Data Warehousing

1

Libro di riferimento

- R. Kimball. The data warehouse toolkit - Practical techniques for building dimensional data warehouses. John Wiley & Sons, Inc. 1996.

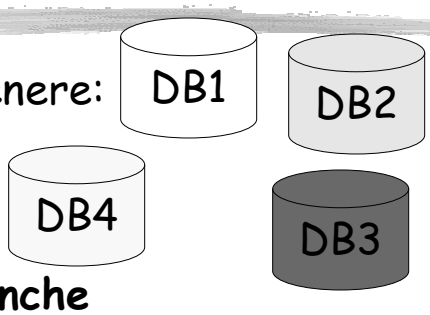
2

Introduzione al data warehousing

3

Il problema

In genere:



↑ abbondanza di dati

ma anche

↓ abbondanza di ridondanza ed inconsistenza che non permette di utilizzare i dati in modo utile a fini decisionali

4

Tipiche richieste

- Qual è il volume delle vendite per regione e categorie di prodotto durante l'ultimo anno?
- Come si correlano i prezzi delle azioni delle società produttrici di hardware con i profitti trimestrali degli ultimi 10 anni?
- Quali sono stati i volumi di vendita dello scorso anno per regione e categoria di prodotto?
- In che modo i dividendi di aziende di hardware sono correlatiai porfitti trimestrali negli ultimi 10 anni?

5

Possibili applicazioni

contesti →

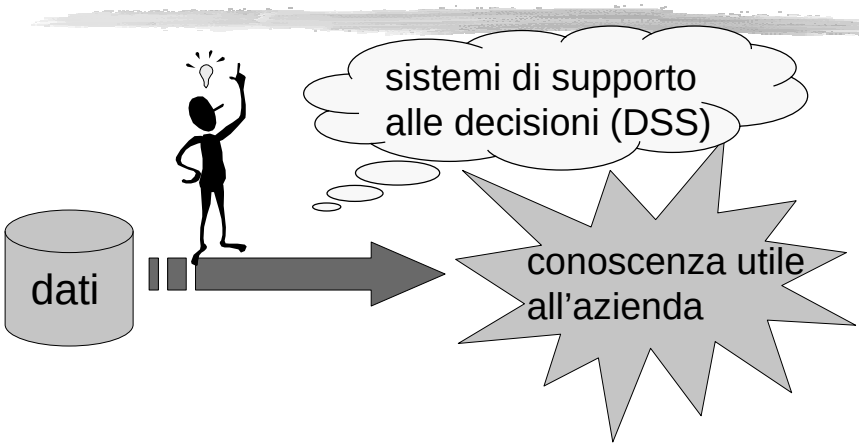
- •gestione dei rischi
- •analisi finanziaria
- •programmi di marketing
- •analisi statistica
- •integrazione DB clienti
- •integrazione relazioni clienti
- •analisi temporale

- •telecomunicazioni
- •banking
- •università
- •assicurazioni
- •beni di consumo
- •salute
- •produzione

← **problematiche**

6

In sintesi ...



DSS: Tecnologia che supporta la dirigenza aziendale nel prendere decisioni tattico-strategiche in modo migliore e più veloce

7

Perché i sistemi tradizionali non sono sufficienti?

- no dati storici
- sistemi eterogenei
- basse prestazioni
- DBMS non adeguati al supporto decisionale
- problemi di sicurezza

8

Più formalmente ...

- Sistemi tradizionali
  - On-Line Transaction Processing (OLTP)
- Sistemi di data warehousing
  - On-Line Analytical Processing (OLAP)

⇒ Profondamente diversi

9

In dettaglio ...

	OLTP	OLAP
funzione	gestione giornaliera	supporto alle decisioni
progettazione	orientata alle applicazioni	orientata al soggetto
frequenza	gironaliera	sporadica
dati	recenti, dettagliati	storici, riassuntivi, multidimensionali
sorgente	singola DB	DB multiple
uso	ripetitivo	ad hoc
accesso	read/write	read
flessibilità accesso	uso di programmi precompilati	generatori di query
# record acceduti	decine	migliaia
tipo utenti	operatori	manager
# utenti	migliaia	centinaia
tipo DB	singola	multiple, eterogenee
performance	alta	bassa
dimensione DB	100 MB - GB	100 GB - TB

10

Evoluzione dei DSS

- Anni '60: rapporti batch
  - difficile trovare ed analizzare i dati
  - costo, ogni richiesta richiede un nuovo programma
- Anni '70: DSS basato su terminale
  - non integrato con strumenti di automazione d'ufficio
- Anni '80: strumento d'automazione d'ufficio
  - strumenti di interrogazione, fogli elettronici, interfacce grafiche
  - accesso ai dati operazionali
- Anni '90: data warehousing, con strumenti integrati OLAP

11

I sistemi di data warehousing

- Il Data Warehousing si può definire come il processo di integrazione di basi di dati indipendenti in un singolo repository (il data warehouse) dal quale gli utenti finali possano facilmente ed efficientemente eseguire query, generare report ed effettuare analisi

12

Il data warehouse

Collezione di dati che soddisfa le seguenti proprietà:

- usata per il supporto alle decisioni
- orientata ai soggetti
- integrata: livello aziendale e non dipartimentale
- correlata alla variabile tempo: ampio orizzonte temporale
- con dati tipicamente aggregati: per effettuare stime
- fuori linea: dati aggiornati periodicamente

13

Il data warehouse

- **Orientata ai soggetti**: considera i dati di interesse ai soggetti dell'organizzazione e non quelli rilevanti ai processi organizzativi
  - basi di dati operazionali dipartimentali:
    - vendita, produzione, marketing
  - data warehouse: prodotti, clienti, fornitori

14

Il data warehouse

- **Integrata**:
  - i dati provengono da tutte le sorgenti informative
  - il data warehouse rappresenta i dati in modo univoco, riconciliando le eterogeneita` delle diverse rappresentazioni:
    - nomi
    - struttura
    - codifica
    - rappresentazione multipla

15

Il data warehouse

- **Correlata alla variabile tempo**: presenza di dati storici per eseguire confronti, previsioni e per individuare tendenze
  - basi di dati operazionali: finestra temporale di pochi mesi
  - daat warehouse: finestra temporale dell'ordine di anni

16

Il data warehouse

- **Dati aggregati**: nell'attività di analisi dei dati per il supporto alle decisioni:
  - non interessa "chi" ma "quanti"
  - non interessa un dato ma la somma, la media, il minimo, il massimo di un insieme di dati

17

Il data warehouse

- **Fuori linea**:
  - base di dati operazionale: i dati venono acceduti, inseriti, modificati, cancellati pochi record alla volta
  - data warehouse:
    - operazioni di accesso e interrogazione diurne
    - operazioni di caricamento e aggiornamento notturne

