavia, ci sono delle situazioni in cui è utile definire delle "sottodimensioni" – con l'apparenza di uno snowflake
  - si veda la tecnica delle mini-dimensioni