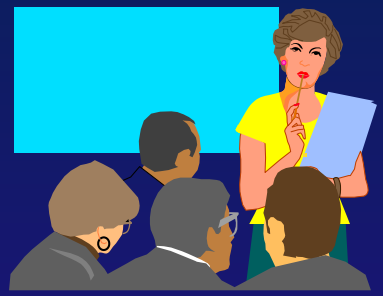
Data warehousing: i concetti, la progettazione, le architetture, i tool

Libri di riferimento

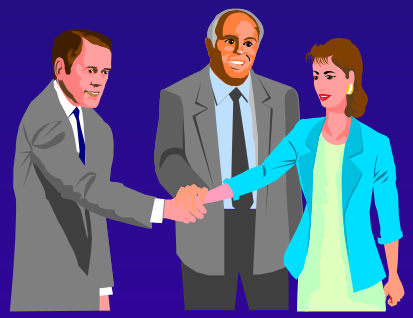
- R. Kimball. The data warehouse toolkit - Practical techniques for building dimensional data warehouses. John Wiley & Sons, Inc. 1996.
- R. Kimball, L. Reeves, M. Ross, W. Thornthwaite. The data warehouse lifecycle toolkit. John Wiley & Sons, Inc., 1998.

Introduzione al data warehousing e concetti di base

Il contesto

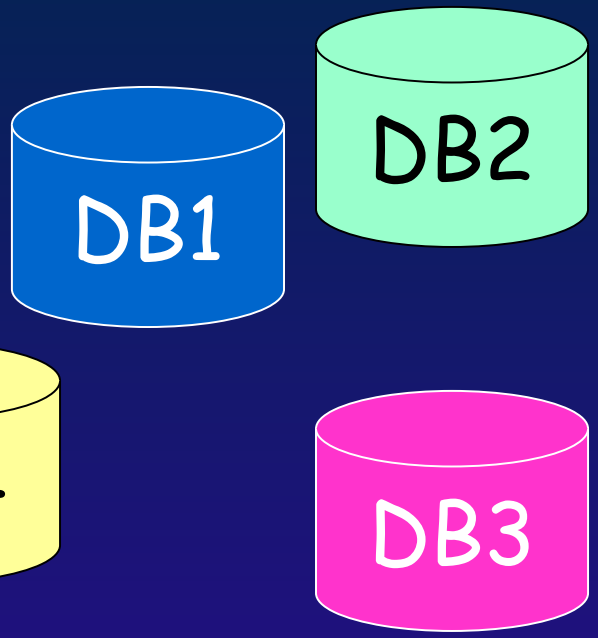


Sistema organizzativo



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

Tipiche frasi di un manager scontento ...

- Abbiamo montagne di dati, ma non riusciamo ad accederli
- Impazzisco ogni volta che mi vengono presentati gli stessi risultati, ma con dati differenti!
- Voglio solo conoscere i dati più importanti!
- Ognuno è cosciente che solo una parte dei dati è significativa ...

Tipiche richieste a cui spesso è difficile dare una risposta

Qual è il volume delle vendite per regione e categorie di prodotto durante l'ultimo anno?

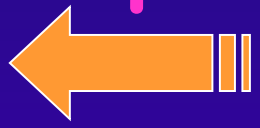
Come si correlano i prezzi delle azioni delle società produttrici di hardware con i profitti trimestrali degli ultimi 10 anni?

Possibili applicazioni

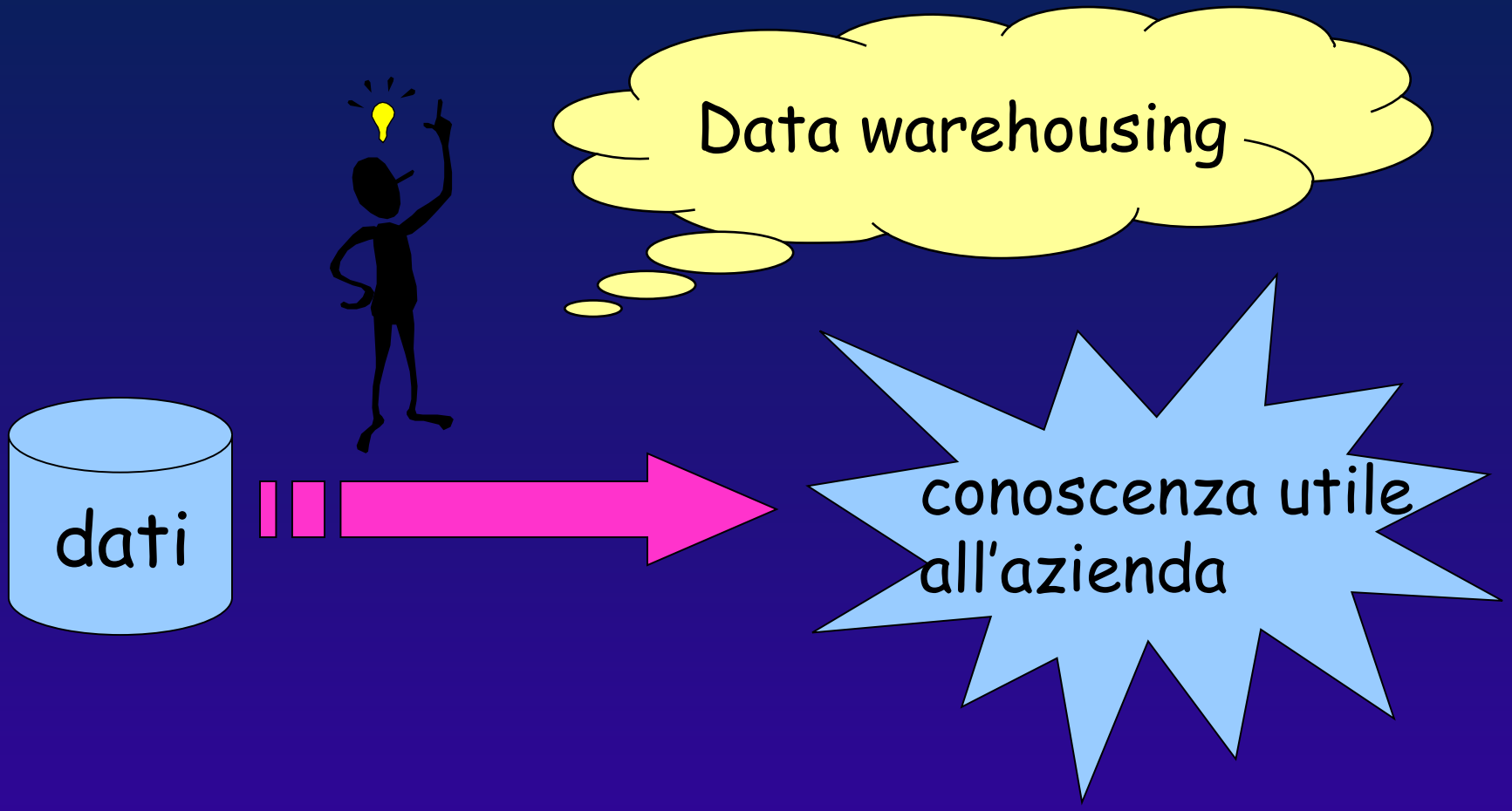
contesti 

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

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

problematiche 

In sintesi ...



Una prima definizione

- 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

Quindi, un data warehouse ...

- Deve garantire un accesso facile e veloce ai dati organizzativi
- i dati devono essere consistenti e di qualità
- deve essere possibile separare e combinare i dati in modi diversi, a seconda delle necessità
- oltre ai dati, devono essere presenti tool per interrogare, analizzare e presentare le informazioni

Perché i DBMS non sono sufficienti?

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

Più formalmente ...

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

⇒ Profondamente diversi

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

Un esempio per capire

- In OLTP, gli utenti
  - prendono gli ordini
  - aprono e chiudono i registratori di cassa
  - effettuano registrazioni
  - inseriscono nuovi dati
  - correggono dati vecchi

⇒gli utenti girano le ruote dell'organizzazione

Un esempio per capire

- In OLAP, gli utenti:
  - contano i nuovi ordini
  - chiedono quando i registratori di cassa sono stati aperti o chiusi
  - chiedono quante prenotazioni sono state efftuate
  - determinano quali sono i nuovi dati
  - chiedono che i dati errati vengano modificati

  ⇒gli utenti guardano le ruote dell'organizzazione

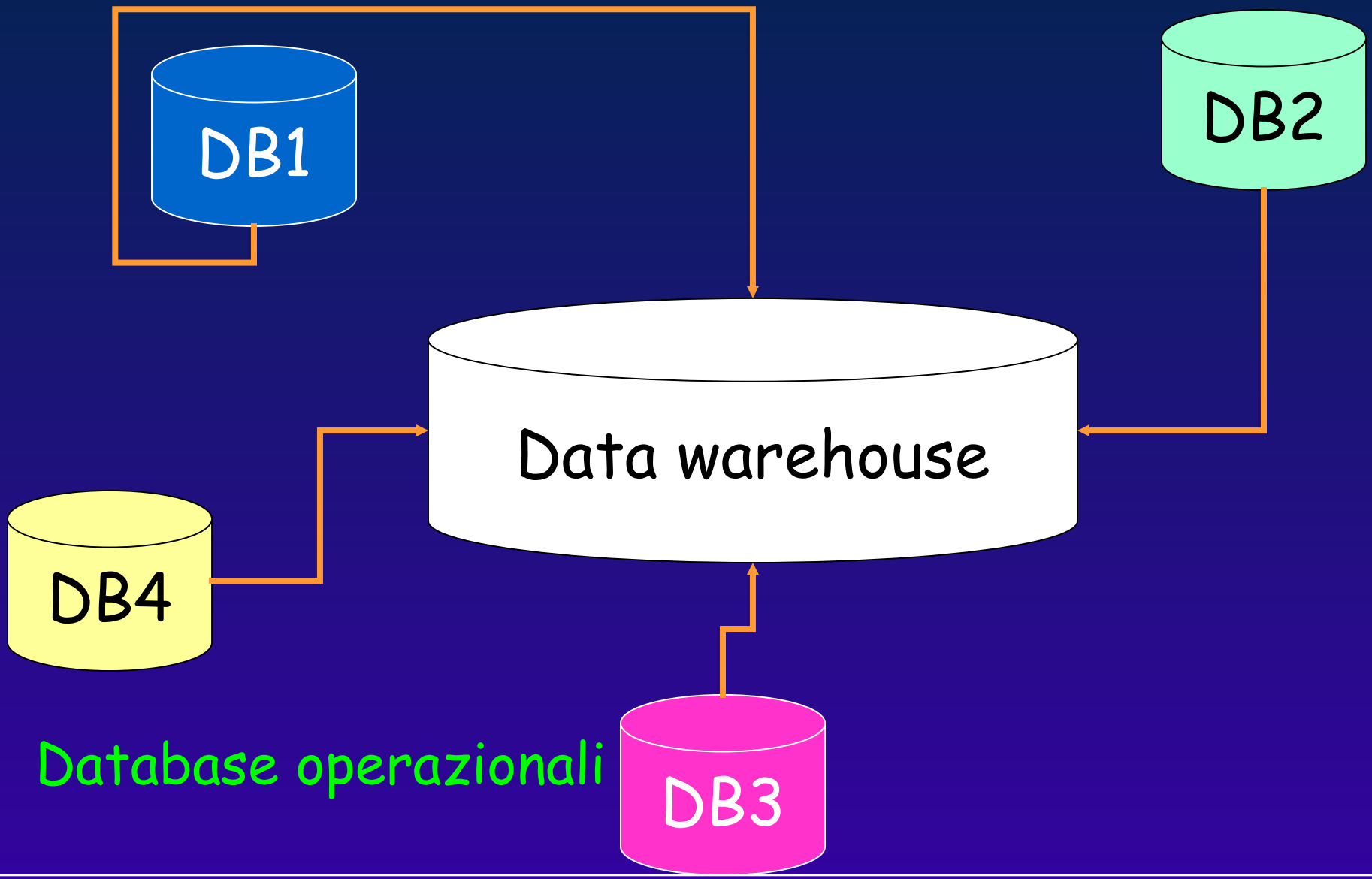
L'aspetto temporale

- OLTP: base di dati come snapshot dell'organizzazione in un certo istante
  - i dati cambiano costantemente
    - inconsistenza temporale
  - incapacità di eseguire interrogazioni storiche, a meno di soluzioni particolari (DBMS temporali)
- OLAP: data warehouse come serie temporale
  - ogni snapshot di un sistema OLTP diventa un livello di un sistema OLAP
  - caricamento ad intervalli regolari

Il modello dei dati

- OLTP:
  - i dati relazionali sono organizzati secondo il modello entità-relazione
  - la rappresentazione dipende dalla struttura dei dati (si cerca di eliminare le ridondanze)
- OLAP:
  - i dati sono organizzati secondo il modello dimensionale
  - la rappresentazione dei dati dipende da come i dati devono essere utilizzati

Il data warehouse



Il data warehouse

Collezione di dati

orientata ai soggetti
integrata
correlata alla variabile tempo
non-volatile
usata per il supporto alle decisioni

Il data warehouse

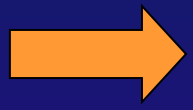
Orientata ai soggetti

Considera i dati di interesse ai soggetti dell'organizzazione e non quelli rilevanti ai processi organizzativi

Il data warehouse

Integrata

I dati arrivano da fonti diverse



diversi formati

integrazione

dati memorizzati in
formati consistenti

Esempio: sesso dell'individuo può essere rappresentato come "M" e "F" o come "0" e "1"

Il data warehouse

Correlata alla variabile tempo

Presenza di dati storici per eseguire confronti, previsioni e per individuare tendenze

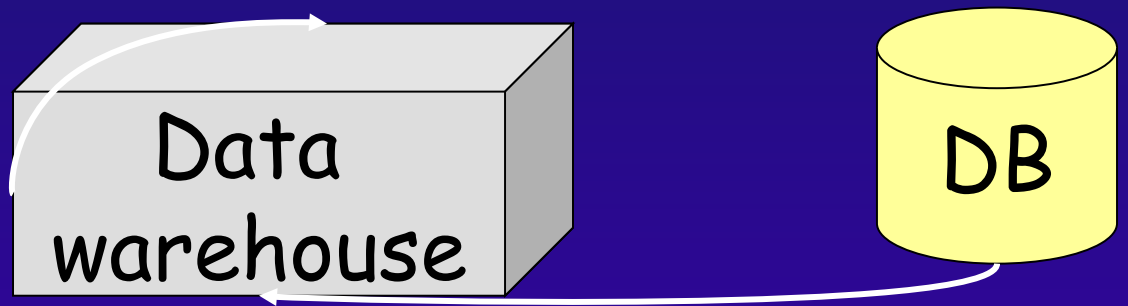
Esempio sistema bancario: dati storici clienti per stabilire quanti clienti chiuderanno il conto nell'anno seguente

Il data warehouse

Non volatile

Una volta che le informazioni sono state inserite nel data warehouse, non possono essere modificate ma solo ricaricate

NO

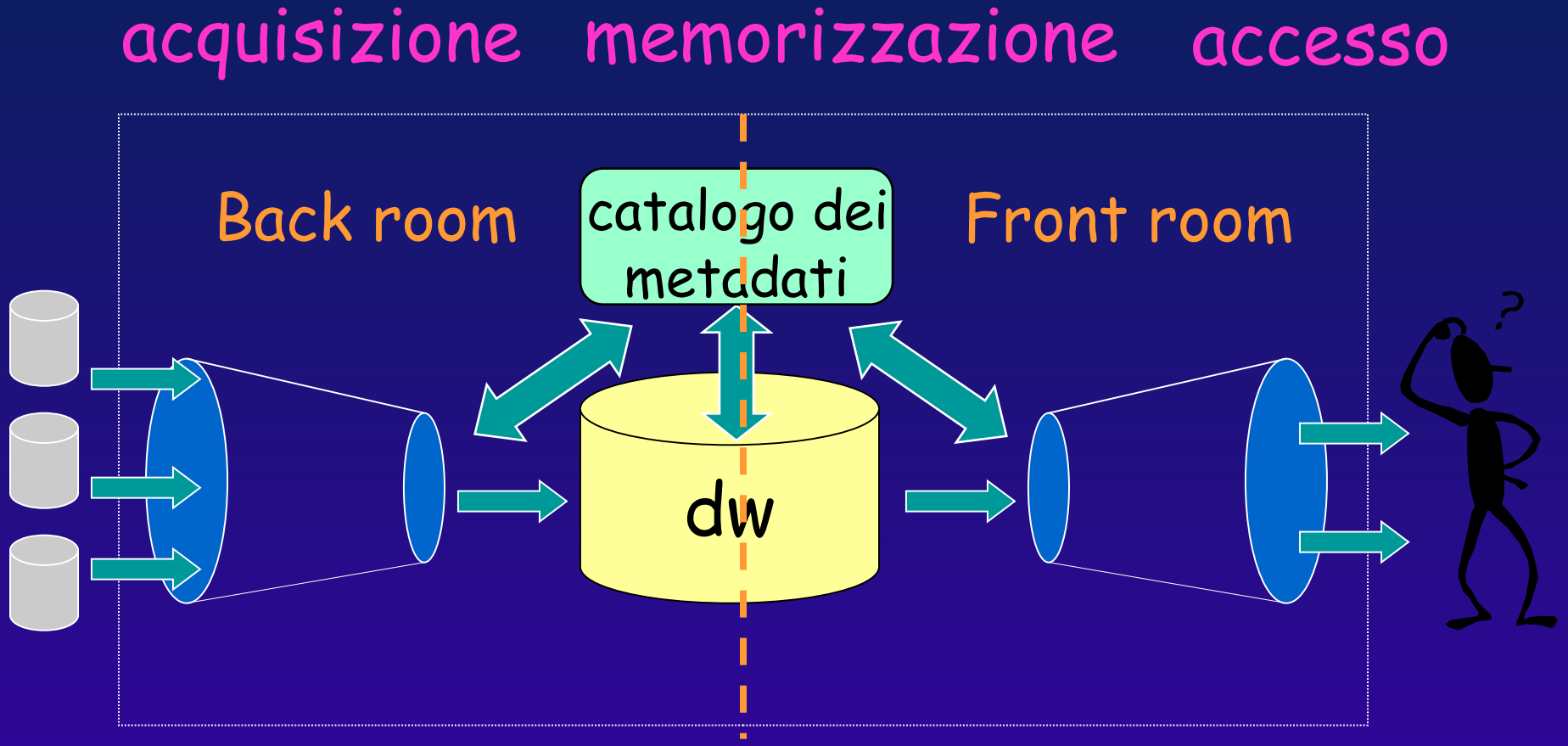


SI

Domanda

- Come si può definire un'architettura per costruire, gestire, interrogare in modo efficiente un data warehouse?

Architettura di base



Acquisizione

- I dati devono essere caricati nel data warehouse, partendo da sorgenti dati esistenti (ad esempio, partendo da un DB operazionale già esistente nella realtà aziendale considerata)

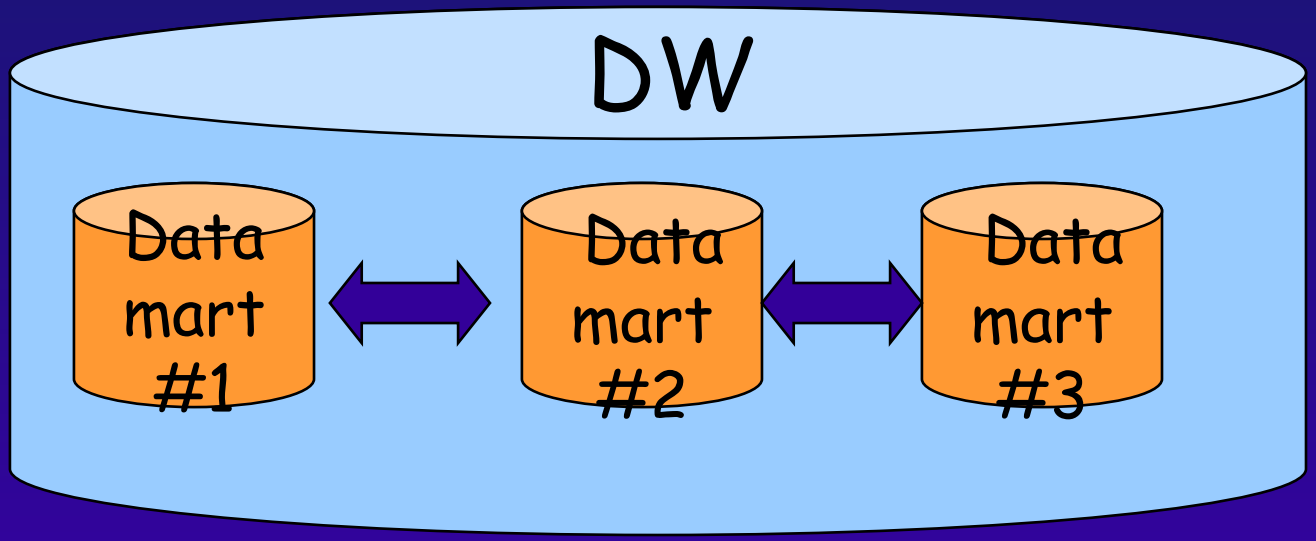
- questo caricamento può richiedere trasformazioni dei dati per ottimizzare il DW rispetto alle esigenze specifiche del supporto decisionale

Memorizzazione

- I dati devono essere memorizzati secondo uno schema che garantisca prestazioni efficienti per l'esecuzione di tipiche operazioni a supporto delle decisioni
- Il DW diventa quindi il nuovo DB informativo, da utilizzare per applicazioni business-oriented

Memorizzazione

- Un DW rappresenta spesso l'unione di più data mart, ciascuno corrispondente ad un ben definito processo aziendale



Accesso

- Il DW viene utilizzato per rispondere a tipiche problematiche business-oriented
- a questo proposito, si possono utilizzare tool più o meno sofisticati a supporto del sistema decisionale aziendale
- i tool devono essere semplici e non devono richiedere conoscenza dei sistemi utilizzati per la memorizzazione e la gestione del DW

Metadati

- Tutte le fasi descritte in precedenza generano metadati (cioè dati che descrivono altri dati)
- un tipico esempio di metadato consiste nelle regole che descrivono come il DW è stato generato a partire dai dati sorgente
- i metadati sono una componente fondamentale per la progettazione e l'utilizzo di ogni sistema di data warehousing

Nel seguito ...