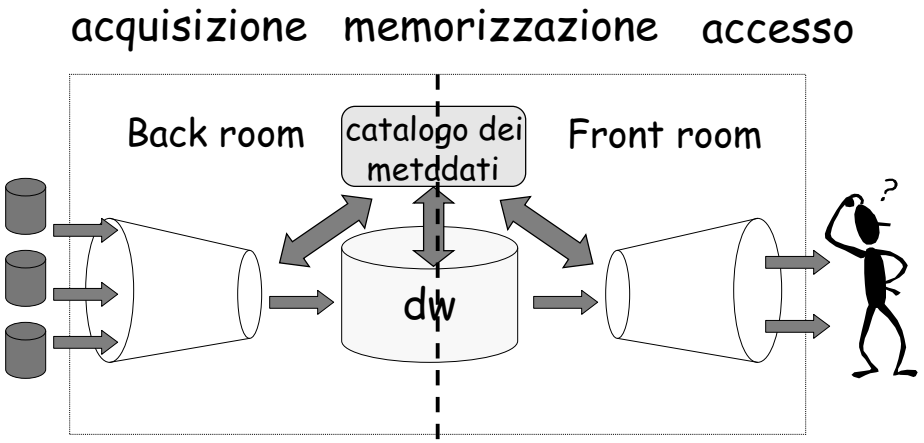
    che riguardano milioni di record

18

Architettura di riferimento

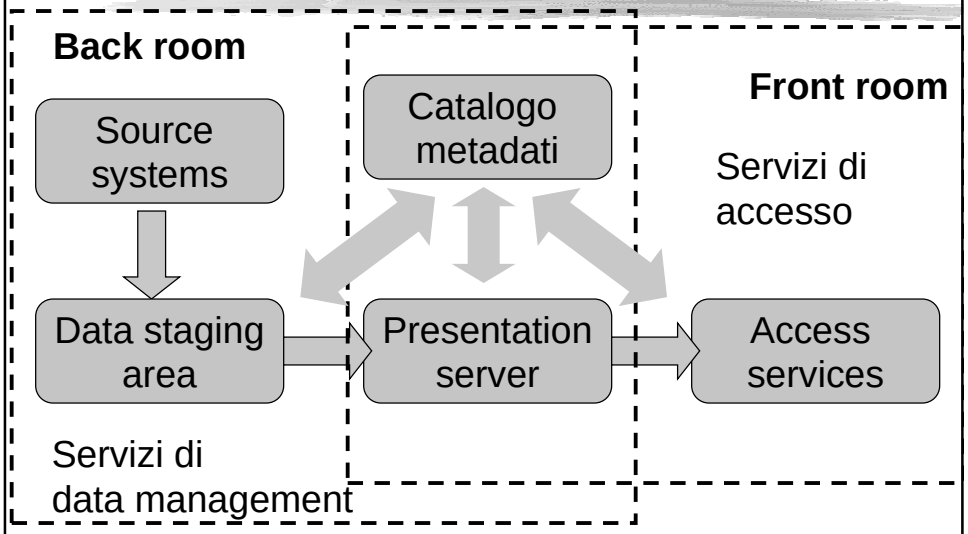
19

Architettura di base



20

Piu' in dettaglio ...



21

Source systems

- Ogni sorgente di informazioni aziendali
- Spesso rappresentate da dati operazionali: insieme di record la cui funzione è quella di catturare le transazioni del sistema organizzativo
- tipico accesso OLTP
- uso di production keys (non vengono usate nel DW)

22

Data staging

- Area di memorizzazione
  - i dati sorgente vengono trasformati
  - tecnologia relazionale ma anche flat files
- insieme di processi che:
  - puliscono, trasformano, combinano, duplicano, archiviano e preparano i dati sorgente per essere usati nel DW

23

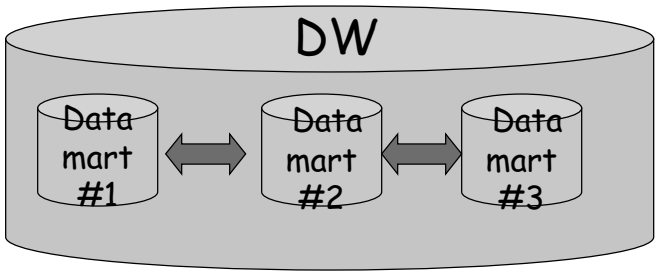
Presentation server

- Componente che permette la memorizzazione e la gestione del data warehouse, secondo un approccio dimensionale
- Può essere basato su:
  - tecnologia relazionale (ROLAP)
  - tecnologia multidimensionale (MOLAP)

24

Presentation server

- Un DW rappresenta spesso l'unione di più data mart
- **Data mart**: restrizione data warehouse ad un singolo processo o ad un gruppo di processi aziendali (es. Marketing)



25

End-user data access tools

- Client del DW, di facile utilizzo
- tools per interrogare, analizzare e presentare l'informazione contenuta del DW a supporto di un particolare bisogno aziendale
- invio specifiche richieste al presentation server in formato SQL

26

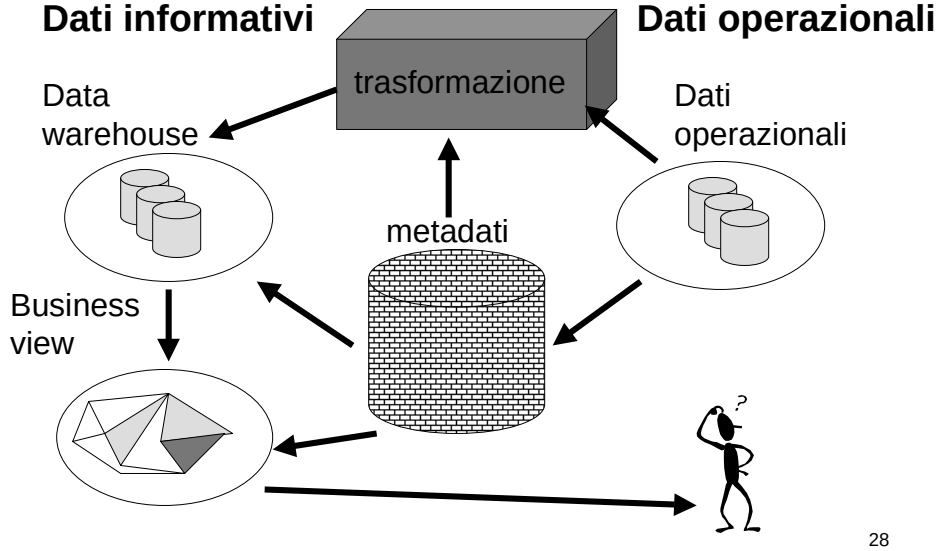
I metadati

- = dati sui dati
- Link tra i DB operazionali e il DW
- ogni passo eseguito durante la costruzione del DW genera metadati che possono poi essere utilizzati dalle fasi successive

- **Esempi**: schema, data in cui un dato è stato creato, quale tool l'ha creato, storia delle trasformazioni di un dato nel tempo, statistiche, dimensione tabelle, ecc. ecc.

27

I metadati



28

Due ritmi diversi ...