- Cercheremo di capire quali dati contiene un Dw e come sono logicamente e fisicamente organizzati
  - progettazione
- analizzeremo la fase di acquisizione dati
  - back room
- analizzeremo la fase di accesso ai dati
  - front room
- chiariremo il ruolo dei metadati nel contesto delle due fasi introdotte

Progettazione di un data warehouse

Parte 1

Le fasi della progettazione

- Progettazione concettuale
- progettazione logica
- progettazione fisica

Progettazione concettuale

- Fornisce una rappresentazione formale del contenuto informativo del DW
- si basa sull'uso di un modello concettuale
- l'output è uno schema concettuale
- indipendente dal sistema che verrà utilizzato per la sua implementazione

Progettazione logica

- Lo schema concettuale viene tradotto nel modello dei dati del sistema prescelto

- l'output è uno schema logico dei dati

Progettazione fisica

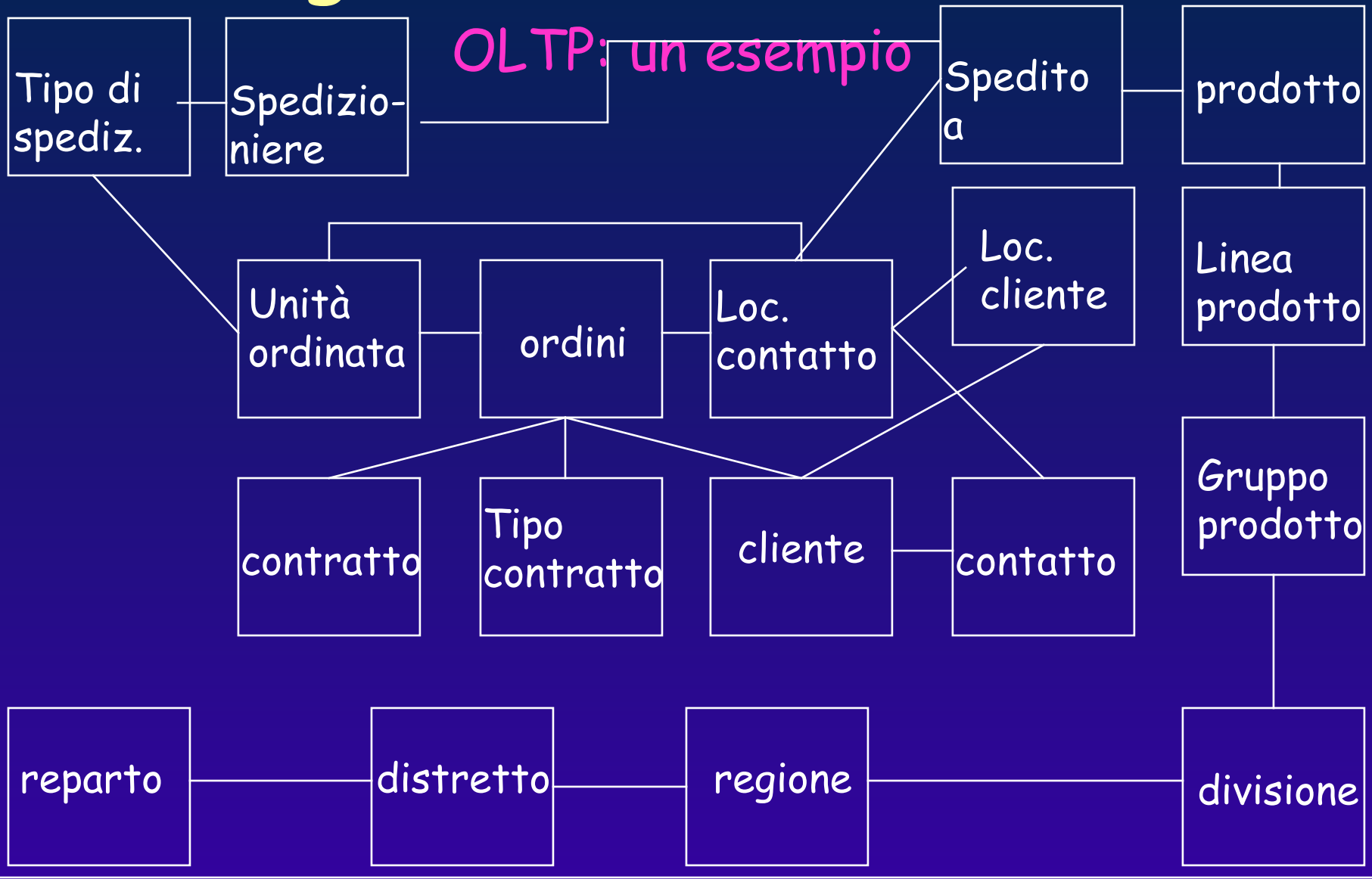
- Fase in vui vengono scelte le caratteristiche fisiche del sistema
- l'output è lo schema fisico, che descrive le strutture di accesso e memorizzazione del DW

Progettazione concettuale

OLTP

- I dati relazionali sono organizzati secondo il modello entità-relazione
- si cerca di eliminare il più possibile la ridondanza
  - maggiore efficienza delle operazioni di aggiornamento
  - le transazioni vengono semplificate
- non esiste una relazione più importante di un'altra schema → simmetrico
- ci possono essere molti modi per connettere (mediante un'operazione di join) due tabelle
- per query che coinvolgono molte tabelle, gli schemi sono troppo complessi per essere navigati e per essere compresi dagli utenti

Progettazione concettuale



Progettazione concettuale

OLAP

- Il DW deve essere progettato per rendere efficiente l'esecuzione di query complesse piuttosto che per velocizzare gli inserimenti e gli aggiornamenti
- l'idea è quella di denormalizzare lo schema del DB operazionale in modo da rendere le query più efficienti
- la denormalizzazione non costituisce un problema, in quanto gli update ad un DW sono periodici e vengono effettuati in modalità batch

Progettazione concettuale

OLAP

- i dati sono organizzati secondo un modello multidimensionale, intuitivo e in grado di garantire alte prestazioni
- l'eliminazione della ridondanza non è un obiettivo
  - non si devono eseguire operazioni di aggiornamento
- esistono alcune relazioni più importanti delle altre (tabelle dei fatti)

  schema → asimmetrico
- un solo modo per connettere (mediante un'operazione di join) due tabelle

Progettazione concettuale

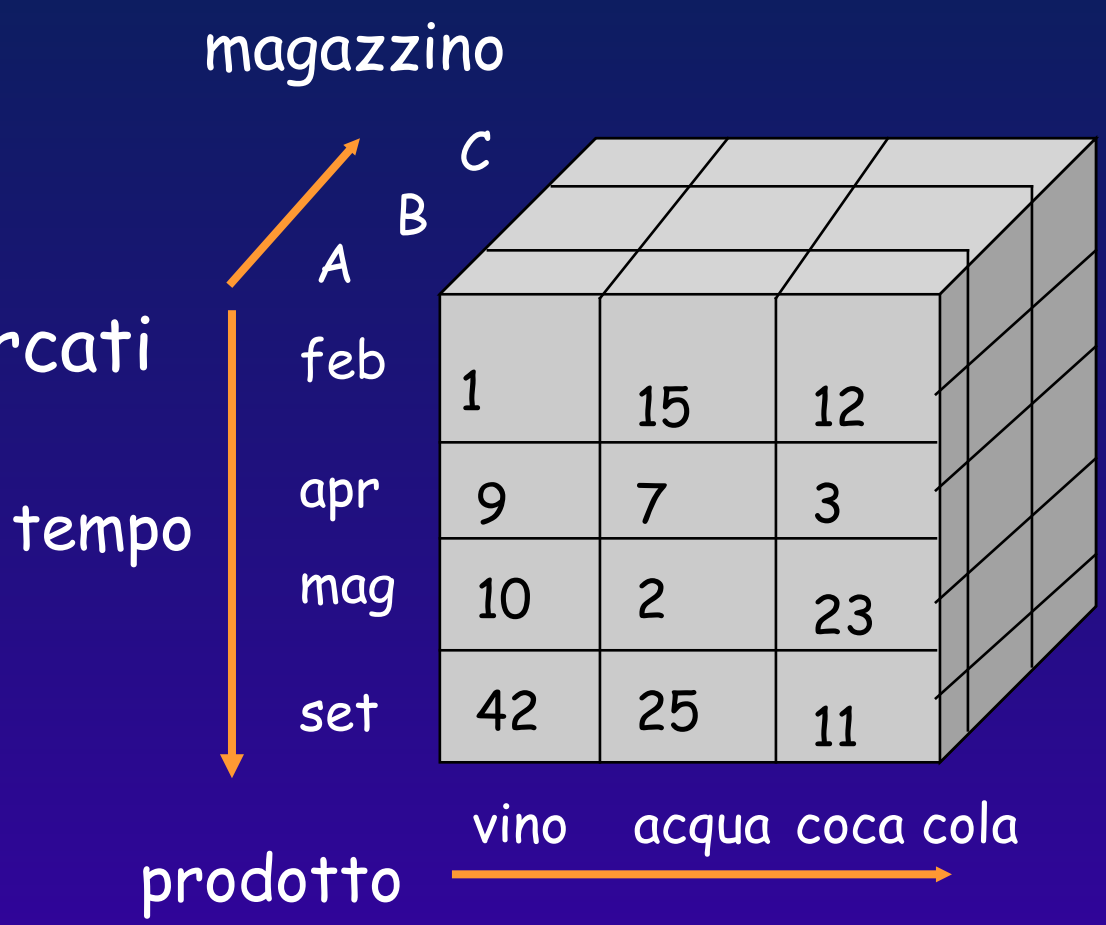
OLAP

- L'idea alla base del modello multidimensionale e` che ogni processo aziendale puo` essere rappresentato come un cubo (in genere, hypercubo) in cui:
  - le celle sono specifiche misure del processo
  - i lati sono le dimensioni naturali dei dati
- in genere:
  - 4-15 dimensioni
  - numero arbitrario di fatti

Progettazione concettuale

OLAP

Processo:
vendite in una
catena di supermercati



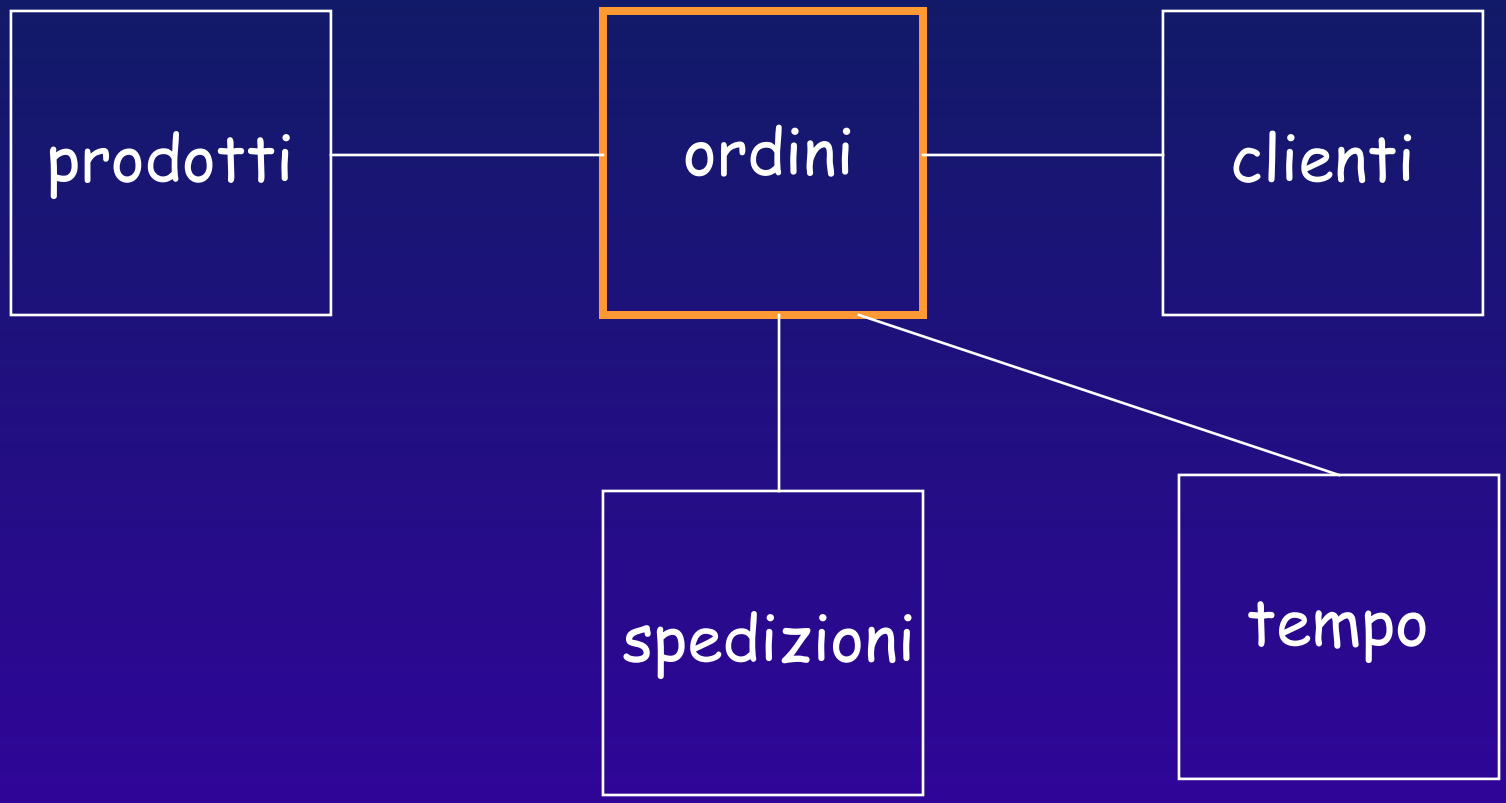
Progettazione concettuale

OLAP

- Esistono vari modelli multidimensionali
  - modello a stella
  - modello a costellazione di fatti
  - modello a fiocco di neve (snowflake)
- I vari modelli si differenziano principalmente rispetto al numero di tabelle dei fatti
- Consentono di ottenere un numero minore di tabelle rispetto al modello ER
- uno schema multidimensionale può essere implementato in
  - sistemi relazionali
  - sistemi multidimensionali
  - sistemi ad oggetti

Progettazione concettuale

OLAP: un esempio



Progettazione concettuale

- Nel contesto del corso, cominceremo ad introdurre il modello noto come modello a stella
- introdurremo poi gli altri modelli, motivandone l'esigenza

Il modello a stella

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

Il modello a stella

- I fatti identificano l'attività principale e sono caratterizzati dai dati di dettaglio che si desidera analizzare
  - sono rappresentati mediante tabelle dei fatti
- le dimensioni sono i parametri che influenzano i dati di dettaglio e rispetto ai quali si analizzano tali dati
  - sono memorizzate in tabelle delle dimensioni
- il collegamento tra tabelle dei fatti e tabelle delle dimensioni avviene mediante 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

Esempio

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

Le dimensioni

- Per determinare le dimensioni occorre porsi la seguente domanda:
  - rispetto a quali parametri voglio analizzare i fatti?
- Devono essere scelte solo le entità rilevanti per le analisi che si intendono effettuare
- è importante che tutte le informazioni su un certo parametro siano raggruppate in un' unica entità (tabella, oggetto, ecc.)
- esiste sempre una dimensione temporale

Le dimensioni

- Le dimensioni sono tipicamente caratterizzate da attributi:
  - testuali
  - discreti
- vengono utilizzate per selezionare i fatti
- possono anche essere numeriche
  - dimensione di un prodotto

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
- se è una descrizione discreta di qualcosa che è ragionevolmente costante
  - attributo di una dimensione

Le dimensioni

- Si consideri ad esempio il costo di un prodotto
- se non ci interessa misurare le variazioni di tale costo nel tempo, questa informazione deve essere classificata come attributo della dimensione prodotto
- altrimenti, deve essere classificata come fatto e inserita nella tabella dei fatti (vedi oltre)

Le dimensioni

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

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

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

I fatti

- Fatto addittivo: può essere sommato rispetto a qualunque dimensione
- fatto semiaddittivo: può essere sommato solo rispetto a determinate dimensioni
- fatto non-addittivo: non può essere sommato
  - i record possono solo essere contati

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, ...

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

Esempio

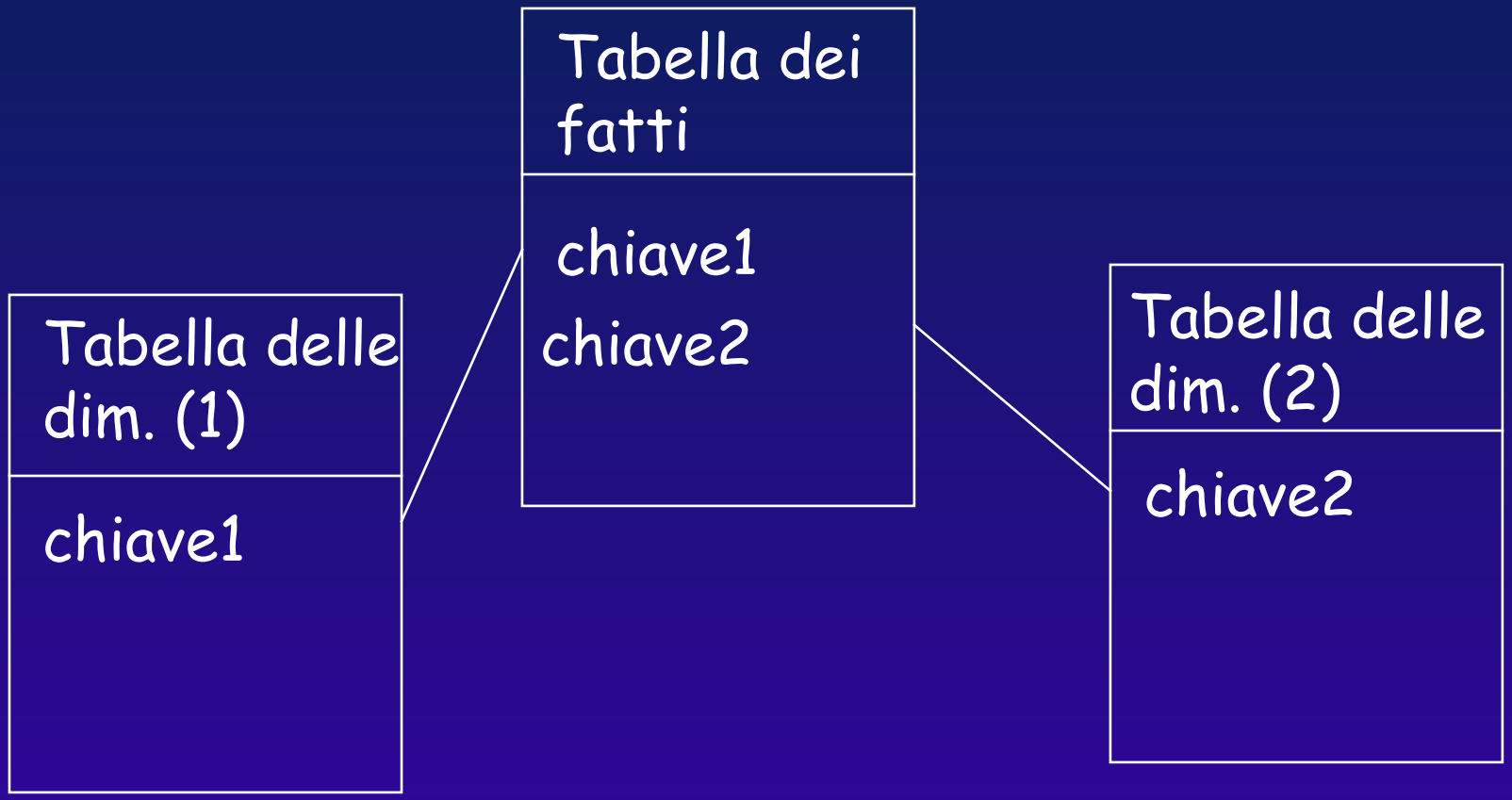


Tabella dei fatti
chiave1
chiave2
Tabella delle dim. (1)
chiave1
Tabella delle dim. (2)
chiave2

Interrogazioni

- Le interrogazioni impongono condizioni su una o più tabelle delle dimensioni che si riflettono in restrizioni sulla tabella dei fatti

- Esempio: selezionare tutti i prodotti di colore rosso venduti nell'ultimo trimestre nell'Italia settentrionale

Interrogazioni

- Tali interrogazioni sono dette interrogazioni di star-join in quanto implicano l'esecuzione di join tra una tabella dei fatti e una o più tabelle delle dimensioni

- l'esecuzione efficiente di tali interrogazioni richiede indici specializzati (join index e star join index)

Schema interrogazione tipo

```
SELECT  lista attributi e aggregati
FROM    tabelle dei fatti e delle dimensioni
WHERE   join tabella fatti e tab. dimensione 1
   AND
          join tabella fatti e tab. dimensione 2
   AND
     ...
   AND
          condizioni su attributi dimensione
GROUP BY      attributo dimensione
ORDER BY      attributo dimensione
```

Esempio


tempo (dim)
Tempo-k
attributi tempo
Vendite (fatti)
Tempo-k
prodotto-k
magazzino-k
promozione-k
incassi
n. clienti
costi
prodotti (dim)
Prodotto-k
…
descrizione
marca
promozioni
Promozione-k
attributi prom.
magazzini (dim)
Magazzino-k
attributi mag.