- Uso bimodale:
  - 16-22 ore al giorno usati per attività di interrogazione
    - funzionalità front room
  - 2-8 ore al giorno per caricamento, indicizzazione, controllo qualità e pubblicazione
    - funzionalità back room

29

Progettazione concettuale di un data warehouse

30

Le fasi della progettazione

- Progettazione concettuale:
  - Fornisce una rappresentazione formale del contenuto informativo del DW
  - indipendente dal sistema che verrà utilizzato per la sua implementazione
- progettazione logica:
  - Lo schema concettuale viene tradotto nel modello dei dati del sistema prescelto
- progettazione fisica:
  - Fase in vui vengono scelte le caratteristiche fisiche del sistema

31

Progettazione concettuale

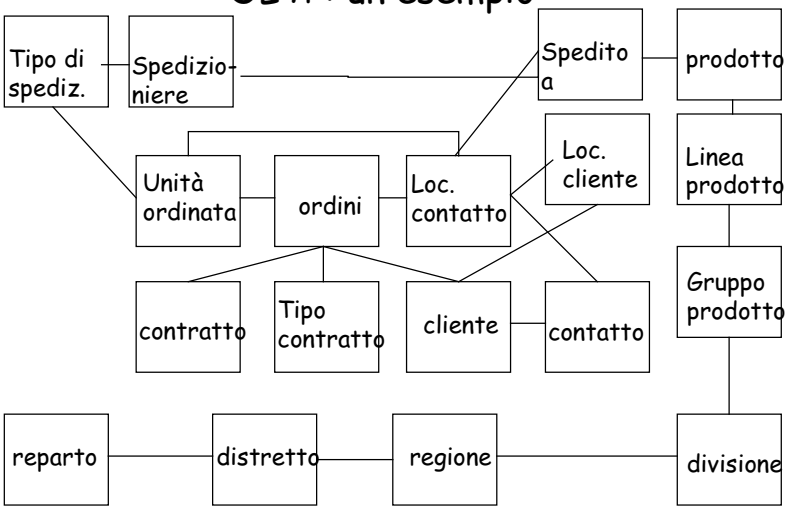
OLTP

- modello entità-relazione
- si cerca di eliminare il più possibile la ridondanza
  - maggiore efficienza delle operazioni di aggiornamento
- schema simmetrico: tutte le entità hanno la stessa importanza
- ci possono essere molti modi per connettere (mediante un'operazione di join) due tabelle
- la rappresentazione dipende dalla struttura dei dati

32

Progettazione concettuale

OLTP: un esempio

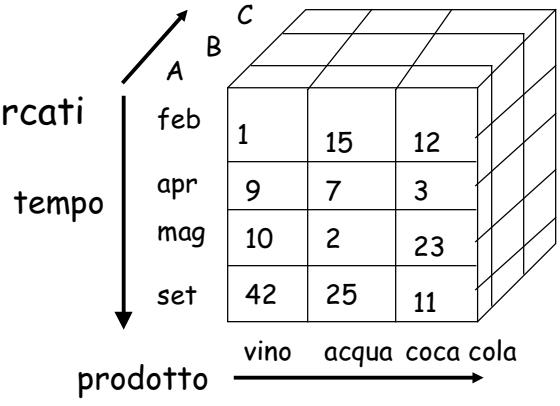


33

Progettazione concettuale

OLAP

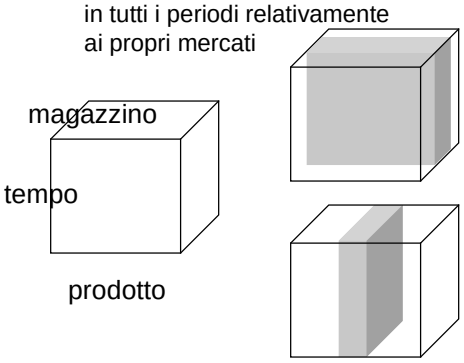
Processo:
vendite in una
catena di supermercati



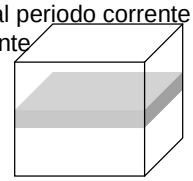
34

Progettazione concettuale

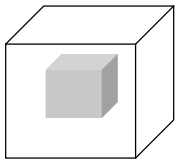
Il manager regionale esamina la vendita dei prodotti in tutti i periodi relativamente ai propri mercati



Il manager finanziario esamina la vendita dei prodotti in tutti i mercati relativamente al periodo corrente e quello precedente



Il manager di prodotto esamina la vendita di un prodotto in tutti i periodo e in tutti i mercati



Il manager strategico si concentra su una categoria di prodotti, un'area regionale e un orizzonte temporale medio

35

Progettazione concettuale

OLAP

- Ogni parametro puo` essere organizzato in una gerarchia che ne rappresenta i possibili livelli di aggregazione:
  - negozio, citta`, provincia, regione
  - giorno, mese, trimestre, anno

36

Progettazione concettuale

OLAP

- L'eliminazione della ridondanza non è un obiettivo
  - non si devono eseguire operazioni di aggiornamento
  - schemi denormalizzati
- schemi asimmetrici: alcune entità sono più importanti di altre
- un solo modo per connettere (mediante un'operazione di join) due tabelle
  - minore numero di join
  - maggiore efficienza
- la rappresentazione dipende dalla struttura dei dati

37

Progettazione concettuale

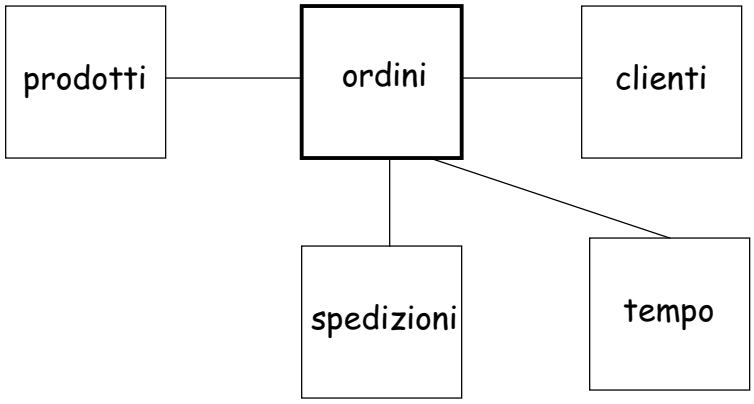
I modelli multidimensionali

- Vari modelli multidimensionali:
  - modello a stella
  - modello a costellazione di fatti
  - modello a fiocco di neve (snowflake)
- possono essere implementati in
  - sistemi relazionali
  - sistemi multidimensionali
  - sistemi ad oggetti
- In genere: implementazione diretta in DBMS relazionali

38

Progettazione concettuale

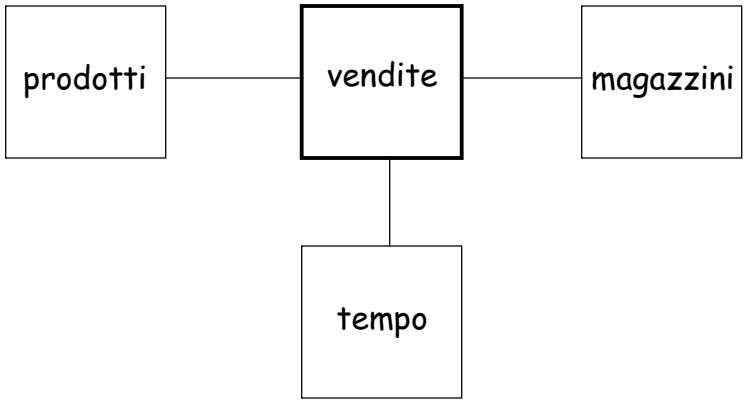
OLAP: un esempio



39

Progettazione concettuale

OLAP: un esempio



40

Il modello a stella

- Nel modello dimensionale, non sono rilevanti i singoli eventi (transazioni) ma il loro accadere durante un determinato intervallo temporale
  $\Rightarrow$ granularità dello schema
- Il modello a stella è basato sull'esigenza di vedere dati di dettaglio (fatti) in funzione di più dimensioni

41

Il modello a stella

- **Fatti**: identificano l'attività principale e sono caratterizzati dai dati di dettaglio che si desidera analizzare
- **Dimensioni**: parametri che influenzano i dati di dettaglio e rispetto ai quali si analizzano tali dati
- Fatti e dimensioni collegati attraverso chiavi esterne
  - in generale, uno schema a stella rappresenta una relazione molti a molti
    - il collegamento tra ogni tabella delle dimensioni e la tabella dei fatti rappresenta una relazione uno a molti

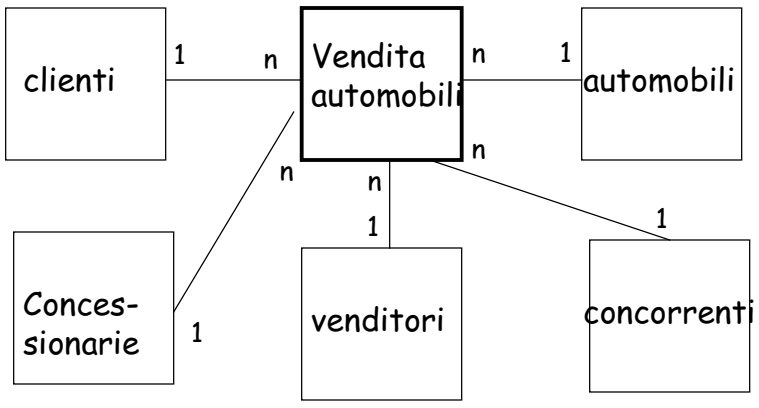
42

Esempio

- Attività principale: vendita automobili
- dimensioni:
  - clienti
  - venditori
  - concorrenti
  - automobili
  - concessionarie

43

Esempio



44

Le dimensioni

- Devono essere scelte solo le entità rilevanti per le analisi che si intendono effettuare
- Le dimensioni sono tipicamente caratterizzate da attributi:
  - testuali
  - discreti

  ma possono anche essere numeriche
  - dimensione di un prodotto
- esiste sempre una dimensione temporale

45

Dimensioni: esempi

- Attività: vendita in una catena di supermercati
  - dimensioni: tempo, prodotti, magazzino
- Attività: ordini
  - dimensioni: tempo, prodotti, clienti, spedizioni
- Attività: iscrizioni universitarie
  - dimensioni: tempo, facoltà, tipologia studenti
- Attività : vendita automobili
  - dimensioni: clienti, venditori, concorrenti, automobili, concessionarie

46

Le dimensioni

- Problema: come si può identificare se un attributo numerico è un fatto o un attributo di una dimensione?
- Se è una misura che varia continuamente nel tempo
  - fatto
    - analisi costo di un prodotto nel tempo
- se è una descrizione discreta di qualcosa che è ragionevolmente costante
  - attributo di una dimensione
    - costo di un prodotto visto come informazione descrittiva

47

Le dimensioni

- Le dimensioni utilizzate sono spesso le stesse in vari contesti applicativi:
  - tempo
  - collocazione geografica
  - organizzazione
  - clienti
- il numero di attributi per ogni dimensione è in genere molto elevato (anche nell'ordine del centinaio)

48

La dimensione tempo

- È presente in ogni DW in quanto virtualmente ogni DW rappresenta una serie temporale
- **Domanda**: perché non campo di tipo DATE nella tabella dei fatti?
- **Risposta**: la dimensione tempo permette di descrivere il tempo in modi diversi da quelli che si possono desumere da un campo date in SQL (giorni lavorativi-vacanze, periodi fiscali, stagioni, ecc.)

49

La dimensione tempo

- Alcuni tipici attributi della dimensione tempo:
  - tempo-k (può essere un campo di tipo data in SQL)
  - giorno-della-settimana
  - n-giorno-nel-mese
  - n-giorno-in-anno
  - n-settimana-in-anno
  - mese
  - stagione
  - periodo fiscale
  - ...

50

I fatti

- La tabella dei fatti mette in evidenza una relazione molti-a-molti
- I fatti sono tipicamente:
  - numerici
  - addittivi
- Numerici, addittivi: possono essere aggregati rispetto agli attributi delle dimensioni, utilizzando l'operazione di addizione

51

I fatti: esempi

- Attività: vendita in una catena di supermercati
  - fatti: n. prodotti venduti, incassi, costi, ...
- Attività: ordini
  - fatti: n. spedizioni, n. clienti, importi, ...
- Attività: iscrizioni universitarie
  - fatti: n. studenti, ...

52

Addittività dei fatti

- Incasso, unità vendute: sono addittivi in quanto si possono aggregare sommando rispetto ad ogni dimensione:
  - somma incassi/unità su tempo
  - somma incassi/unità su prodotti
  - somma incassi/unità su dipartimenti

53

Semiaddittività dei fatti

- Numero clienti non è un fatto addittivo:
  - somma n. clienti su tempo OK
  - somma n. clienti su dipartimenti OK

  MA:
  - somma n. clienti su prodotto genera problemi
  - si supponga che
    - clienti che hanno comprato carne 20
    - clienti che hanno comprato pesce 30
  - il numero di clienti che hanno comprato carne o pesce è un qualunque numero tra 30 e 50

54

Semiaddittività dei fatti

- Il numero clienti è un fatto semiaddittivo, poiché può essere sommato solo rispetto ad alcune dimensioni
- Soluzione: cambiare la granularità del database, portandola a livello singola transazione

55

Semiaddittività dei fatti

- Tutte le misure che memorizzano una informazione statica, quali:
  - bilanci finanziari
  - misure di intensità (temperatura di una stanza)

  sono semiaddittivi rispetto al tempo
- ciò che comunque si può fare è calcolare la media su un certo periodo di tempo

56

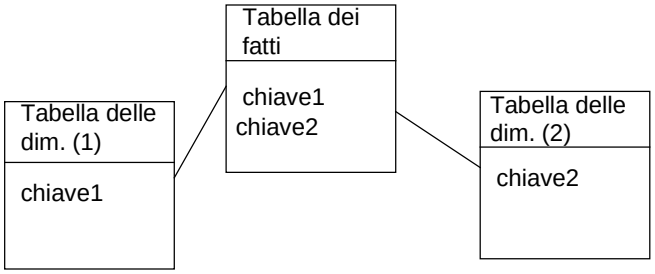
Non addittività dei fatti

- I fatti non addittivi sono fatti che non possono essere sommati
- Esempi:
  - fatti: costo unitario e quantità nel contesto di un ordine
  - dimensioni: clienti, spedizioni, tempo, …
  - i costi unitari non possono essere sommati se prima non sono moltiplicati per le rispettive quantità, quindi tali costi sono fatti non addittivi