Esempio

Determinare quante unità sono state vendute e quali incassi sono stati ricavati per ogni prodotto di marca "Findus"

```
SELECT  p.descrizione, sum(f.incassi), sum(f.n-unità)
FROM    vendite f, prodotti p
WHERE   f.Prodotto-k = p.Prodotto-k
                                AND
          p.marca = "Findus"
GROUP BY      p. descrizione
ORDER BY      p.descrizione
```

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

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

Interviste

- Devono fornire suggerimenti su cosa gli utenti si aspettano ed eventualmente correggere alcune aspettative "errate"
- le domande tipiche agli utenti sono volte a determinare granularita` , dimensioni e fatti
- le domande rivolte al DBA sono volte a comprendere il sistema ed i dati esistenti

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

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

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

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)

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

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?

Passo 2: scelta della granularita

- Quindi, un DW spesso richiede che i dati siano espressi alla granularità più fine possibile, non tanto perché le interrogazioni vogliono vedere i singoli record individuali, ma perchè le interrogazioni hanno bisogno di selezionare parti del database in un modo molto preciso

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

Passo 3: scelta delle dimensioni

- Esempio:

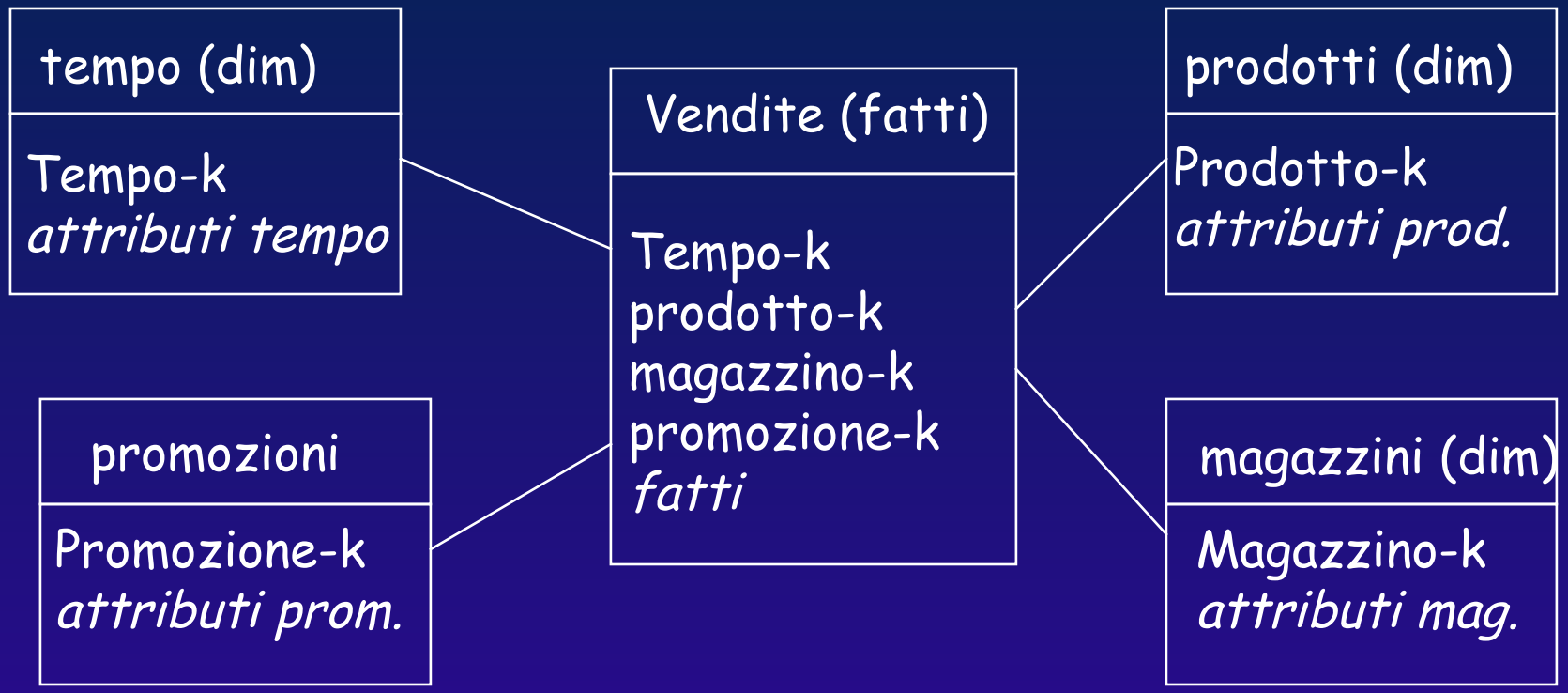
  nel nostro esempio, le dimensioni primarie sono:
  - tempo
  - prodotti
  - magazzini

  dimensioni aggiuntive
  - promozioni

Passo 3: scelta delle dimensioni

- Per ciascuna dimensione, devono essere specificati gli attributi che la caratterizzano
- spesso si tratta di attributi alfanumerici
- gli attributi all'interno di una singola dimensione dovrebbero essere correlati (ci deve essere attinenza tra di loro)
- se una dimensione contiene attributi non correlati, è meglio suddividere la dimensione in due dimensioni distinte

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

Osservazioni sulla normalizzazione dello schema

- 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

Passo 4: scelta dei fatti

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

La dimensione tempo

- È presente in ogni DW in quanto virtualmente ogni DW rappresenta una serie temporale

- Domanda: perché non si può usare un campo di tipo data nella tabella dei fatti ed evitare di usare una dimensione per il tempo?

- 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.)

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
  - ...

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

Gerarchie

- Tutti gli attributi della gerarchia devono essere inseriti all'interno della dimensione
- questo comporta ridondanza ma questo è accettabile in quanto la dimensione delle tabelle delle dimensioni in genere è inifluente sulla dimensione totale del database
- Esempio: se ci sono 30000 prodotti e 30 dipartimenti distinti, ci saranno in media 1000 ripetizioni

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

Gerarchie

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

Drill-up e drill-down

- Drill-down: aggiungere informazioni estratte da una tabella delle dimensioni ad un report
- Drill-up: sottrarre informazioni contenute da una tabella delle dimensioni da un report
- Le gerarchie possono essere utilizzate per effettuare operazioni di drill-up e drill-down ma non sono necessarie
- nel caso di gerarchie multiple, tali operazioni devono potere essere effettuate rispetto a ciascuna gerarchia

Esempio di drill-down e -up

down

up

Dipartimento	Incassi	Unità vendute
Panificio	Lit. 12100000	5088
Cibo surgelato	Lit. 23000000	15000
…		

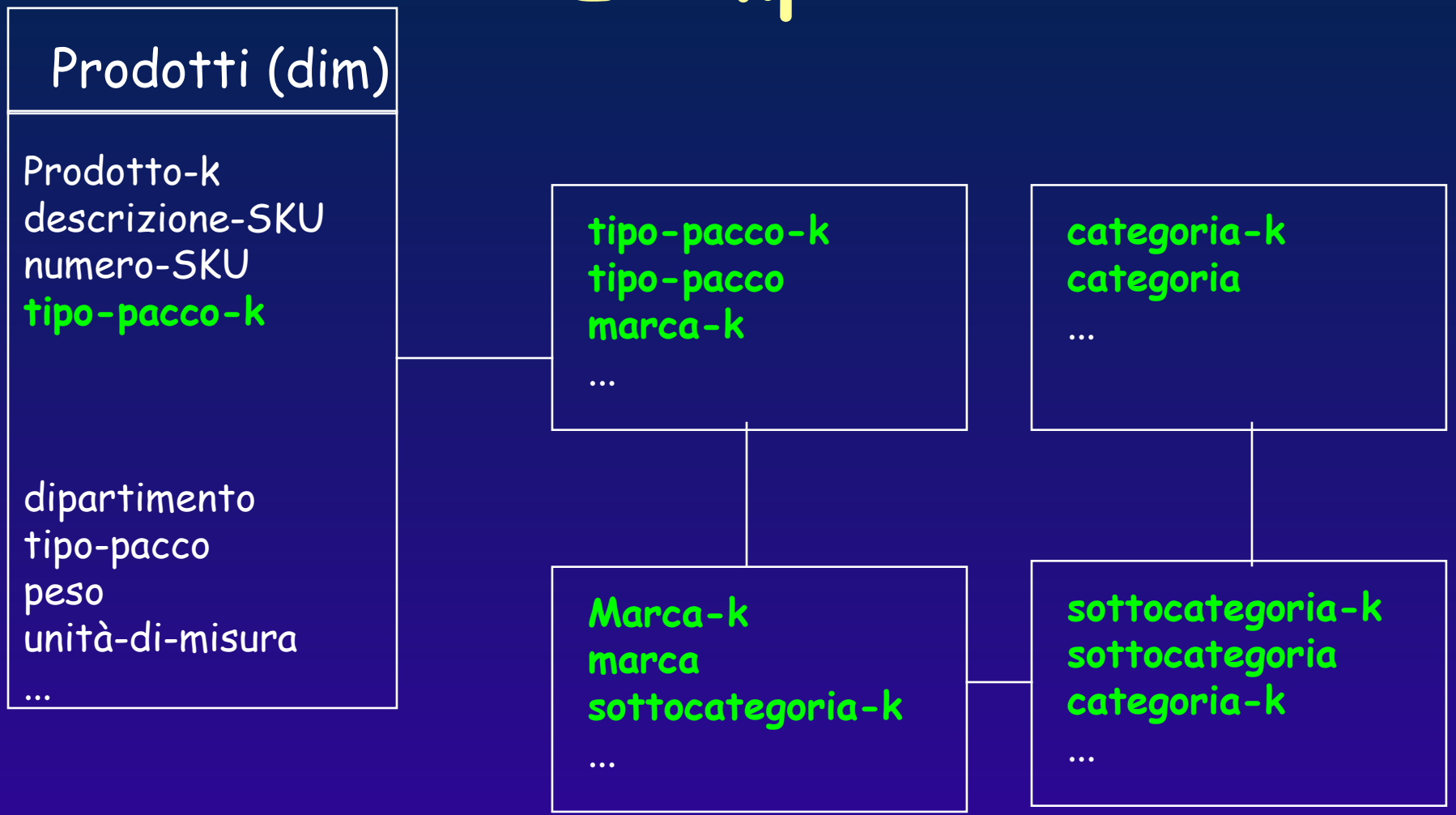
Dipartimento	Marca	Incassi	Unità vendute
Panificio	Barilla	6000000	2600
Panificio	Agnesi	6100000	2488
Cibo surgelato	Findus	15000000	6500
Cibo surgelato	Orogel	8000000	8500
…			

Schemi snowflake

- Di fronte ad una gerarchia, si potrebbe pensare che sia piu` conveniente suddividere gli attributi della tabella in piu` tabelle

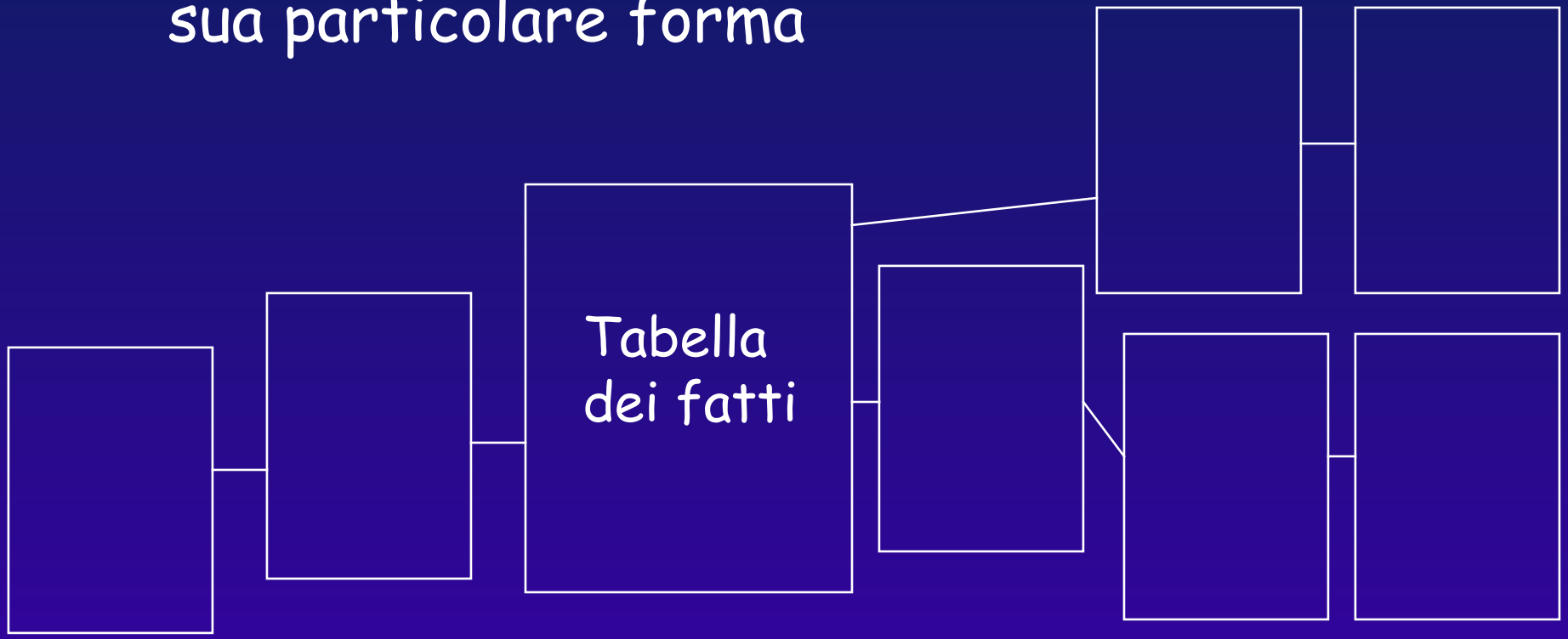
- piu` in particolare, poiche` una gerarchia rappresenta molte relazioni molti-a-uno, si potrebbe pensare di utilizzare una tabella per ogni relazione

Esempio


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

Schemi snowflake

- Lo schema risultante si definisce schema a fiocco di neve (snowflake schema) per la sua particolare forma



Schemi snowflake

- Uno schema snowflake rende meno efficienti le operazioni di ricerca, anche se la tabella e` grande
- e` conveniente utilizzare uno schema snowflake solo se questo approccio aumenta la leggibilita` dello schema e le prestazioni globali

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

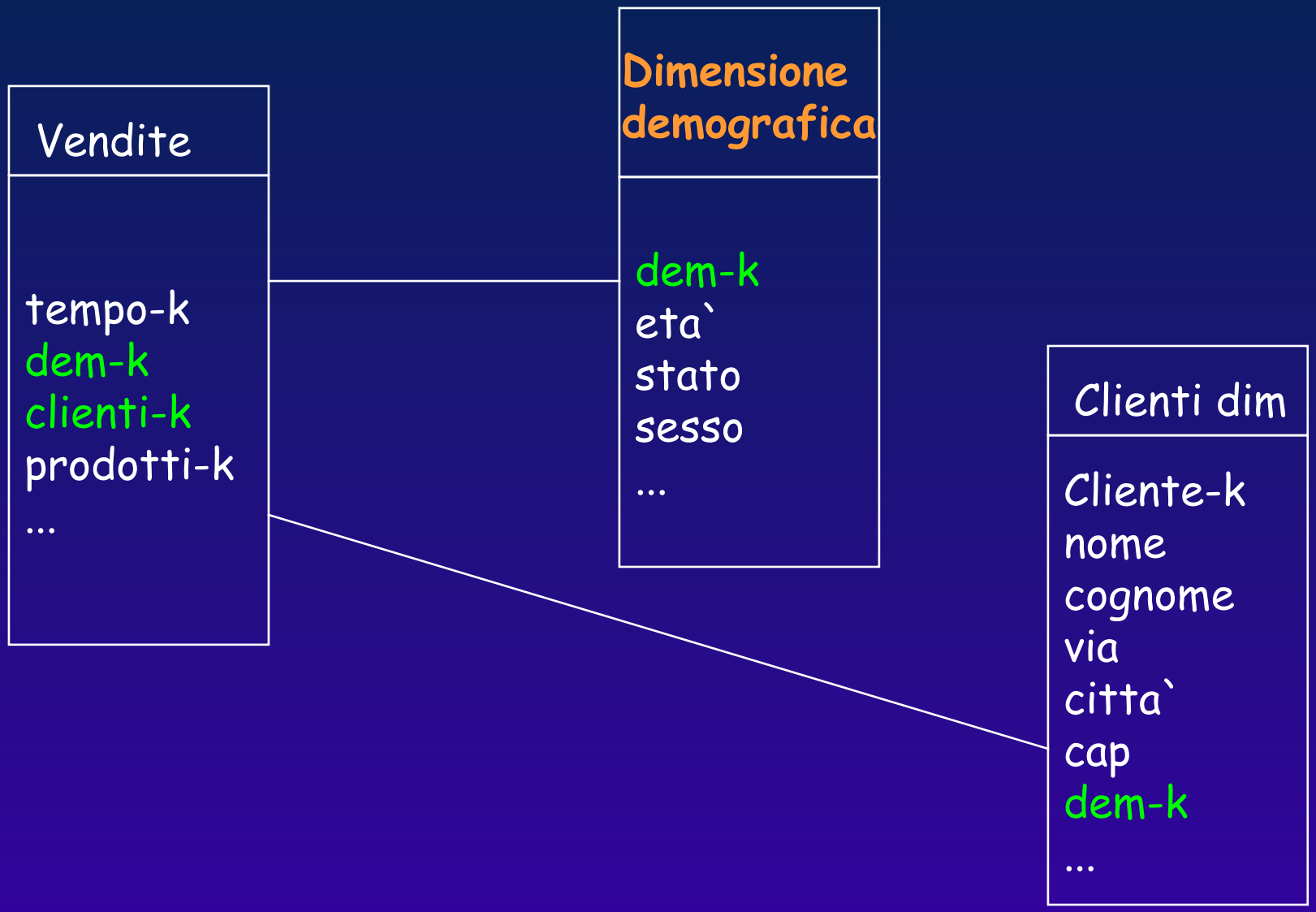
Minidimensioni

- Solo in alcuni casi particolari e` conveniente utilizzare gli schemi snowflake
- uno di questi casi e` relativo alle informazioni demografiche per dimensioni quali clienti, persone, ecc.:
  - eta`
  - stato
  - sesso
  - ...

Minidimensioni

- Queste informazioni sono piuttosto statiche e il numero delle possibili combinazioni e` piuttosto basso
- inoltre questi campi sono quelli tipicamente utilizzati in fase di interrogazione (per i quali e` quindi conveniente costruire un indice)
- in questi casi, puo` convenire generare una o piu` nuove tabelle dei fatti

Esempio


Vendite
tempo-k
dem-k
clienti-k
prodotti-k
...
Dimensione demografica
dem-k
eta`
stato
sesso
...
Clienti dim
Cliente-k
nome
cognome
via
citta`
cap
dem-k
...