57

Collegamenti tra dimensioni e fatti

- Avviene tramite chiavi esterne
- ogni tabella delle dimensioni ha una chiave
- la tabella dei fatti deve contenere come attributi la chiave di ciascuna dimensione
- tali attributi sono chiavi esterne e, complessivamente, rappresentano la chiave della tabella dei fatti



58

Vantaggi nell'uso dello schema a stella

- Permette di fare assunzioni circa i dati, da utilizzare in fase di ottimizzazione
- simmetrico
- facilmente estensibile
- approcci standard alla costruzione
- tipiche query eseguibili efficientemente (vedi oltre)

59

Metodologia di progettazione

- La progettazione dipende da:
  - requisiti utente
    - si determinano attraverso interviste con gli utenti finali
  - dati disponibili
    - si determinano analizzando la documentazione esistente e attraverso interviste con i DBA

60

Passi nel processo di progettazione

- Scegliere il processo aziendale che si intende modellare
- scegliere la granularità con la quale si intende modellare il processo aziendale
- scegliere le dimensioni e i loro attributi
- scegliere i fatti

61

Un esempio per capire

La gestione di una catena di supermercati

- 500 supermercati distribuiti su un'area che comprende 3 stati negli USA
- ogni supermercato è composto da diversi reparti
- ogni reparto vende molti prodotti, identificati da stock keeping unit (SKU)
- i dati delle vendite vengono raccolti nei punti di vendita (casse)
- i prodotti sono spesso soggetti a promozioni di vendita
- problematiche che si intendono analizzare:logistica degli ordini, massimizzazione profitti in ciascun supermercato

62

Passo 1: scelta del processo aziendale

- Si deve decidere quale processo modellare, combinando la conoscenza aziendale con la conoscenza di quali dati sono disponibili
- Esempio

  movimento giornaliero delle varie unità

63

Passo 2: scelta della granularità

- La granularità identifica il contenuto della tabella dei fatti nel processo considerato
- è importante perché :
  - determina le dimensioni del database
  - condiziona la dimensione del database
- Esempio

  SKU per magazzino per promozione per giorno (granularità a livello di giorno e di singola unità)

64

Passo 2: scelta della granularità

- Perché giornaliero:
  - granularità a livello transazione (operazione di acquisto)
    - il database diventerebbe enorme e quindi ingestibile
  - granularità a livello settimana o mese:
    - molti effetti delle vendite non sarebbero visibili
    - Ad esempio: differenza in vendite tra Lunedì e Sabato

65

Passo 2: scelta della granularità

- Perché a livello singola unità:
  - granularità a livello pacco:
    - non sarebbe più possibile rispondere a domande quali:
      - le vendite di quali prodotti si riducono quando un certo prodotto viene messo in promozione di vendita?
      - se confrontiamo le vendite con quelle della concorrenza, quali sono i 10 prodotti che la concorrenza vende e noi non vendiamo?

66

Passo 3: scelta delle dimensioni

- Una definizione accurata della granularità comporta immediatamente la definizione delle dimensioni principali del DW
  (dimensioni primarie)

- è quindi possibile aggiungere altre dimensioni, purchè queste dimensioni assumano un singolo valore per ogni combinazione delle dimensioni primarie

67

Passo 3: scelta delle dimensioni

- Esempio:
  nel nostro esempio, le dimensioni primarie sono:
  - tempo
  - prodotti
  - magazzini

  dimensioni aggiuntive
  - promozioni

68

Passo 3: scelta delle dimensioni

- Per ciascuna dimensione, devono essere specificati gli attributi che la caratterizzano
- spesso si tratta di attributi alfanumerici
- se una dimensione contiene attributi non correlati, è meglio suddividere la dimensione in due dimensioni distinte

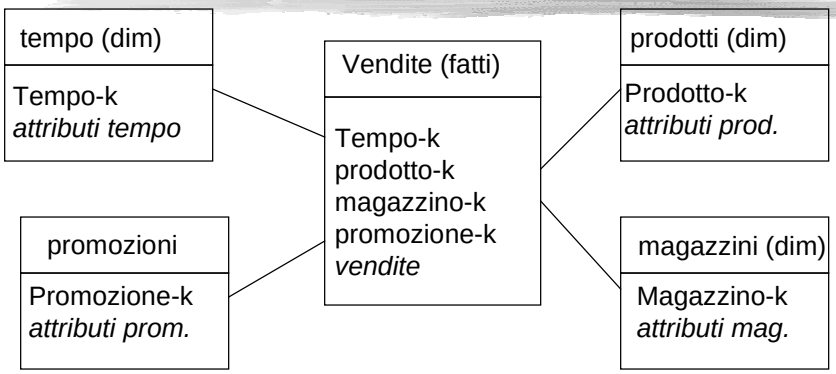
69

Passo 4: scelta dei fatti

- Fatti tipici sono numerici e addittivi
- Esempio: alcuni utili fatti misurabili:
  - incasso
  - unità vendute
  - numero clienti
- nel caso di granularità transazionale, l'unico fatto in genere rilevante è una quantità (livello di granularità più fine)

70

Schema iniziale per l'esempio considerato



Per il momento: assumiamo che lo schema precedente rappresenti sia lo schema logico che lo schema fisico del database, quindi il database contiene 5 tabelle

71

Osservazioni sulla normalizzazione

- La tabella dei fatti è completamente normalizzata
- le tabelle delle dimensioni possono non essere normalizzate, ma:
  - la dimensione delle tabelle delle dimensioni è in genere irrilevante rispetto alla dimensione della tabella dei fatti
  - quindi, ogni sforzo per normalizzare queste tabelle ai fini del DW è una perdita di tempo
  - lo spazio guadagnato è in genere meno dell'1% dello spazio richiesto dallo schema complessivo
- la normalizzazione delle tabelle delle dimensioni può ridurre la capacità di browsing (navigazione) dello schema (si veda oltre)

72

Gerarchie

- Molto spesso una singola dimensione può contenere dati organizzati gerarchicamente
- Esempio: Ogni unità può essere rappresentata all'interno di una gerarchia:
  - sku
  - pacco
  - marca
  - sottocategoria
  - categoria
  - dipartimento

73

Gerarchie

- Tutti gli attributi della gerarchia devono essere inseriti all'interno della dimensione

⇒ ridondanza accettabile in quanto la dimensione delle tabelle delle dimensioni in genere è inifluente sulla dimensione totale del database

- Esempio: 30000 prodotti distinti
  30 dipartimenti distinti
  ⇒ in media 1000 ripetizioni

74

Esempio

Prodotti (dim)
Prodotto-k descrizione-SKU numero-SKU tipo-pacco marca sottocategoria categoria dipartimento tipo-pacco peso unità-di-misura ...

• La tabella dei prodotti è una delle tabelle principali di quasi tutti i DW

• è utile inserire più attributi possibile

75

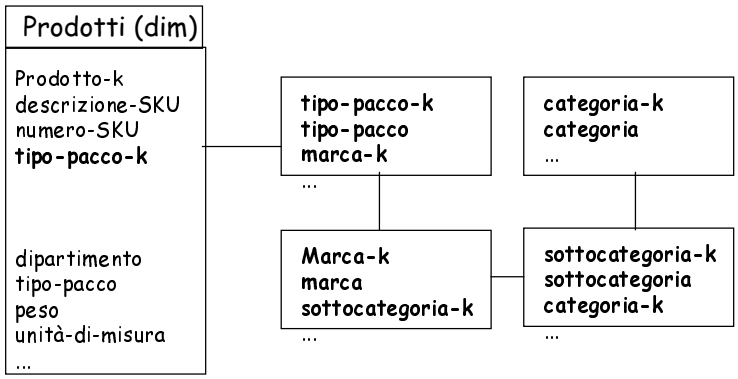
Gerarchie

- Gerarchia tipica: gerarchia geografica:
  - comune
  - provincia
  - regione
  - stato
  - continente
- una dimensione può contenere attributi relative a gerarchie multiple

76

Schemi snowflake

- Una gerarchia rappresenta molte relazioni molti-a-uno
  - si potrebbe pensare di utilizzare una tabella per ogni relazione ⇒ schema snowflake



77

Schemi snowflake

- Uno schema snowflake rende meno efficienti le operazioni di ricerca, anche se la tabella è grande (+ join)
- è conveniente utilizzare uno schema snowflake solo se questo approccio aumenta la leggibilità dello schema e le prestazioni globali

78

Esempio

- Dim. tabella dei fatti: 30 GB
- dim. indice tabella dei fatti 20 GB
- dim. max tabella delle dim. 0.1 GB
- usando schema snowflake 0.005 GB
- dim DB senza snowflake 50 GB
- dim DB con snowflake 50 GB

79

Progettazione concettuale avanzata di un data warehouse

80

Composizione degli schemi

- Lo schema risultante da ogni processo aziendale può essere visto come lo schema associato ad uno specifico data mart
- problema: combinare i fatti e le dimensioni contenuti negli schemi associati a ciascun processo, cioe' contenuti in ciascun data mart

81

Composizione degli schemi

- Gli schemi associati ai vari processi possono avere dimensioni a comune
  - Una singola tabella delle dimensioni puo` essere usata in relazione a diverse tabelle dei fatti
- per potere passare dalle informazioni contenute in uno schema alle informazioni contenute in un altro (drill-across): le dimensioni con lo stesso nome devono avere lo stesso significato e contenere gli stessi attributi
  - dimensioni conformate
- Conseguenza: i vincoli su attributi delle dimensioni a comune devono restituire le stesse entità per ogni schema considerato

82

Esempio: catena di produzione

- inventario dei prodotti
  - dimensioni: tempo, prodotti, warehouse
- spedizione ai centri di distribuzione
  - dimensioni: tempo, prodotti, warehouse, centri di distribuzione, contratti, tipi di spedizione
- inventario del centro di distribuzione
  - dimensioni: tempo, prodotti, centri di distribuzione
- distribuzione ai magazzini
  - dimensioni: tempo, prodotti, centri di distribuzione, magazzini, contratti, tipi di spedizione
- inventario dei magazzini
  - dimensioni: tempo, prodotti, magazzini
- vendite
  - dimensioni: tempo, prodotti, magazzini, promozioni, clienti

83

Composizione degli schemi: eccezione

- Eccezione: la stessa dimensione può comparire in schemi diversi con un sottoinsieme di attributi (diversa conoscenza di un particolare aspetto applicativo)
  - drill-across si può fare solo sugli attributi in comune
- Esempio: i produttori conoscono i prodotti ad un livello di dettaglio maggiore rispetto a quello noto ai venditori, ma il tipo di prodotto comparira` in entrambe le dimensioni

84

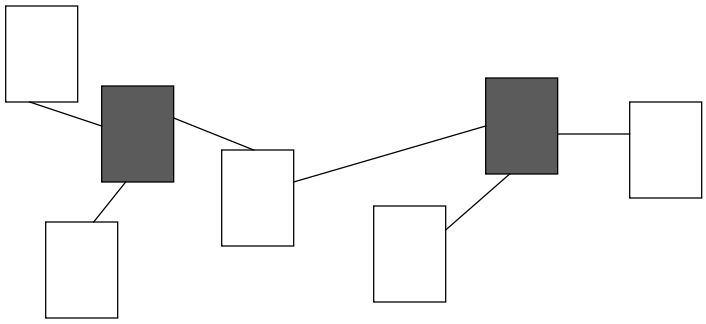
Fatti conformati

- Anche i fatti devono essere conformati
  - fatti con lo stesso nome in tabelle diverse hanno la stessa granularita` e le stesse unita` di misura
  - stesso periodo temporale
  - stesso riferimento geografico

85

Costellazione di fatti

- Schema risultante:
  - costellazione di fatti



86

Aggregazione

- In alcune situazioni, non si hanno vincoli su tutte le dimensioni ma solo per alcune
- Esempio:
  - qual'e` il rapporto tra vendite effettuate nei week-end e vendite effettuate nei giorni lavorativi in ogni magazzino?
  - Quale prodotto e` stato maggiormente venduto negli ultimi 3 mesi?
- L'esecuzione di queste interrogazioni e` molto costosa se viene effettuata sui dati di base
  - Idea: **precalcolare aggregati**

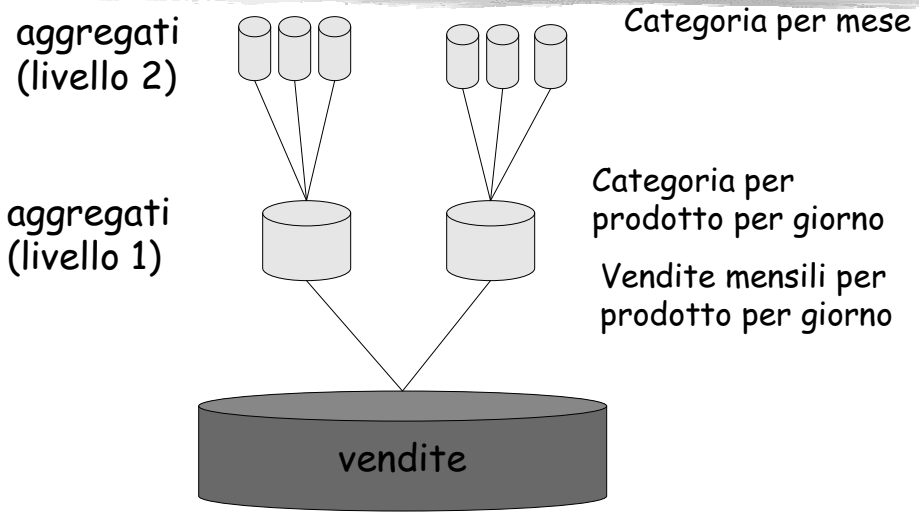
87

Aggregazione

- Un aggregato e` un record di una tabella dei fatti che rappresenta una sintesi di vari record contenuti nella tabella dei fatti di base
- una tabella dei fatti aggregata e` sempre associata ad una o piu` tabelle delle dimensioni aggregate
- un aggregato viene utilizzato per due motivi:
  - efficienza
  - impossibilita` di rappresentare gli stessi dati al livello di dettaglio
  - Esempio: costi di promozione possono essere espressi a livello categoria e non a livello di singolo prodotto

88

Esempio



89

Due problemi

- Quali dati aggregare?
- Come e dove memorizzare i dati aggregati?