Dimensioni causali

- Una dimensione si definisce causale se descrive i fattori che hanno provocato cambiamenti ai fatti (cioè al processo aziendale considerato)
- Esempio: la tabella delle promozioni è una dimensione causale in quanto le promozioni possono generare una variazione alle vendite di un certo prodotto in un certo dipartimento, in un certo periodo

Dimensioni causali

- Spesso, le cause possono essere diverse
- Esempio: nel caso delle promozioni:
  - riduzione prezzo
  - presentazione coupons alla cassa
  - pubblicità su cartelloni
- due possibili criteri di progettazione:
  - una sola dimensione per tutte le cause
  - una dimensione per ogni causa

Dimensioni causali

- Una sola dimensione per tutte le cause
  - spesso le cause sono fortemente correlate, quindi la dimensione è solo di poco più grande della somma di eventuali dimensioni indipendenti
  - la dimensione può essere analizzata in modo molto efficiente
  - richiede la costruzione di una chiave artificiale

Dimensioni causali

- Una dimensione per ogni causa:
  - il significa di ciascuna dimensione è più comprensibile all'utente
  - le singole dimensioni sono più facili da amministrare
  - non vengono generate chiavi artificiali

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
  - somma incassi/unità su promozioni

Semiaddittività dei fatti

- Numero clienti non è un fatto addittivo
- Infatti:
  - somma n. clienti su tempo OK
  - somma n. clienti su dipartimenti OK
  - somma n. clienti su promozioni OK
  - MA:
    - somma n. clienti su prodotto genera problemi

Semiaddittività dei fatti

- Si supponga di volere determinare quanti clienti hanno comprato carne o pesce
- 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

Semiaddittività dei fatti

- Il numero clienti è un fatto semiaddittivo, poiché può essere sommato solo rispetto ad alcune dimensioni
- l'unica soluzione al problema dei fatti semiaddittiviti è quella di cambiare la granularità del database, portandola a livello singola transazione

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

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
  - è conveniente avere prezzi unitari perché si riduce lo spazio utilizzato

Tabelle dei fatti sparse

- Una tabella dei fatti è detta sparsa se solo per alcune combinazioni dei record contenuti nelle tabelle delle dimensioni esiste un record nella tabella dei fatti

- altrimenti è detta densa

Tabelle dei fatti sparse

- Esempio
  - tabella dei fatti per le vendite (solo alcuni prodotti - circa 10%-sono venduti in un certo giorno)
    - sparsa
  - tabella dei fatti per gli inventari: in questo caso, si assume di avere almeno una dimensione tempo e una dimensione prodotto
    - per ogni combinazione prodotto-tempo, esiste un record nella tabella dei fatti
    - densa

Dimensioni degeneri

- Una dimensione si definisce degenere se la sua chiave viene utilizzata nella tabella dei fatti senza che esista una tabella delle dimensioni corrispondente
- Esempi tipici: n. associati a documenti, senza che gli stessi documenti siano rappresentati come dimensioni
  - n. ordine, n. spedizione, …
- gli attributi che identificano i documenti corrispondenti possono essere inseriti nella tabella dei fatti

Cambiamenti nelle dimensioni

- Finora si è assunto che le dimensioni fossero indipendenti
- in particolare, si è assunto che fossero indipendenti dal tempo
- questo non è sempre vero in quanto nel tempo una certa dimensione (ad esempio i prodotti o i clienti) può variare (nuovo sistema di impacchettamento, nuovo indirizzo)
- questi cambiamenti in genere sono rari, nel senso che le dimensioni cambiano lentamente

Cambiamenti nelle dimensioni

- Questo implica che sono debolmente dipendenti dalla dimensione tempo
- ci sono tre soluzioni per rappresentare questa situazione all'interno dello schema:
  - a) sovrascrivere i vecchi valori
  - b) creare una nuova dimensione per ogni cambiamento
  - c) tenere due valori: precedente e corrente

Esempio

- Si consideri una dimensione cliente
- si supponga che cambi l'indirizzo di un certo cliente

Cliente-k

nome

cognome

indirizzo

telefono

fax

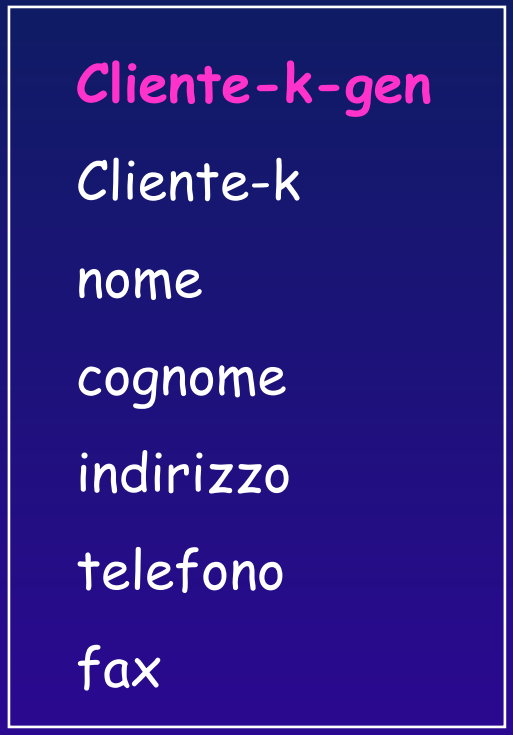
Soluzione a)

- Il vecchio indirizzo viene sostituito con il nuovo
- facile da implementare
- si perde la storia dei cambiamenti
- le chiavi non cambiano
- utile in caso di errori

Soluzione b)

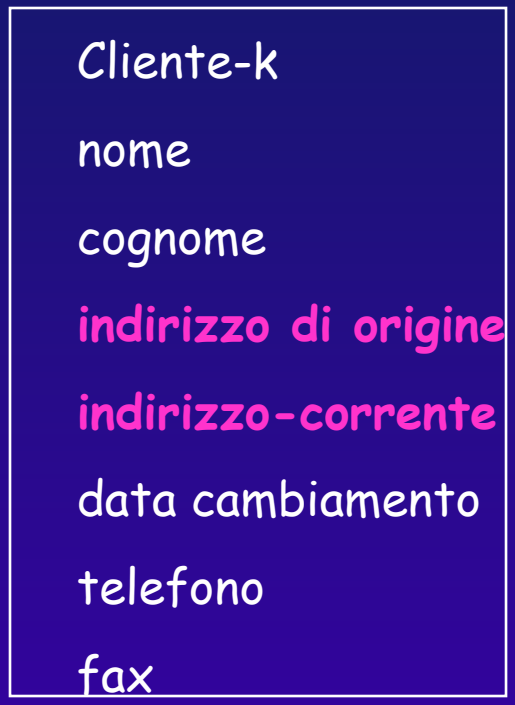
- Si aggiunge una tupla nella tabella delle dimensioni e si crea una chiave generalizzata
- ogni tupla sarà valida relativamente ad un certo periodo
- la chiave precedente non è più chiave (più record avranno lo stesso valore per il vecchio campo chiave)
- si noti che la nuova chiave dovrà essere inserita a livello di DW in quanto difficilmente sarà presente nel DB operazionale

Soluzione b)


Cliente-k-gen
Cliente-k
nome
cognome
indirizzo
telefono
fax

Soluzione c)

- Si aggiunge un campo indirizzo-corrente, che contiene il nuovo valore
- si aggiunge un campo data, per tenere conto della data del cambiamento
- si ridenomina il campo indirizzo come indirizzo-di-origine



Soluzione c)

- Questa soluzione non è precisa come la soluzione b)
- non permette di tenere conto di tutta la storia ma solo dell'ultima variazione
- non aggiunge record alla tabella delle dimensioni

Dimensioni sporche

- Contengono molti duplicati e dati estranei
- non hanno quindi un significato concettuale ben definito
- Esempio:
  - tabella individui in una DB per un'applicazione bancaria
  - le banche in genere operano sui conti e non sugli individui
  - per gli individui, e` facile avere informazioni errate e duplicate

Dimensioni eterogenee

- In un DW dove una dimensione deve descrivere un insieme di entita` molto diverse tra di loro, la tecnica raccomandata e` quella di creare una tabella dei fatti e una tabella delle dimensioni di base

- queste tabelle verranno utilizzate per interrogazioni che coinvolgono diversi tipi di entita`

Dimensioni eterogenee

- Quindi si puo` creare una tabella dei fatti e una tabella delle dimensioni per ogni tipo specifico di entita`
- queste tabelle vengono utilizzate per interrogazioni che coinvolgono un solo tipo di entita`
- gli attributi comuni a tutti i tipi di entita` devono essere presenti in tutte le tabelle delle dimensioni

Esempio

- Attivita` principale: gestione bancaria
- dimensioni:
  - tempo
  - filiali
  - conti
  - prodotti
  - …
- I dati degli intestatari sono inseriti nella dimensione prodotti

Esempio

- Esistono molte varieta` di prodotti bancari:
  - c\c
  - depositi bancari
  - carte di credito
  - fidi
  - mutui
  - ...
- ciascuna categoria di prodotti e` caratterizzata da un insieme specifico di attributi

Esempio

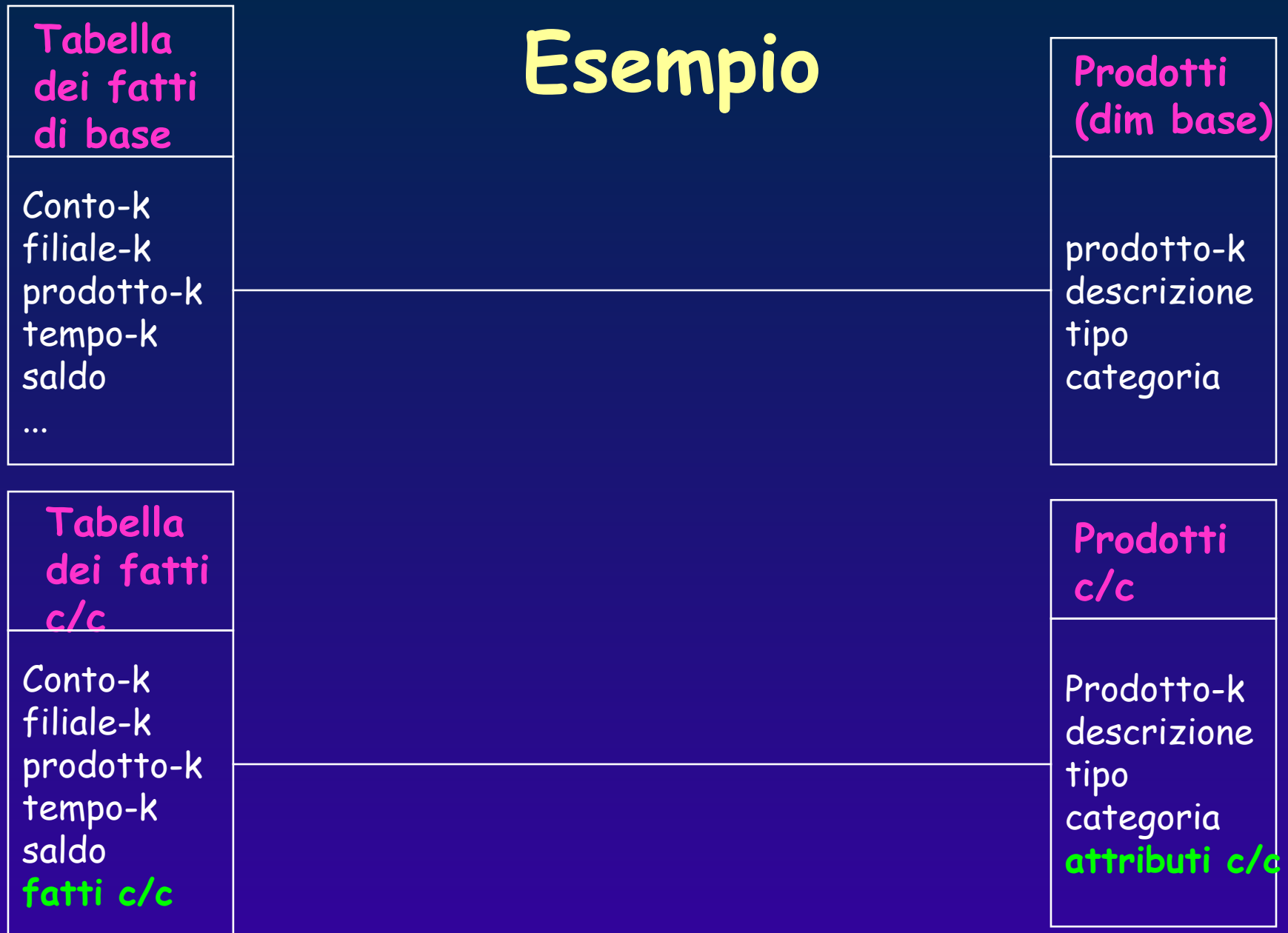


Tabella dei fatti di base
Conto-k
filiale-k
prodotto-k
tempo-k
saldo
...
Prodotti (dim base)
prodotto-k
descrizione
tipo
categoria
Tabella dei fatti c/c
Conto-k
filiale-k
prodotto-k
tempo-k
saldo
fatti c/c
Prodotti c/c
Prodotto-k
descrizione
tipo
categoria
attributi c/c

Dimensioni eterogenee

- In alcuni casi, la tabella dei fatti puo` non essere duplicata
- questa situazione si verifica ad esempio quando la granularita` prescelta e` a livello transazionale
- in questo caso, si considera un unico fatto e quindi non ha senso duplicare le tabelle dei fatti
- al contrario, le tabelle delle dimensioni possono essere duplicate

Granularita` transazionale e snapshot periodico

- Nel caso di granularita` transazionale, potrebbe capitare di avere anche bisogno di informazioni sintetiche periodiche, ad esempio mensili
- in questa situazione puo` essere conveniente mantenere accanto alla tabella dei fatti transazionale una tabella dei fatti periodica, nella quale siano contenuti come precalcolati i fatti periodici di interesse
- si noti che questi fatti possono anche essere calcolati a partire dal DW transazionale, ma ovviamente questa soluzione e` molto meno efficiente

Granularita` transazionale

- In genere e` sempre conveniente rappresentare i fatti al livello piu` basso di granularita`
  - spesso a livello transazionale
- tabelle dei fatti a livello transazionale sono spesso la destinazione naturale dei dati operazionali
- base per l'applicazione di tecniche di Data Mining

Tabelle dei fatti anomale

- Come abbiamo visto, le tabelle dei fatti contengono, per ogni combinazione dei chiavi relative alle varie dimensioni, alcune informazioni numeriche e molto spesso addittive
- in alcuni contesti applicativi, puo` capitare di costruire tabelle dei fatti senza fatti!
- Tali tabelle vengono definite tabelle dei fatti anomale

Tabelle dei fatti anomale

- In questo caso, la tabella dei fatti rappresenta semplicemente una relazione molti-a-molti, senza aggiungere alcuna nuova informazione
- Esempi:
  - Attivita` principale: corsi universitari
    - dimensioni: corsi, professori, studenti, tempo
  - attivita` principale: assegnazione cure negli ospedali
    - dimensioni: ospedali, dottori, diagnosi, tempo, pazienti, assistenti, procedure

Tabelle di copertura

- In alcune situazioni, si ha interesse a modellare fatti che non sono accaduti
- Esempio: quali prodotti in promozione non sono stati venduti?

- Con gli schemi proposti, non e` possibile rispondere a domande di questo tipo
- infatti lo schema per le vendite proposto contiene fatti solo per i prodotti venduti

Tabelle di copertura

- Per ovviare a questo problema, si possono creare specifiche tabelle di copertura che tengono conto di tutte le combinazioni esistenti di un certo insieme di dimensioni
- Le tabelle di copertura sono per definizione tabelle dense e anomale
- Possono contenere un attributo fittizio, il cui valore e` sempre 1, che rappresenta il fatto che una certa combinazione di valori ha permesso il verificarsi di un certo evento

Esempio

- Si consideri il problema delle vendite

- La tabella di copertura contiene un record per ogni prodotto in promozione ad una certa data
- si puo` considerare ad esempio granularita` settimanale

Composizione degli schemi

- Lo schema risultante da ogni processo aziendale, può essere visto come lo schema associato ad uno specifico data mart

- in presenza di più data marts, e quindi di più processi aziendali, è necessario potere combinare i fatti e le dimensioni contenuti negli schemi associati a ciascun processo

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

Composizione degli schemi

- Gli schemi associati ai vari processi possono avere dimensioni a comune
- per potere passare dalle informazioni contenute in uno schema alle informazioni contenute in un altro (drill-across) è fondamentale che le dimensioni con lo stesso nome hanno lo stesso significato e contengono gli stessi attributi
  - **dimensioni conformate**
- ciò implica che vincoli su attributi delle dimensioni a comune devono restituire le stesse entità per ogni schema considerato

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

- Si consideri la dimensione prodotto
- i produttori conoscono i prodotti ad un livello di dettaglio maggiore rispetto a quello noto ai venditori
- in entrambi i casi, ci saranno comunque degli attributi in comune, come ad esempio il tipo di prodotto
- drill-across si può applicare solo a questi attributi

Conseguenze dimensioni conformate

- Una singola tabella delle dimensioni puo` essere usata in relazione a diverse tabelle dei fatti
- interfacce utente e dati sono consistenti nell'uso della dimensione
- unica interpretazione attributi

Fatti conformati

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

Data warehouse bus

- Dimensioni e fatti conformati permettono di mettere insieme data mart distinti, in modo da formare un unico data warehouse
- permettono di effettuare operazioni di drill-across
- per questo motivo si dice che rappresentano il bus dell'architettura

Progettazione di un data warehouse

Parte 2

Aggregazione

- In alcune situazioni, non si hanno vincoli su tutte le dimensioni ma solo per alcune
- per le altre, si puo` avere bisogno di dati di sintesi
- 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?

Aggregazione

- In tutti i casi illustrati si ha bisogno di dati di dettaglio per alcune dimensioni e dati di sintesi per altre
- Esempio:
  - nel primo caso, si vogliono dati di dettaglio per ogni magazzino ma si somma rispetto ai prodotti
  - nel secondo caso, si vogliono dati di dettaglio per i prodotti ma si somma rispetto ai magazzini
- l'esecuzione di queste interrogazioni e` molto costosa se viene effettuata sui dati di base

- Idea: **precalcolare aggregati**

Aggregazione

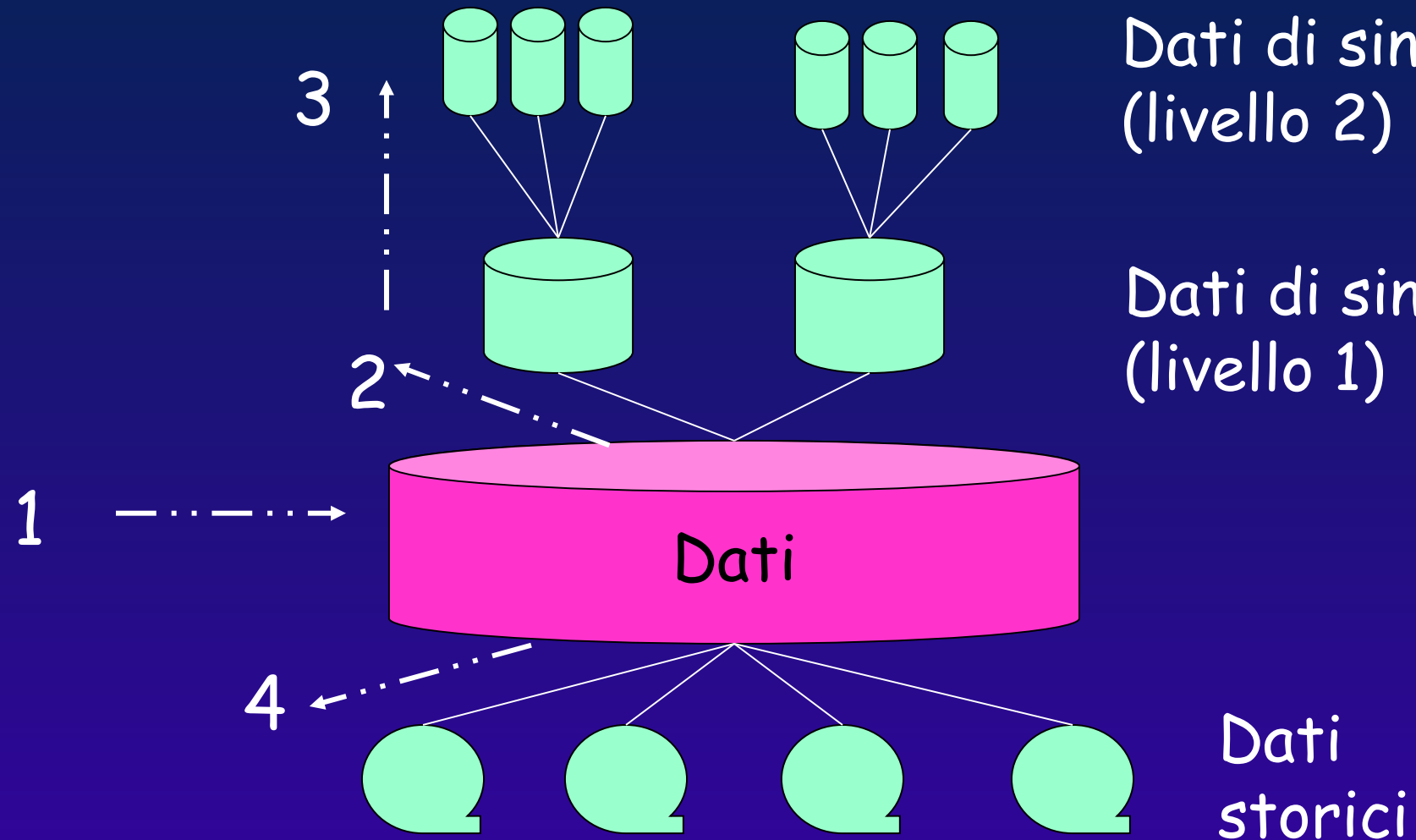
- Un aggregato e` un record di una tabella dei fatti che rappresenta una sintesi di vari record contenuti nella tabella dei fatti di base