90

Quali dati aggregare?

- È importante considerare:
  - **tipiche richieste aziendali**
    - distribuzione geografica, linee di prodotti, periodicità generazione reportistica
    - per ogni dimensione, identificare gli attributi e le combinazioni di attributi che può essere utile aggregare
  - **distribuzione statistica dei dati**
    - stimare la dimensione delle tabelle aggregate
    - se la dimensione della tabella aggregata non riduce di molto la dimensione della tabella di partenza, forse non conviene aggregare
    - aggregazioni non molto usate possono essere utili come punto di partenza per effettuare altre aggregazioni più significative

91

Come e dove memorizzare i dati aggregati?

- Esistono due approcci di base:
  - nuove tabelle dei fatti
    - vengono create nuove tabelle per i fatti e le dimensioni aggregate
  - nuovo campo Livello
    - vengono aggiunti nuovi campi nelle tabelle dei fatti e delle dimensioni
- Vedremo solo il primo approccio

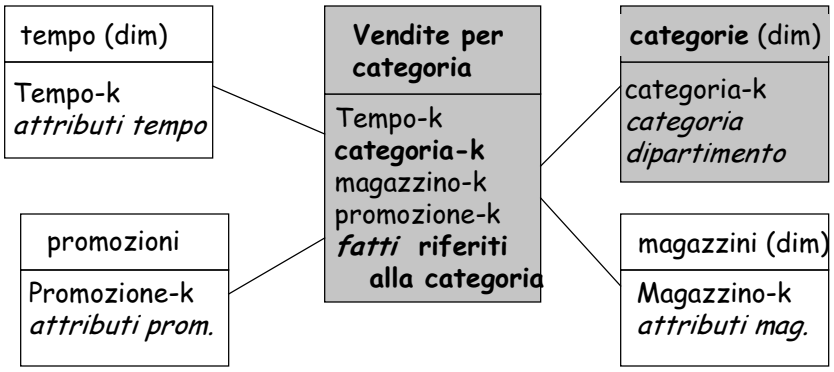
92

Nuove tabelle dei fatti

- Per ogni aggregato di interesse viene generata una nuova tabella dei fatti
- si generano tabelle delle dimensioni derivate da quelle di base ma contenenti solo i dati di interesse per la tabella dei fatti aggregata
  - generazione di chiavi artificiali per le tabelle delle dimensioni aggregate

93

Esempio



94

Nuove tabelle dei fatti

- L'uso di tabelle dei fatti e delle dimensioni aggregate accelera anche l'esecuzione di interrogazioni rispetto ad attributi che generalizzano (in base ad opportune gerarchie) l'attributo aggregato
- Esempio:
  - interrogazioni sui dipartimenti partendo dagli aggregati di categoria
- rispetto a questi attributi, si può evitare di costruire tabelle aggregate ad hoc

95

Vantaggi e svantaggi nell'uso degli aggregati

- Svantaggi:
  - L'uso degli aggregati aumenta di molto la dimensione del DB (anche del 300%!)
  - usare aggregazione nel caso in cui ogni aggregato sintetizza almeno 10-20 record di base
- Vantaggi:
  - Miglioramento delle prestazioni
  - possono essere utilizzati in modo trasparente all'utente

96

Granularita` transazionale e snapshot periodico

- Nel caso di granularita` transazionale, potrebbe capitare di avere anche bisogno di informazioni sintetiche periodiche, ad esempio mensili

Tabella dei fatti transazionale → Dati operazionali base per **data mining**

+

tabella dei fatti periodica precalcolata → Base per analisi a piu` alto livello

97

Progettazione logica di un data warehouse

98

Scelta sistema di gestione dei dati

- DBMS operazionale: in genere relazionale
- DBMS informativo:
  - relazionale (Oracle 8/8i, RedBrick- Informix,…) (ROLAP)
  - multidimensionale (Oracle Express Server) (MOLAP)

99

ROLAP & MOLAP

- ROLAP:
  - sistema di data warehouse in grado di supportare le interrogazioni tipiche
  - presentation server relazionale
    - Oracle 8i + Discoverer
- MOLAP:
  - sistema di data warehouse in grado di supportare le interrogazioni tipiche
  - presentation server multidimensionale
    - Express Server
- DOLAP (Desktop OLAP):
  - i dati vengono recuperati da un DW relazionale o multidimensionale e copiati localmente
    - Business Objects

100

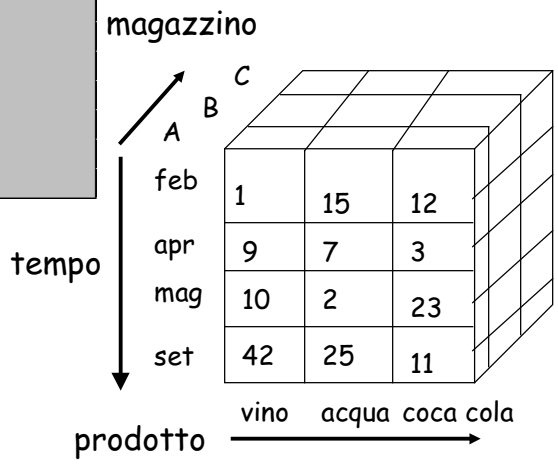
DBMS relazionali

- Tecnologia consolidata
- molto efficienti su dati di dettaglio
- estesi in modo da permettere la materializzazione degli aggregati
  - (Oracle 8i)
- performance
- scalabilità
- general-purposes

101

DBMS multidimensionali

vendite	prodotto	mese	magazzino
1	vino	febbraio	A
2	acqua	febbraio	B
3	coca cola	aprile	A
4	acqua	maggio	A
5	acqua	settembre	C
…	…	…	…



102

DBMS multidimensionali

- Modello dei dati basato su hypercubi (array multidimensionali)
- precalcolo aggregazioni
- aumento prestazioni per le query utente ma
  - … no join
  - … no interfaccia SQL (API)
  - … necessità sistema relazionale per dati dettaglio
  - … file molto grandi
  - … limitazioni a circa 10GB 8problemi scalabilità)
- Per superare questi problemi:
  - aggiunta capacità di navigare da un MDBMS ad un RDBMS

103

ROLAP & MOLAP

- Performance
  - Query: MOLAP
  - Caricamento: ROLAP
- Analisi: MOLAP
- Dimensione DW: ROLAP
  - MOLAP: problema sparsità
- Flessibilità nello schema: ROLAP
  - MOLAP: minor numero di dimensioni ammesse

104

Progettazione logica

- Durante questa fase, lo schema concettuale del DW viene tradotto in uno schema logico, implementabile sullo strumento scelto
- Il modello logico deve essere il più possibile vicino al modello concettuale, anche se alcune variazioni possono essere rese necessarie dal particolare tool prescelto
- supponiamo che il sistema prescelto sia ROLAP
  - tabella ⟹ relazione

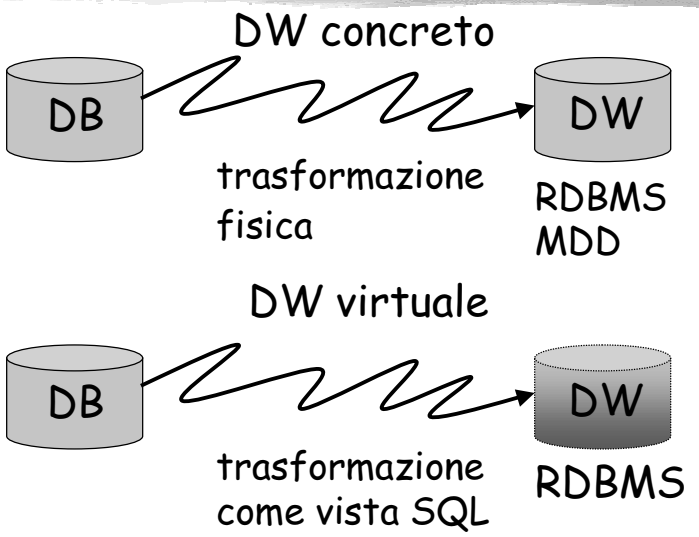
105

Progettazione logica

- Un'eccezione alle regole precedenti è dato dall'eventuale uso di views
- View per creare un DW a stella partendo da un DB normalizzato (basato su un generico schema ER)!
- Questo è utile ed efficiente solo su DW di piccole dimensioni, in cui l'accesso dimensionale è limitato

106

Progettazione logica



107

Progettazione fisica di un data warehouse

108

Problematiche

- Durante questa fase si definiscono le strutture di memorizzazione e indicizzazione da utilizzare per l'implementazione del DW
- Aspetti principali:
  - **aggregazione**
  - indici
  - parallelizzazione
  - partizionamento
  - **materializzazione query**
  - ottimizzazione query

109

Influenza aggregati sul codice SQL

- Se gli aggregati sono presenti, per poterli utilizzare bisogna ovviamente scrivere codice SQL opportuno
- partendo da una query sulle tabelle di base, le tabelle aggregate possono essere utilizzate sostituendole alle corrispondenti tabelle di base

110

Esempio query di base

Query sullo schema di base

```
SELECT categoria, SUM(vendite)
FROM vendite, prodotti, magazzini, tempo
WHERE vendite.prodotto-k = prodotti.prodotto-k AND
   vendite.magazzino-k = magazzini.magazzini-k AND
   vendite.tempo-k = tempo.tempo-k AND
   magazzini.città = 'Milano' AND
   tempo.giorno = 1 Gennaio, 1996'
GROUP BY prodotti.categoria
```

111

Esempio query aggregata

- Si supponga adesso che esista una tabella aggregata per categoria

  vendite-aggreg-per-cat(categoria-k, magazzino-k,tempo-k, vendite)

112

Esempio query aggregata

Query sullo schema aggregato

```
SELECT descrizione_categoria, SUM(vendite)
FROM vendite-aggreg-per-cat, categoria, magazzini, tempo
WHERE
   vendite-aggreg-per-cat.categoria-k = categoria.categoria-k
   AND
   vendite-aggreg-per-cat.magazzino-k = magazzini.magazzini-k
   AND
   vendite-aggreg-per-cat.tempo-k = tempo.tempo-k
   AND
   magazzini.città = 'Milano' AND
   tempo.giorno = 1 Gennaio, 1996'
GROUP BY categoria. categoria-k
```

113

Influenza sul codice SQL

- Gli utenti finali e i tool di accesso devono generare codice differente in relazione che esistano o meno le tabelle agrgegate
  - discontinuità delle applicazioni
- Soluzione: aggregate navigator

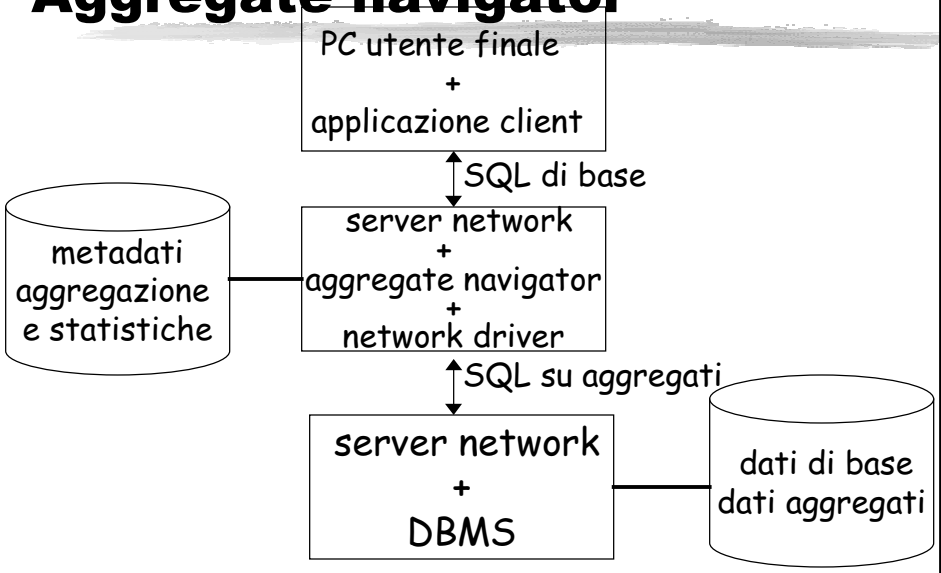
114

Aggregate navigator

- Livello software il cui obiettivo è quello di intercettare le richieste SQL e tradurle utilizzando nel modo migliore le tabelle aggregate
  - si scelgono le più piccole
- le richieste SQL si assumono utilizzare le tabelle di base
- si rende trasparente l'uso degli aggregati all'utente finale

115

Aggregate navigator



116

Strategia di navigazione

1 Ordinare le tabelle dei fatti aggregate dalla più piccola alla più grande
2 si consideri la tabella più piccola
   - si verifichi se tutti gli attributi dimensionali presenti nella query compaiono nelle tabelle delle dimensioni associate alla tabella dei fatti considerata
   - se sì, rimpiazzare le tabelle dei fatti e delle dimensioni di base con le tabelle aggregate
   - se no, ripetere il passo 2 considerando la successiva tabella dei fatti aggregata
3 se nessuna tabella aggregata soddisfa le condizioni precendenti, la query deve essere eseguita utilizzando le tabelle di base

117

118