- una tabella dei fatti aggregata e` sempre associata ad una o piu` tabelle delle dimensioni aggregate

Aggregazione

- 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

I dati contenuti nel DW



Esempio di aggregati per il DW di vendite

- Categoria prodotto aggregata per magazzino per giorno
- vendite mensili aggregate per prodotto per giorno
- categoria prodotto aggregata per mese

Esempio

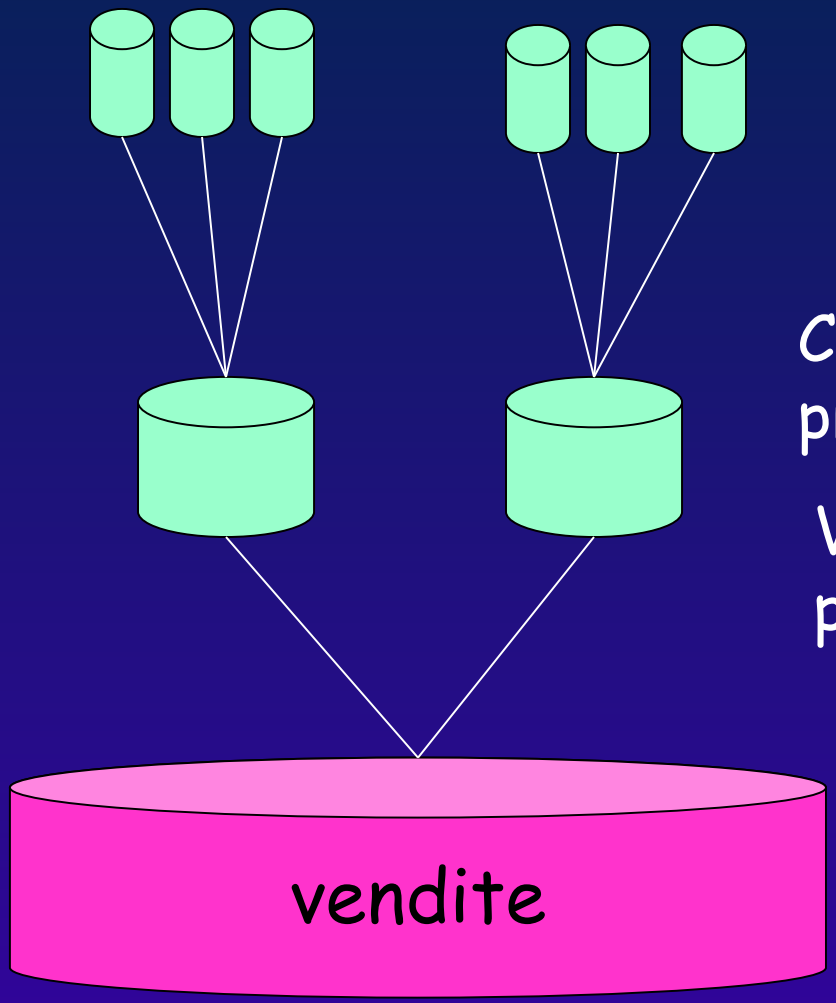
aggregati (livello 2)

Categoria per mese

aggregati (livello 1)

Categoria per prodotto per giorno

Vendite mensili per prodotto per giorno



Due problemi

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

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

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

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
- questo processo implica la generazione di chiavi artificiali per le tabelle delle dimensioni aggregate

Esempio


tempo (dim)
Tempo-k
attributi tempo
Vendite per categoria
Tempo-k
categoria-k
magazzino-k
promozione-k
fatti riferiti alla categoria
categorie (dim)
categoria-k
categoria
dipartimento
promozioni
Promozione-k
attributi prom.
magazzini (dim)
Magazzino-k
attributi mag.

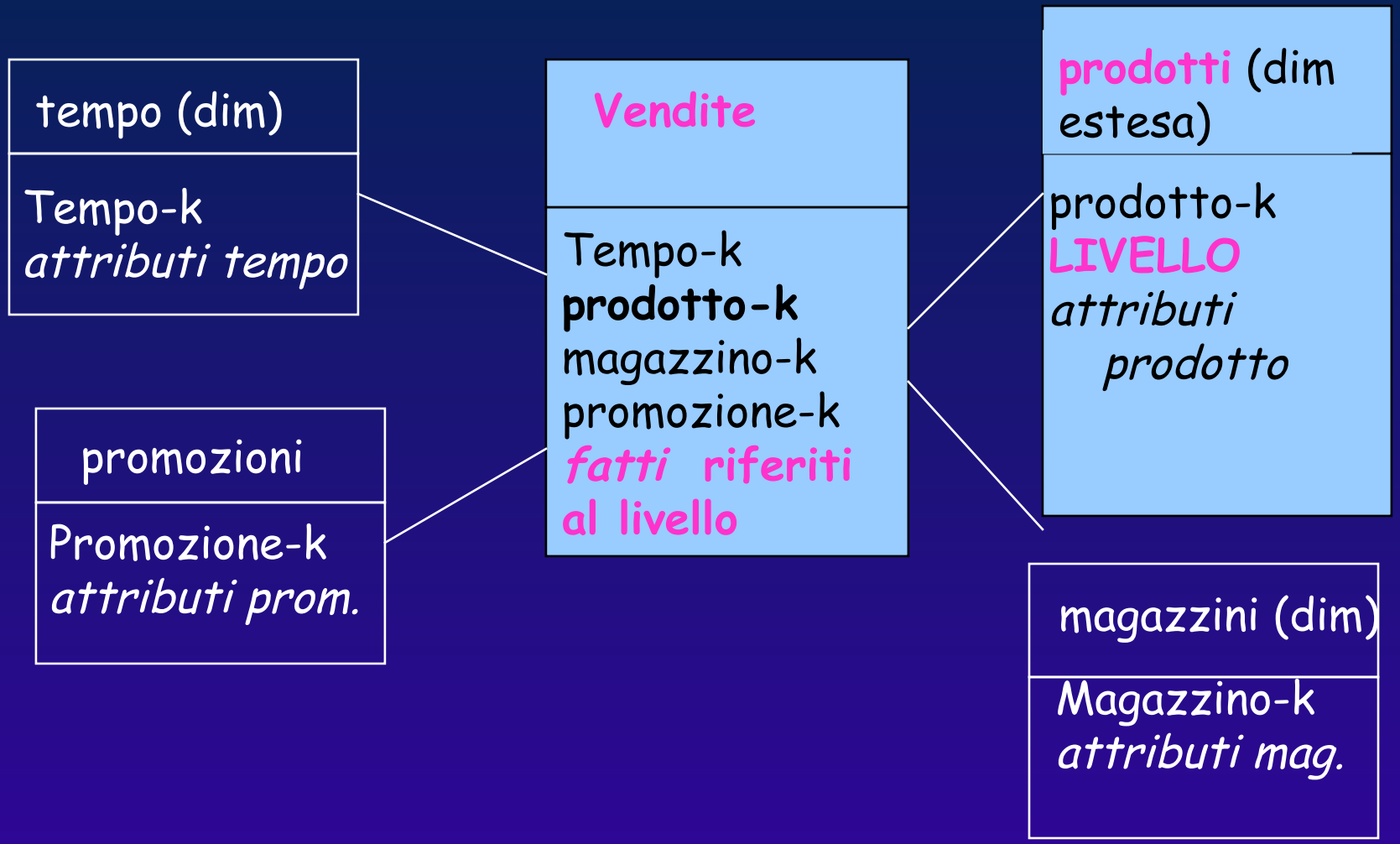
Nuove tabelle dei fatti

- 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 puo` evitare di costruire tabelle aggregate ad hoc

Nuovi campi

- Si inseriscono i fatti aggregati nella tabella dei fatti di base
- si inserisce un campo Livello nelle tabelle delle dimensioni coinvolte nell'aggregazione
- questo campo identifica il campo in base al quale si aggrega
- Esempi:
  - Livello = Base
  - Livello = categoria
- rispetto alla soluzione precedente
  - stesso numero di record
  - stessa necessita` di creare nuove chiavi
  - cambia solo il luogo in cui le nuove informazioni vengono memorizzate

Esempio


tempo (dim)
Tempo-k
attributi tempo
promozioni
Promozione-k
attributi prom.
Vendite
Tempo-k
prodotto-k
magazzino-k
promozione-k
fatti riferiti
al livello
prodotti (dim
estesa)
prodotto-k
LIVELLO
attributi
prodotto
magazzini (dim)
Magazzino-k
attributi mag.

Nuovi campi

- Non tutti gli attributi delle tabelle delle dimensioni hanno senso per ogni valore del campo Livello
- nell'esempio precedente, l'attributo numero-SKU non ha senso se

  Livello= categoria
- a questi attributi deve essere assegnato il valore NULL

Nuovi campi: problema

- Puo` capitare che un'interrogazione possa contare piu` volte la stessa informazione
- Esempio:
  - determina le vendite della categoria carta
  - l'unico vincolo e` sulla categoria
  - se non si mettono vincoli sul campo Livello verranno considerati sia i record con Livello = Base sia Livello = Categoria
    - **ERRATO!!!**
- Questo problema non si verifica se vengono create nuove tabelle, in quanto una sola tabella dei fatti verra` scelta in fase di esecuzione

Esplosione degli aggregati

- L'uso degli aggregati aumenta di molto la dimensione del DB (anche del 300%!)
- questo fenomeno e` particolarmente evidente quando ogni aggregato sintetizza solo pochi record di base
- si consiglia di applicare l'aggregazione solo nel caso in cui ogni aggregato sintetizza almeno 10-20 record di base

Vantaggi dell'uso degli aggregati

- Miglioramento delle prestazioni
- se ben progettati, lo spazio aumenta ragionevolmente (si duplica)
- come vedremo, possono essere utilizzati in modo trasparente all'utente
- se ben progettati, impattano limitatamente il caricamento dei dati

Influenza 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

Esempio query di base

```
SELECT descrizione_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 descrizione_categoria
```

Esempio query aggregata

```
SELECT descrizione_categoria, SUM(vendite)
FROM vendite-aggreg-per-cat, categoria, magazzini, tempo
WHERE vendite-aggreg-per-cat.prodotto-k =
   categoria.prodotto-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 descrizione_categoria
```

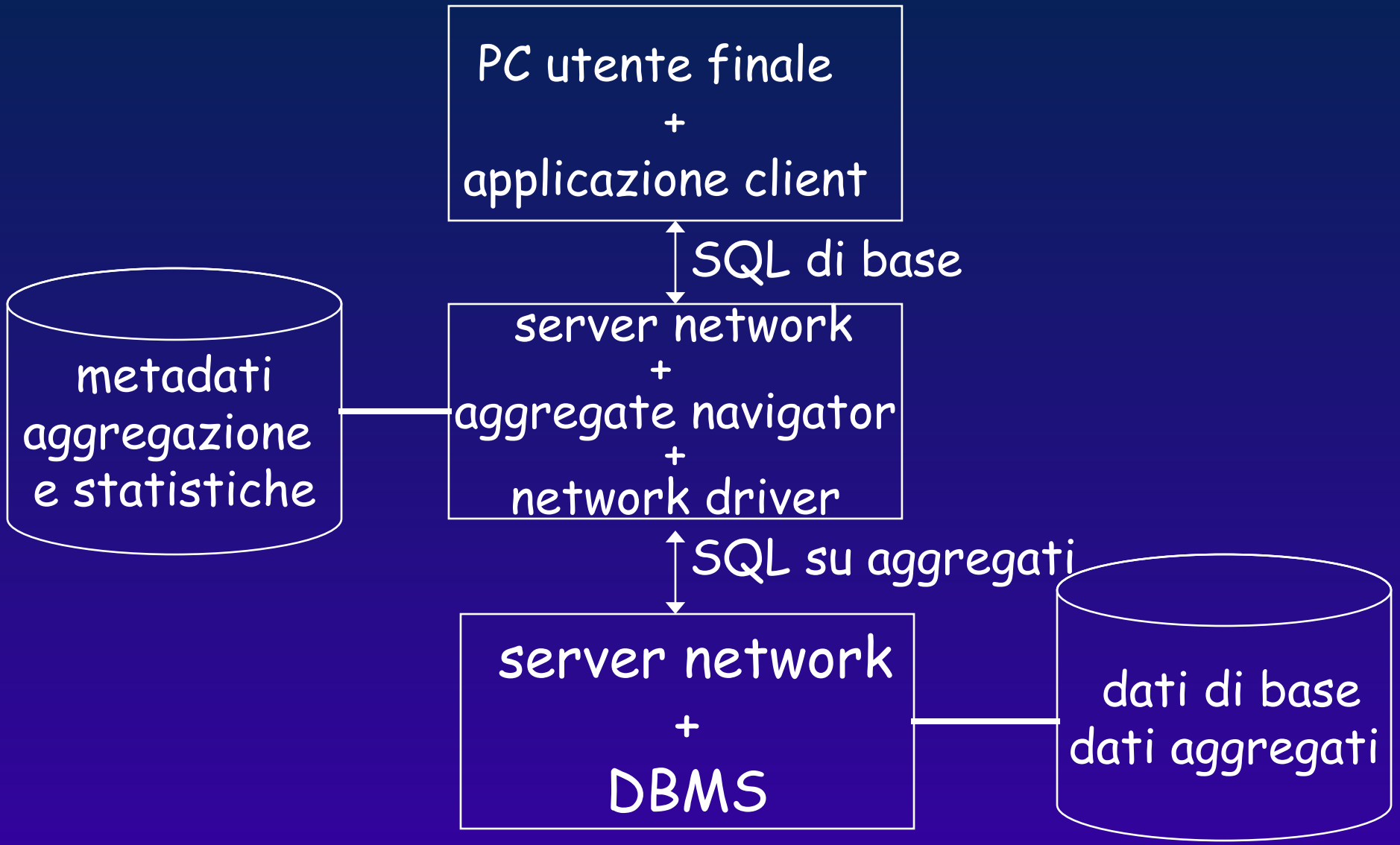
Influenza sul codice SQL

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

Aggregate navigator

- Livello software il cui obiettivo è quello di intercettare le richieste SQL e tradurle utilizzando nel modo migliore le tabelle aggregate
- le richieste SQL si assumono utilizzare le tabelle di base
- si rende trasparente l'uso degli aggregati all'utente finale
- spesso risiede sulla stessa macchina del DBMS

Aggregate navigator



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

Metodologia di progettazione rivisitata

- Identificare il processo aziendale
- identificare la granularita` della tabella dei fatti
- identificare le dimensioni principali
- identificari le dimensioni aggiuntive
- identificare gli attributi delle dimensioni
- decidere come trattare dimensioni che cambiano lentamente
- decisioni circa aggregazione, dimensioni eterogenee, minidimensioni e ogni altra decisione aggiuntiva
- specificare la durata del DB
- specificare la frequenza con cui i dati sono estratti e caricati nel DW

Metodologia di progettazione rivisitata

- La durata specifica per quanto tempo i dati devono essere accumulati del DB
- passato il periodo, e` possibile copiare i dati su memoria terziaria
- la frequenza di caricamento indica, indipendentemente dalla granularita', con quale periodicita` i dati vengono caricati nel DW a partire dai dati operazionali

Progettazione logica

- Durante questa fase, lo schema concettuale del DW viene tradotto in uno schema logico, implementabile sullo strumento scelto
- per il momento: supponiamo che il sistema prescelto sia relazionale

- introdurremo altri sistemi quando descriveremo l'architettura della back room

Progettazione logica

- Il modello logico deve essere il più possibile vicino al modello concettuale, anche se alcune variazioni possono essere rese necessarie dal particolare tool prescelto
- Se il sistema prescelto è un DBMS relazionale, tabella → relazione
- è consigliabile mantenere i nomi degli attributi prescelti durante la progettazione concettuale
- è consigliabile utilizzare un tool di modellazione (es. Oracle Designer 2000)

Progettazione logica

- Un'eccezione alle regole precedenti è dato dall'eventuale uso di views
- le view possono essere create per rendere più semplice l'accesso ai dati
- Esempio:
  - il tool usato richiede uno schema snowflake
  - utilizzando una view, si può fare vedere all'utente uno schema a stella, equivalente a quello snowflake

Progettazione logica

- Idea: si potrebbero usare le view per creare un DW basato su schema 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

Progettazione fisica

- Durante questa fase, si definiscono le strutture di memorizzazione e indicizzazione da utilizzare per l'implementazione del DW
- Aspetti principali:
  - stima dimensione
  - indici
  - scelta risorse di memorizzazione

Stima della dimensione del DB

- La dimensione del DB progettato puo` essere stimata considerando la dimensione della tabella dei fatti
  - (come abbiamo visto, le tabelle delle dimensioni non influiscono pesantemente sulla dimensione globale del DB)
- un'altra misura importante è data dalle dimensioni degli indici costruiti sulla tabella dei fatti
- la stima della dimensione puo` portare a variazioni nello schema

Esempio

- Si consideri l'esempio delle vendite
- Dim tempo: 2 anni per 365 giorni = 730 giorni
- dim magazzino: 300 magazzini, che riportano le vendite ogni giorno
- dim prodotti: 30000 prodotti, di cui 3000 venduti ogni giorno in un dato magazzino
- dim promozioni: un oggetto venduto soddisfa solo una condizione di promozione in un singolo magazzino in un certo giorno
- n. record tabella fatti: 730 x 300 x 3000 x 1 = 657 milioni
- n. attributi chiave: 4
- n. fatti: 4
- dim tabella vendite: 657 milioni x 8 campi x 4 bytes = 21 GB

Stima della dimensione del DB

- Volendo fare un conto preciso:
  - stima dimensione tabella dei fatti +
  - stima dimensione tabelle delle dimensioni +
  - circa 20 MB per metadati +
  - stessa dimensione della tabella dei fatti per aggregati e indici =

Stima della dimensione del DB

- Finora, si e` sempre assunto che i dati considerati fossero atomici
- le tecniche proposte in precedenza sono adeguate per DB con dimensione variabile tra 1 GB e 52 GB, escudendo indici ed eventuali aggregazioni
- in generale, tabelle dei fatti con meno di un bilione di record e piu` piccole di 100 GB possono essere caricate, indicizzate ed efficientemente indicizzate
- esistono pero` DW che richiedono dimensioni di molto maggiori

Esempio

- Attivita`: tracciamento delle chiamate telefoniche
- dimensione tempo: 3 anni = 1095 giorni
- n. chiamate giornaliere: 100 milioni
- n. record tabella fatti: 1095 x 100000000 =109 bilioni
- n. chiavi: 5, n. fatti: 3
- dim. tabella dei fatti: 109 bilioni x 8 x 4 bytes= 3490 GB!!!!!

Indicizzazione

- Tabella dei fatti:
  - indice sulla chiave primaria (composta)
    - B-tree
    - automatico
    - ordine è importante
  - altri indici sui fatti:
    - dipendono dalle tecniche supportate dall'RDBMS
    - in genere non composti

Indicizzazione

- Tabelle delle dimensioni:
  - indice sulla chiave primaria (singola)
    - automatico
  - indici singoli su attributi che si pensa vengano usati in condizioni di join, selezioni o group by
  - ridurre uso indici composti (solo se strettamente necessari)

Indicizzazione

- Si consideri una tabella dei fatti per le vendite
- si supponga che le dimensioni siano: tempo, prodotti, magazzini
- sicuramente verrà creato un indice composto su (tempo, prodotto, magazzino)
- se un'interrogazione vincola solo tempo e prodotto (ad es., seleziona le vendite del prodotto A a partire dal 1 gennaio 2000), per l'esecuzione verranno considerati solo i primi due livelli dell'indice
- ma cosa succede se un'interrogazione vincola solo tempo e magazzino (ad es, seleziona le vendite nel magazzino B a partire dal 1 gennaio 2000)?

Indicizzazione

- In questo secondo caso, il modo più veloce di eseguire la query è applicare una scansione sequenziale
- Soluzioni:
  - creo indice (tempo, magazzini, prodotti)
    - costoso (circa 80% tabella dei fatti)
  - sfrutto eventuali tabelle aggregate

Indicizzazione

- Se il prodotto non è vincolato, vuol dire che si vuole sommare rispetto ai prodotti

- se esiste una tabella dei fatti aggregata con le sole dimensioni tempo e magazzini, l'interrogazione può essere eseguita più efficientemente su questo schema

Indicizzazione

- Quindi, un vincolo debole o l'assenza di vincolo in una tabella dei fatti di base si trasforma in un vincolo forte in una tabella dei fatti aggregata

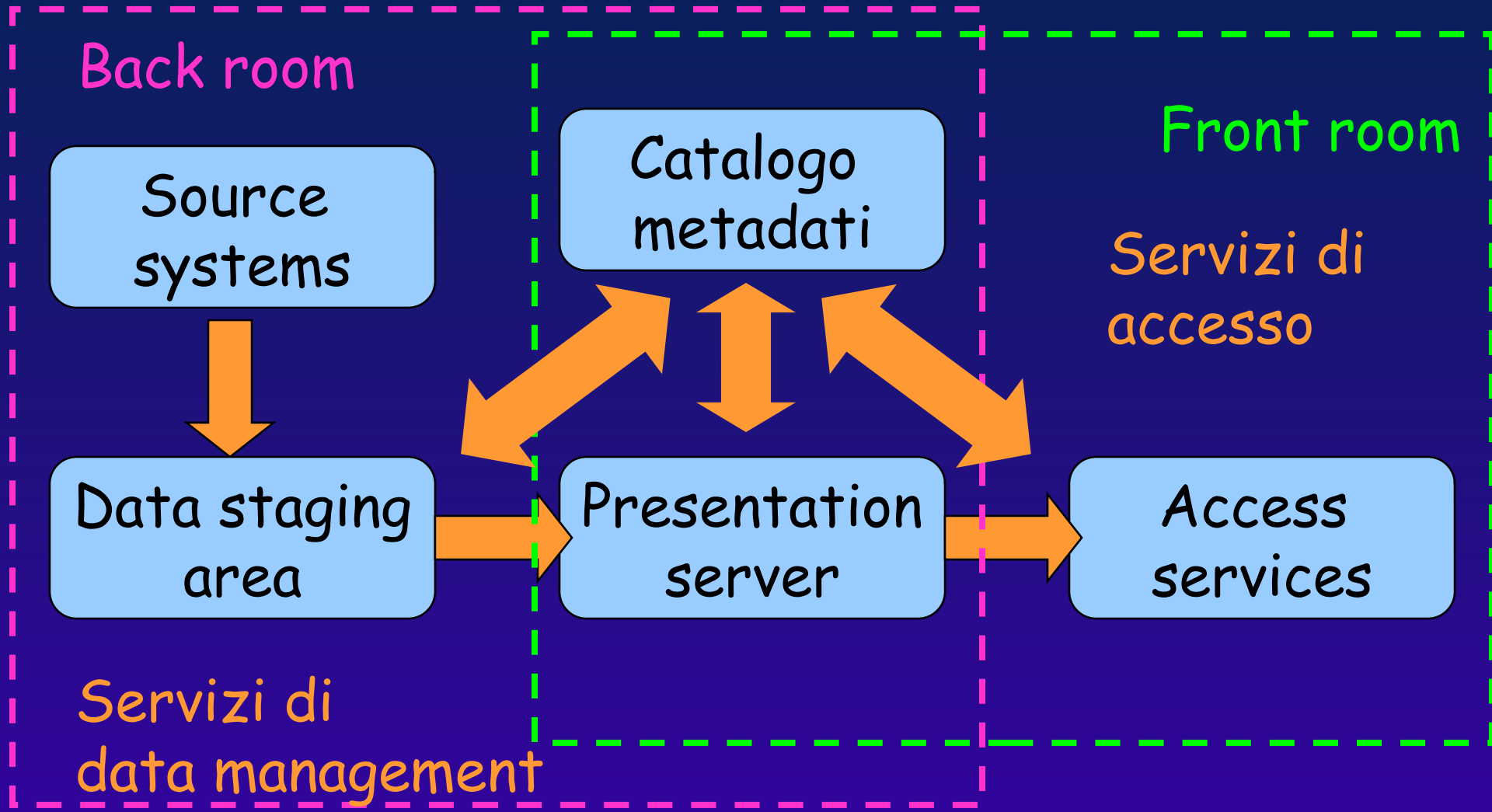
- solo un indice composto deve essere costruito sulla chiave della tabella dei fatti

Scelta risorse di memorizzazione

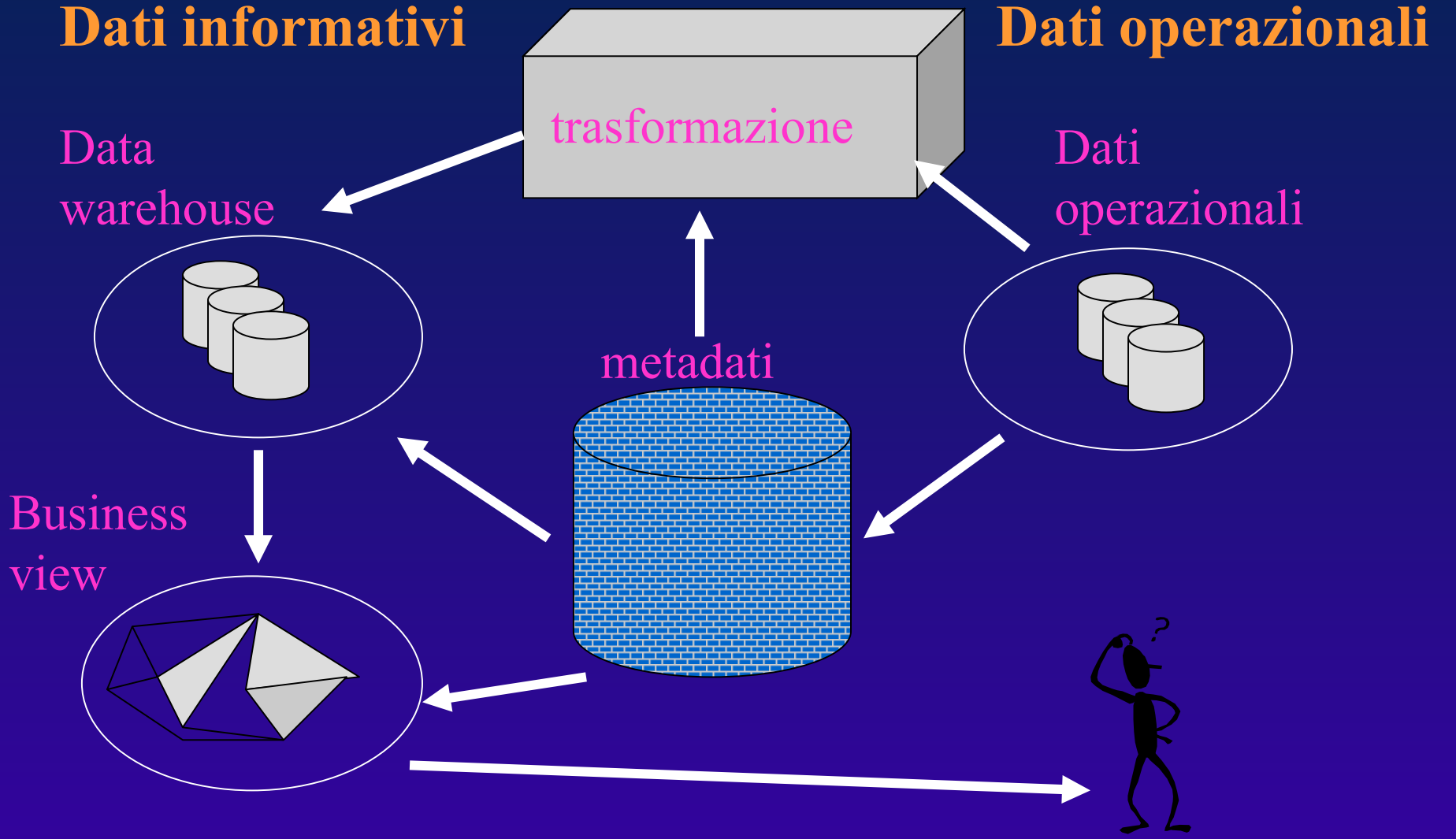
- Parametri importanti:
  - memoria assegnata all'applicazione
  - dimensione blocchi

Architettura di un data warehouse

Architettura di riferimento



Importanza dei metadati

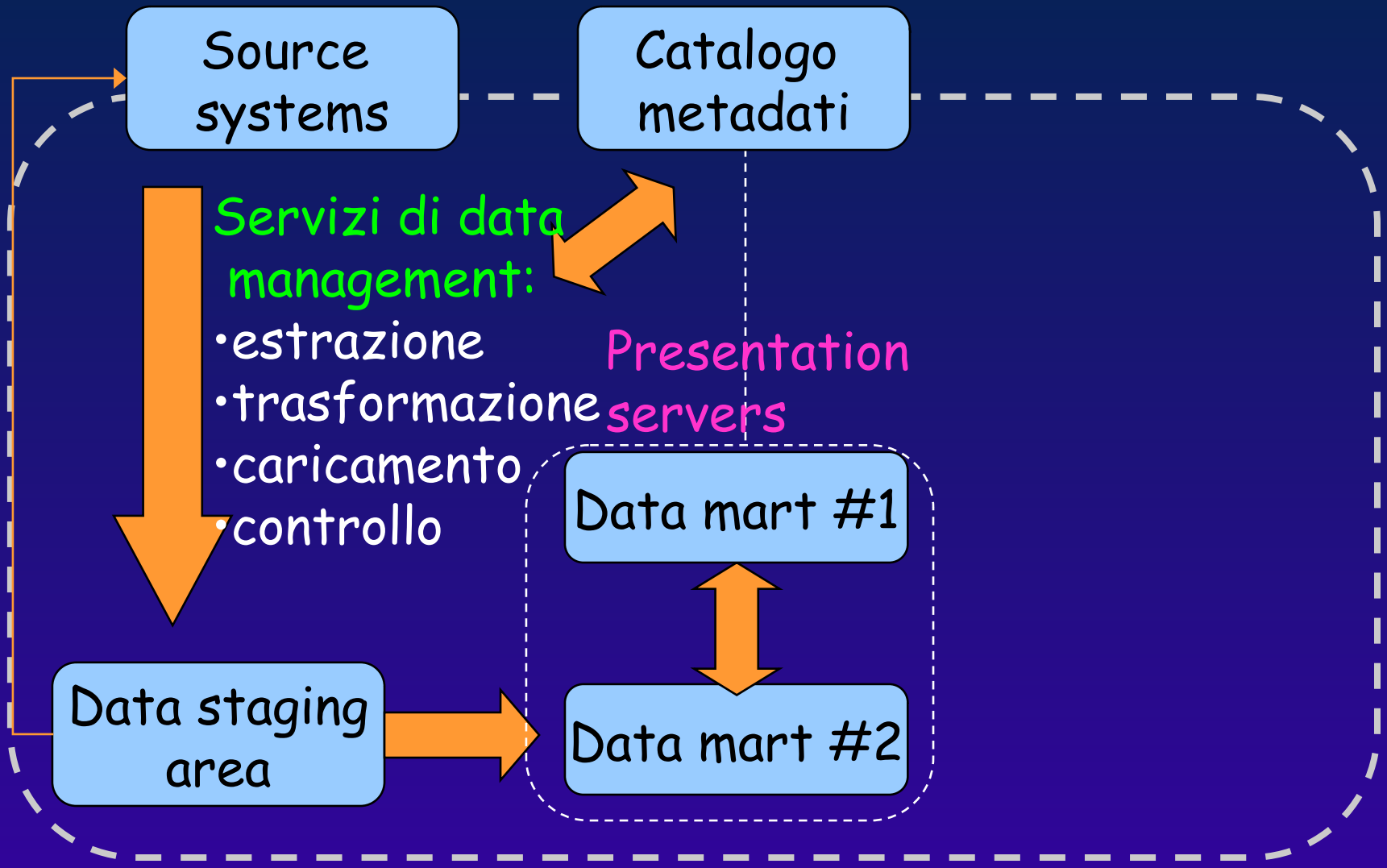


Due ritmi diversi ...

- 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

Architettura della back room

Dove siamo?


Source systems
Catalogo metadati
Servizi di data management:
•estrazione
•trasformazione
•caricamento
•controllo
Presentation servers
Data mart #1
Data mart #2
Data staging area

Architettura logica

- La back room di un sistema di data warehousing gestisce:
  - data stores
  - processi
- nel seguito analizzeremo entrambi

Data stores

- Sistemi sorgente:
  - ogni sorgente di informazione aziendale (database operazionali)
    - interni al sistema: sistemi operazionali (transazionali)
    - esterni al sistema: informazioni demografiche
  - possono anche essere costituiti da un altro data warehouse
  - vari formati (tipicamente flat files)

Data stores

- Area data staging:
  - area in cui i dati sorgente vengono trasformati e viene creato il valore aggiunto del DW
  - Spesso suddivisa su macchine diverse
  - non è necessariamente basata su tecnologia relazionale, ma può anche essere costituita da flat files

Data stores

- Presentation server:
  - piattaforma in cui vengono memorizzati i dati del DW per poi utilizzarli durante le operazioni di accesso
  - in origine: struttura monolitica
  - attualmente: struttura parallela tramite l'uso dei data marts

Data stores

- Data marts:
  - Sottoinsieme del data warehouse
  - spesso, rappresenta la restrizione del DW ad un singolo processo aziendale o ad un gruppo di processi aziendali
  - i dati devono essere conformati, cioè dati con lo stesso nome in data marts diversi devono avere lo stesso significato
  - ogni data mart può risiedere su DBMS diversi

Data stores

- Data marts: ne esistono due tipi:
  - atomici: contengono dati al livello di dettaglio più fine per fare fronte alla maggior parte dei processi aziendali (dati comuni a più processi)
    - costruiti utilizzando il modello dimensionale
    - possono contenere dati aggregati
    - tecnologia relazionale
  - aggregati: contengono dati utili ad un singolo processo aziendale
    - costruiti utilizzando il modello dimensionale
    - possono contenere dati aggregati
    - tecnologia relazionale o multidimensionale

Data stores

- Data warehouse:
  - L'insieme dei dati interrogabili
  - unione di tutti i data marts (possibile se i dati sono conformati)
  - progettato in base ad un approccio dimensionale e non in base al modello E-R
  - il DW è aggiornato perdiodicamente, per mantenerlo consistente con i dati operazionali, che possono variare nel tempo

Servizi principali

- Estrazione dati
- trasformazione
- caricamento e indicizzazione
- servizi di controllo
- controllo di qualità
- pubblicazione dati
- data feedback

Estrazione

- Estrazione dei dati dai sistemi sorgenti
- copia di parte di essi nell'area data staging
- necessità di interagire con diverse piattaforme e formati
- problematiche significative:
  - tipi di estrazione
  - replicazione

Tipi di estrazione

- Caricamenti incrementali
- transazioni evento
- rimpiazzamento completo

Caricamenti incrementali

- Esiste indicatore che specifica quando effettuare il caricamento
- ad esempio: per snapshot mensili, una volta al mese
- si carica solo ciò che è stato inserito a partire da una certa data

Transazioni evento

- Viene utilizzata quando si devono identificare diversi tipi di aggiornamento (inserimenti, aggiornamenti, cancellazioni…)
- serve un time stamp per capire quali record sono stati modificati e quando
- si cambia tutto giò che è cambiato a partire dal caricamento precedente

Rimpiazzamento completo

- Non ci si preoccupa di cosa è cambiato ma si ricaricano completamente i dati
- utilie eseguirlo se la sorgente è di piccole dimensioni
- altrimenti, utile eseguirlo comunque circa 3 volte all'anno

Replicazione

- Per ridurre i tempi di estrazione, può essere utile applicare una strategia di replicazione
- permette di ridurre l'overhead dell'estrazione ed è utile in fase di recovery

Replicazione

- Nel caso di due copie:
  - una copia può essere utilizzata per effettuare caricamenti durante il giorno
  - di notte, si mantiene una copia attiva per le interrogazioni e si completa il caricamento sull'altra copia
  - a caricamento terminato:
    - buon fine: si copia la copia aggiornata sull'altra
    - altrimenti: si copia la copia non aggiornata sull'altra

Trasformazione

- Trasformazione dei dati estratti in dati significativi per il data warehouse:
  - pulizia, cioè eliminazione di conflitti, inserimento di elementi mancanti, formato standard, eliminazione dati non significativi
  - identificazione dimensioni che cambiano lentamente e generazione chiavi
  - check di integrità referenziale
  - denormalizzazione
  - conversione tipi di dato e valori nulli
  - generazione di dati aggregati (spesso esternamente al DBMS)
  - trasformazioni dipendenti dal tool che si intende utilizzare

Caricamento e indicizzazione

- Copia dei dati trasformati nei vari data marts in modalità batch
- indicizzazione dei nuovi dati:
  - si consiglia un'indicizzazione bulk (cioè complessiva al termine del caricamento) per le tabelle delle dimensioni a seguito dell'alto numero di indici che possono essere stati creati su queste tabelle

Servizi di controllo

- Controlla l'intero processo e genera statistiche (metadati)
  - definizione dei processi
  - schedulazione dei processi
  - monitoraggio
  - trattamento eccezioni
  - trattamento errori
  - notifica

Controllo di qualità

- Verifica consistenza dei dati caricati
- tecniche applicabili:
  - controllo totali con sistemi di produzione
  - confronti unità periodo precedente e attuale (ad esempio, si contano i magazzini e si aggiunge una piccola variazione addittiva)

Pubblicazione e data feedback

- Pubblicazione: Comunicazione dell'avvenuto upload agli utenti
- Data feedback: modifica di dati operazionali riconosciuti come errati durante l'esecuzione dei vari processi

Infrastrutture

- Finora: architettura da un punto di vista logico
- Adesso: architettura dal punto di vista delle infrastrutture necessarie per l'implementazione dell'architettura logica

Parametri che influenzano le scelte

- Dimensione dei dati
- dinamicità del sistema
- numero utenti
- numero di processi aziendali
- natura dell'utilizzo
- software disponibile
- risorse economiche

Sistema di gestione dei dati

- DBMS operazionale: in genere relazionale

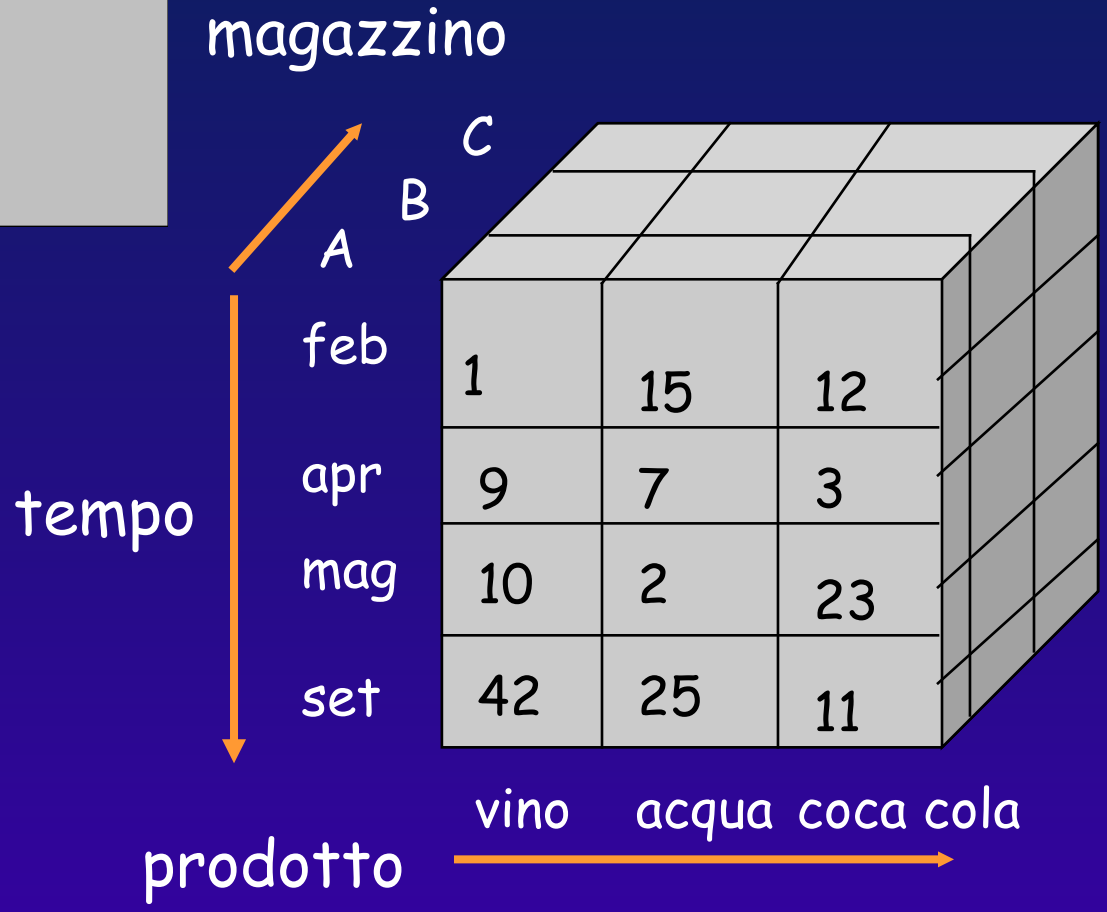
- DBMS informativo: - relazionale
  - multidimensionale

DBMS relazionali

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

DBMS multidimensionali

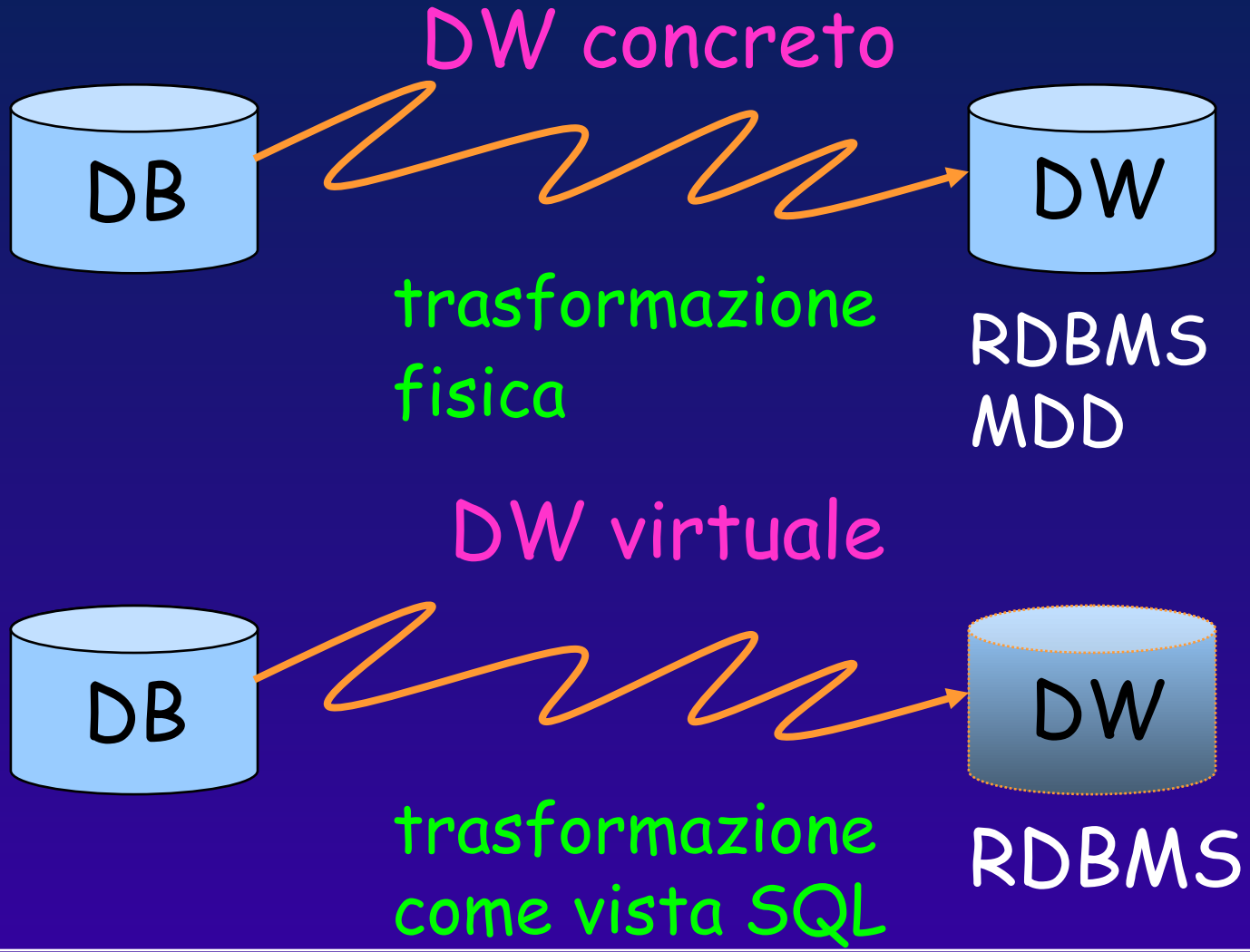
vendite	prodotto	mese	magazzino
1	vino	febbraio	A
2	acqua	febbraio	B
3	coca cola	aprile	A
4	acqua	maggio	A
5	acqua	settembre	C
…	…	…	…



DBMS multidimensionali

- Modello dei dati basato su hypercubi
- 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

Approcci alla costruzione del data warehouse



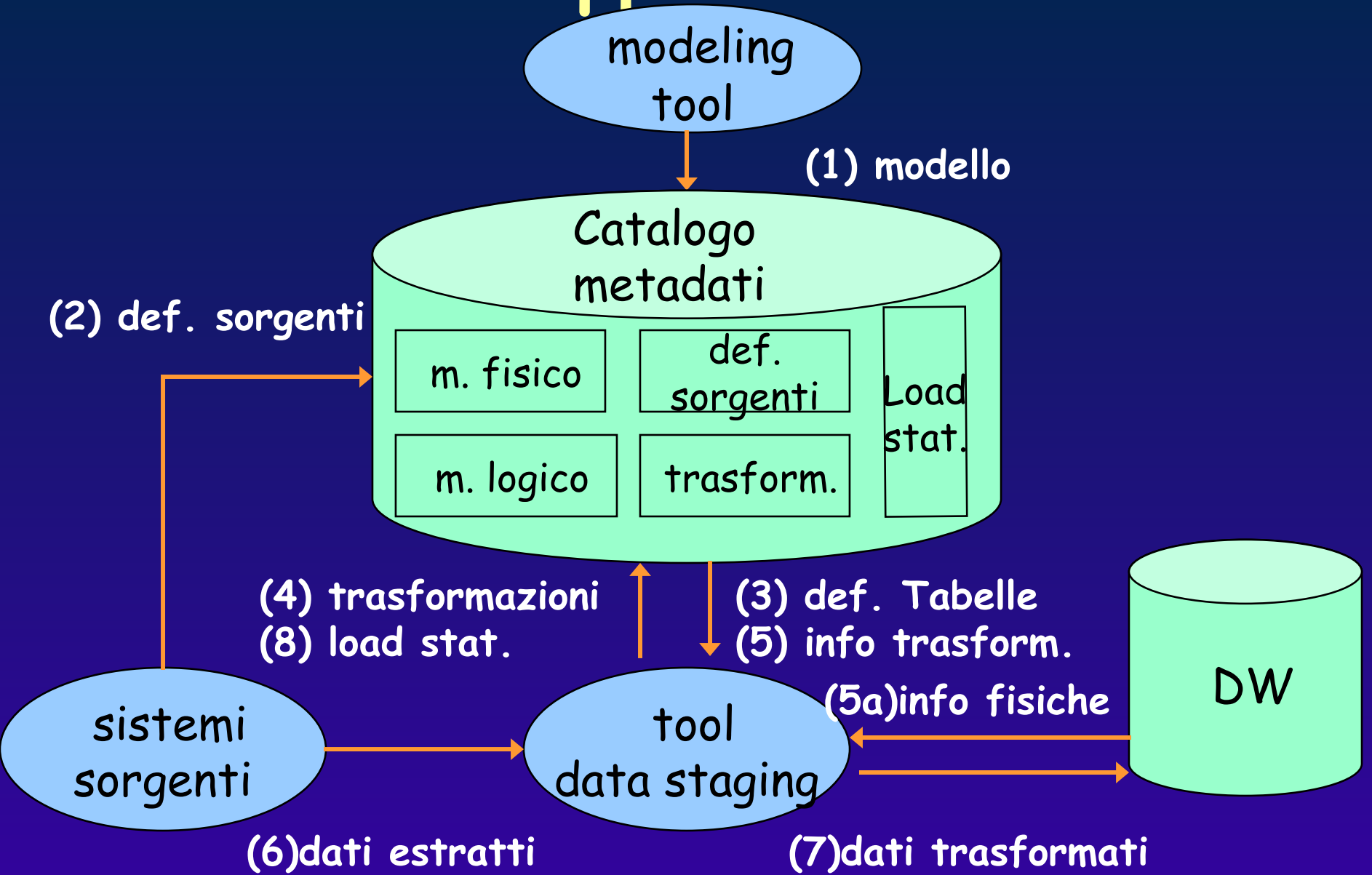
Ruolo dei metadati nell'architettura back room

- Metadati che dipendono dai processi che vengono eseguiti:
  - estrazione, trasformazione, caricamento
- aiutano l'amministratore di sistema a costruire il DW
- utili anche agli utenti per capire da dove arrivano i dati
- Non ci sono tool robusti per la generazione e il mantenimento dei metadati
- ogni tool mette a disposizione specifiche funzionalità

Esempi

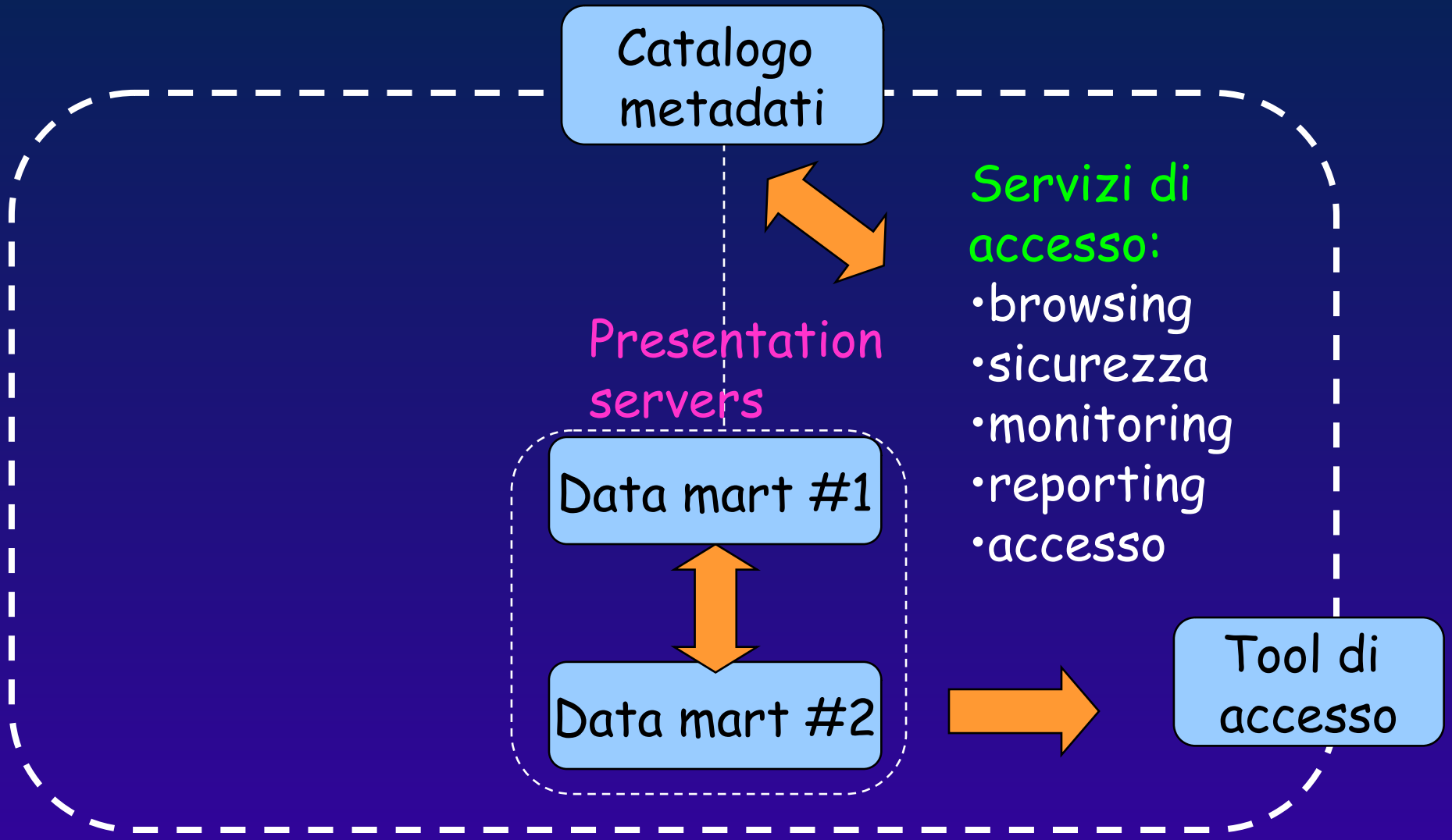
- Schemi sorgente (ad esempio DDl per dati relazionali)
- descrizioni
- frequenza di update dei dati sorgente
- metodi di accesso
- dimensioni e fatti conformati
- dimensioni che cambiano lentamente
- specifiche per la pulizia dei dati e per la trasformazione
- definizione aggregazioni
- aspetti di sicurezza
- indici DBMS
- view DBMS

Approccio


modeling tool
(1) modello
Catalogo metadati
m. fisico
def. sorgenti
Load stat.
m. logico
trasform.
(2) def. sorgenti
(4) trasformazioni
(8) load stat.
(3) def. Tabelle
(5) info trasform.
(5a)info fisiche
DW
sistemi sorgenti
tool data staging
(6)dati estratti
(7)dati trasformati

Architettura della front room

Dove siamo?


Catalogo metadati
Servizi di accesso:
•browsing
•sicurezza
•monitoring
•reporting
•accesso
Presentation servers
Data mart #1
Data mart #2
Tool di accesso

Servizi

- Browsing: deve permettere l'accesso ai dati, anche partendo dai metadati
  - (da una business area ad un folder)
- Sicurezza: autenticazione
- Monitoraggio:
  - performance
  - training nuovi utenti
  - generazione statistiche di uso
  - pianificazione (tempi di caricamento, tempo medio di query)

Servizi

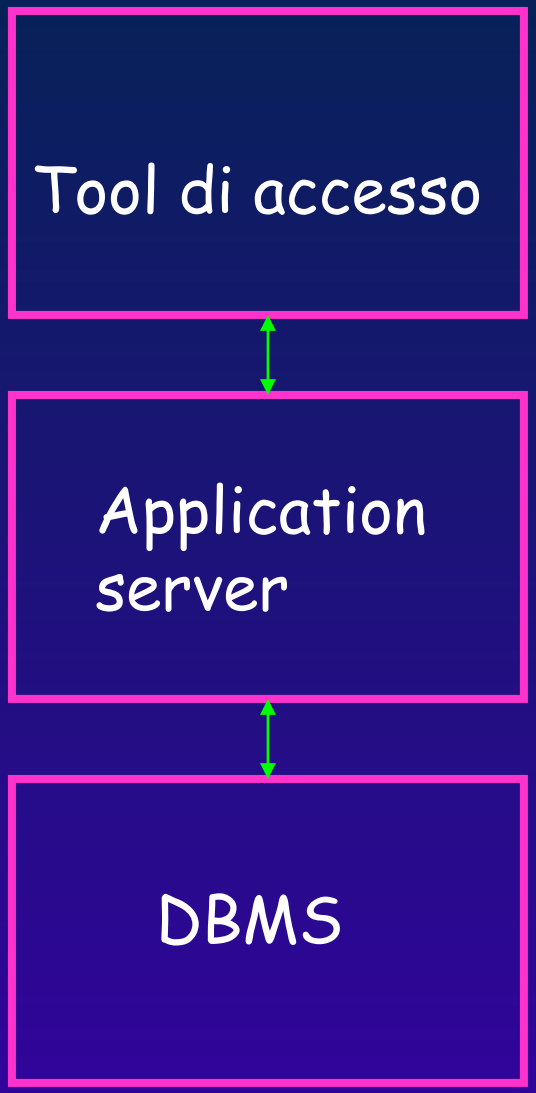
- Query management: capacità di gestire la formulazione della query, la sua esecuzione e la presentazione del risultato
  - semplificazione vista dati all'utente
  - generazione codice SQL di base
  - aggregate navigation
  - regolamentazione query (es. tempo limite per l'esecuzione)

- Reporting: creazione di report mediante una ridotta interazione con l'utente, distribuzione agli interessati, schedulazione delle esecuzioni

Servizi

- Supporto di tool di accesso: tool che permettono all'utente di accedere in modo intuitivo ed altamente espressivo ai dati contenuti nel DW:
  - capacità di effettuare confronti
  - presentazione dati avanzata
  - risposte alla domanda: perche?

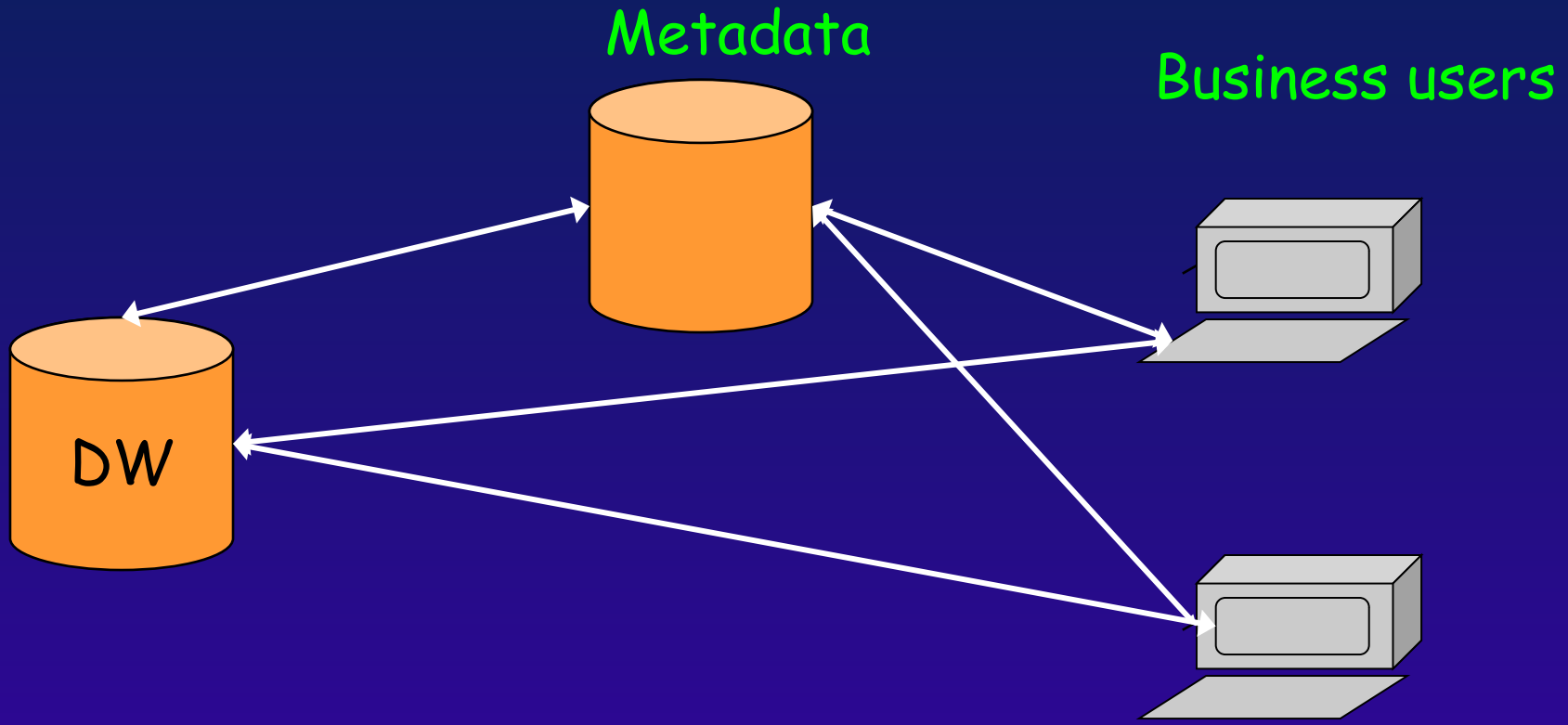
Dove vengono gestiti i servizi?



1. Soluzione tipica ma svantaggiosa (vincolati dal tool)
2. Soluzione ottimale (flessibile)
3. Limita la possibilità di usare piattaforme multiple

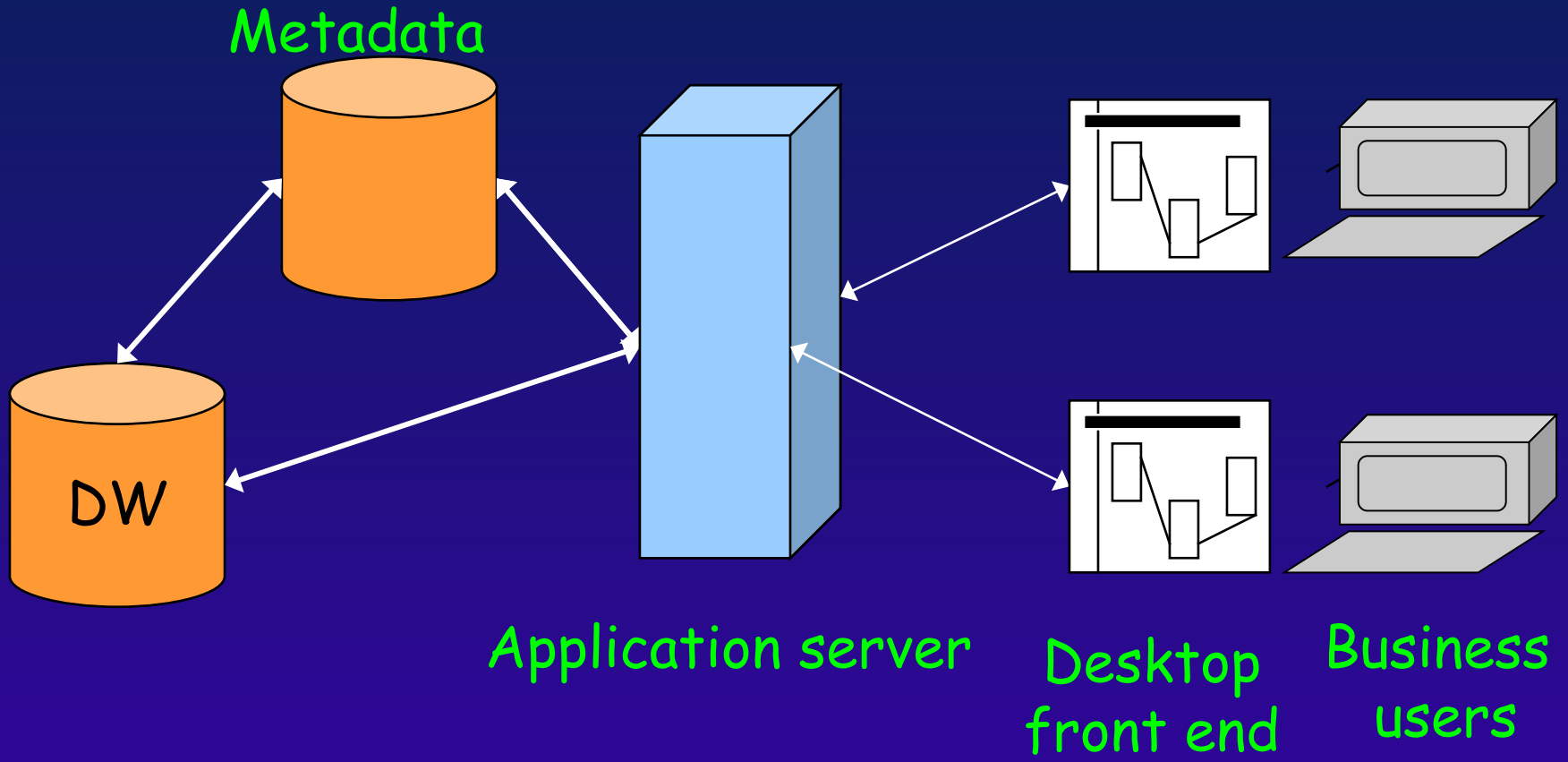
Architettura fisica

Connessione diretta



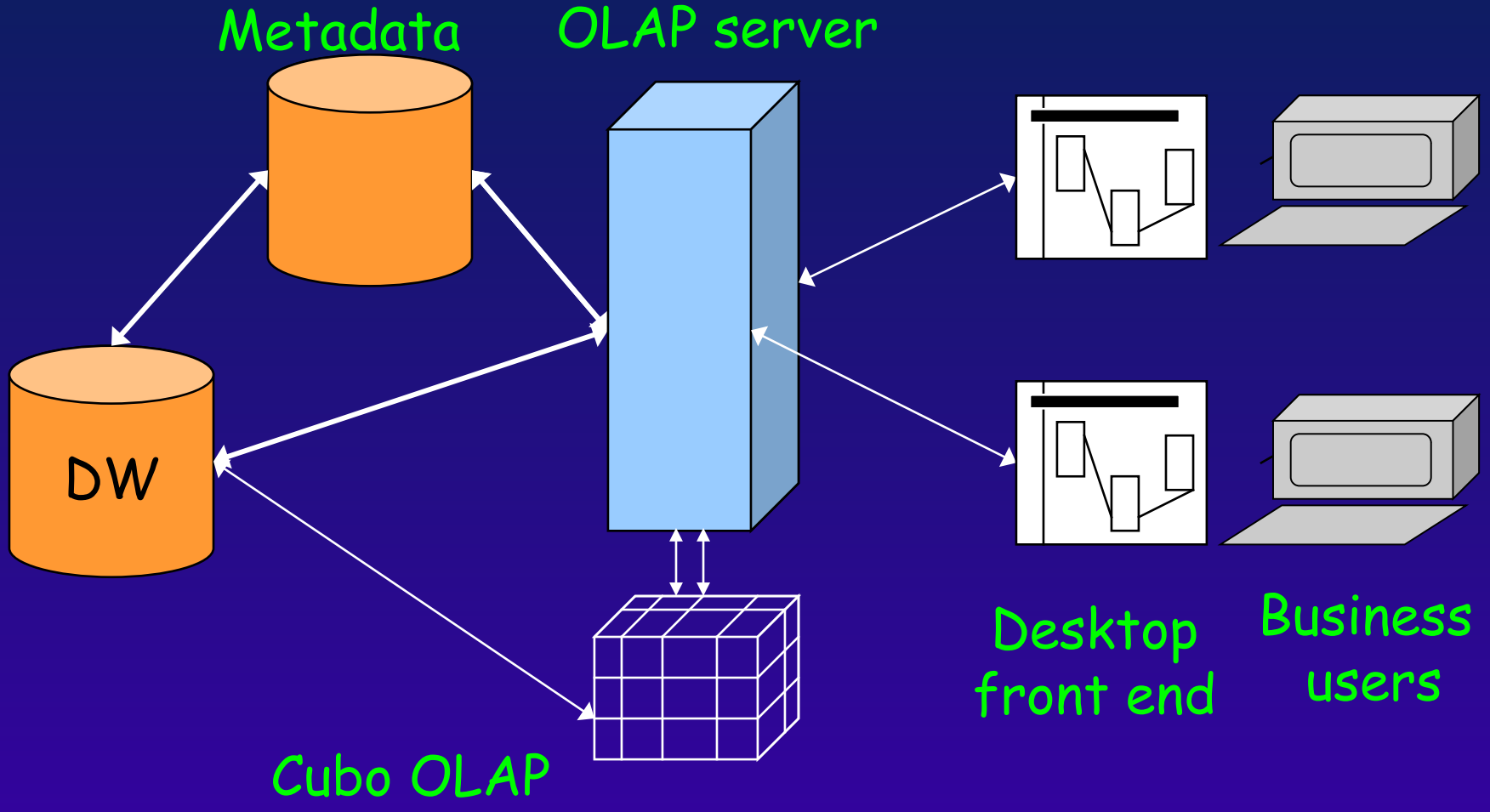
Architettura fisica

ROLAP application layer



Architettura fisica

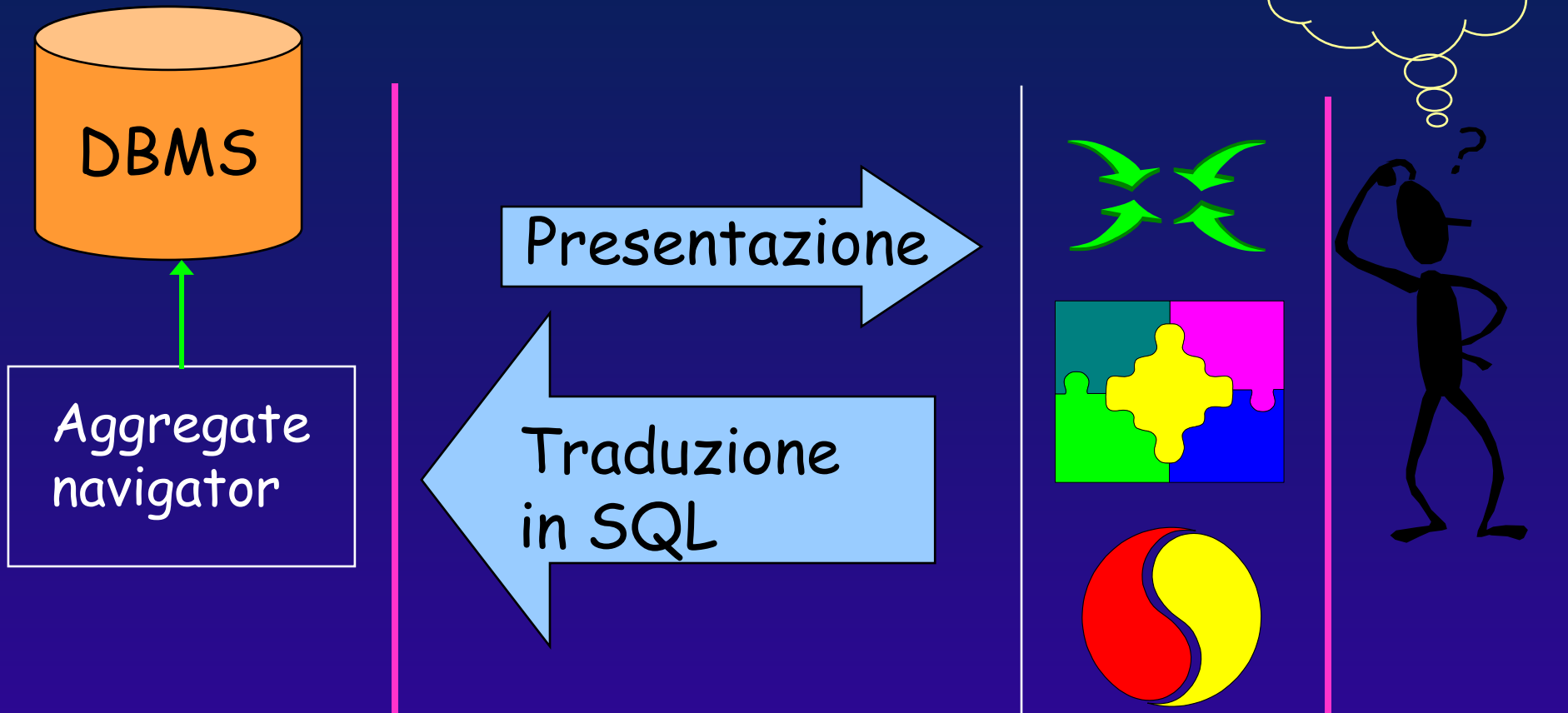
MOLAP application layer



Tool di accesso

- Ad hoc
  - permettono all'utente di specificare le proprie query attraverso interfaccie user-friendly
- tools per la generazione di reportistica
- applicazioni avanzate
  - applicazioni che permettono di applicare operazioni molto sofisticate al DW
    - previsione
    - **DATA MINING**
    - ...

Tool di accesso



Operazione tipica: confronti

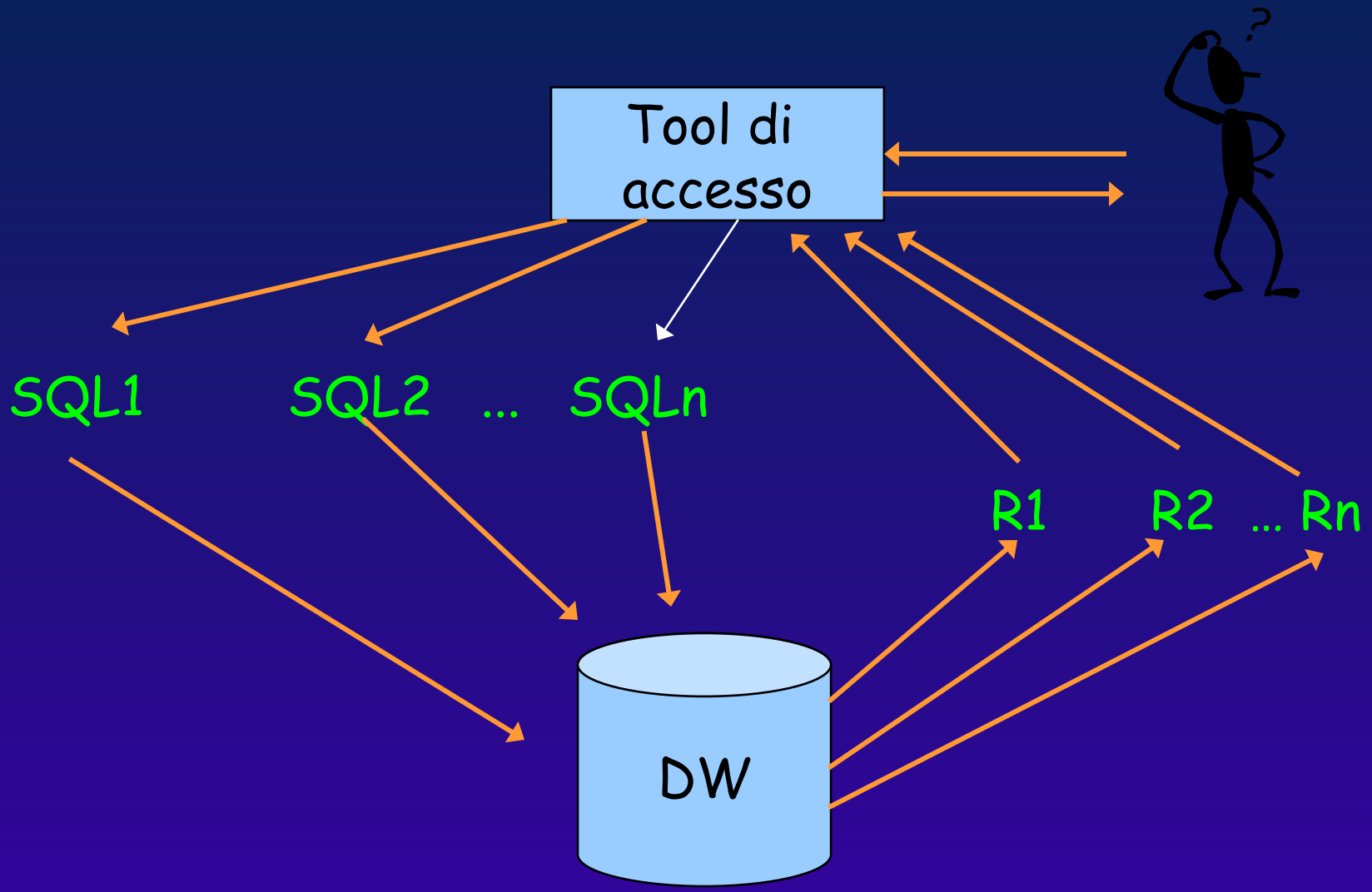
- Dimensioni e fatti analizzati:
  - prodotti, regioni, vendite mese corrente
- crescita % vendite rispetto al mese precedente
  - confronto semplice
- vendite come % su vendite della categoria del prodotto di quel mese (A)
  - confronto multiplo
- variazione vendite come % su vendite della categoria del prodotto, rispetto a quelle del mese precedente
  - confronto multidimensionale
- variazione vendite come % vendite di una certa categoria, rispetto a quelle dell'anno precedente
  - confronto a granularità multipla

Gestione delle interrogazioni

- Le richieste utente, specialmente le comparazioni, in genere sono eseguibili utilizzando query SQL molto complesse
  - difficilmente ottimizzabili
  - complicano l'utilizzo dell'aggregate navigator

- La soluzione in genere è quella di suddividere la query utente in molte query SQL semplici, eseguite indipendentemente e composte direttamente dal tool di accesso

Gestione delle interrogazioni



Caratteristiche dell'interfaccia

- Visibilità struttura dati:
  - dimensioni
  - vincoli
  - fatti
  - stato attuale report in construzione
- possibilità di editare i valori
- possibilità di interrompere l'esecuzione di una query
- interfaccie distinte per gruppi distinti di utenti:
  - interfaccia esecutiva
  - interfaccia di analisi
  - interfaccia di sviluppo

Caratteristiche dell'accesso

- Browsing: possibilità di visualizzare i valori associati agli attributi delle dimensioni e vincolare la ricerca ad un particolare valore

- campi calcolati: possibilità di visualizzare liste di calcoli comuni, da applicare alle dimensioni
  - vendite --> vendite %

Caratteristiche dell'accesso

- drilling down/up:
  - tra gli attributi: si aggiungono/tolgono attributi
  - tra report: si passa da un report ad un altro ad esso collegato
- gestione eccezioni: possibilità di individuare dati anomali, rispetto a
  - valori indicati
  - trend, previsioni
  - determinati eventi

Caratteristiche dell'accesso

- Interazione con aggregati: uso aggregate navigator
- drilling across: possibilità di passare da uno schema all'altro, utilizzando dimensioni e fatti comuni
- totali parziali: possibilità di aggiungere al report totali per gruppi di record

Caratteristiche dell'accesso

- Pivoting: risistemazione righe e colonne nei report
- Ordinamenti: possibilità di ordinare record rispetto a svariati parametri
- Import/export dei risultati in altri tool

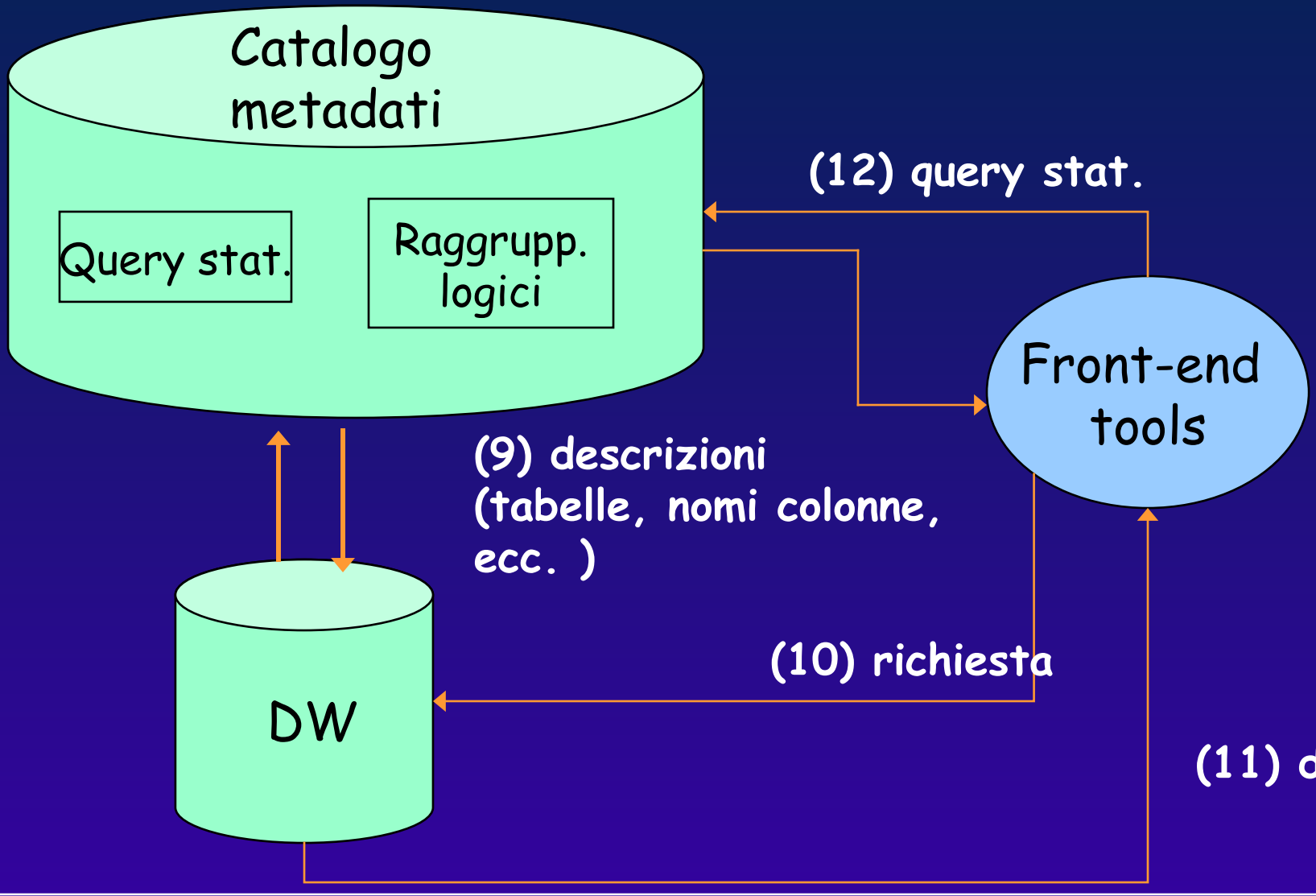
Infrastrutture

- Molto più dipendenti dal tool rispetto alle infrastrutture necessarie alla back room
- Parametri da considerare nella scelta di un tool:
  - caratteristiche server
    - limiti memoria e disco, supporto piattaforme multiple, bottlenecks, ...
  - caratteristiche client
    - supporto piattaforme multiple, web, ...
  - connettività
    - gateways per accedere dati in database diversi dal data warehouse
    - connettività con il DW (ODBC, JDBC,...)

Ruolo dei metadati nell'architettura front room

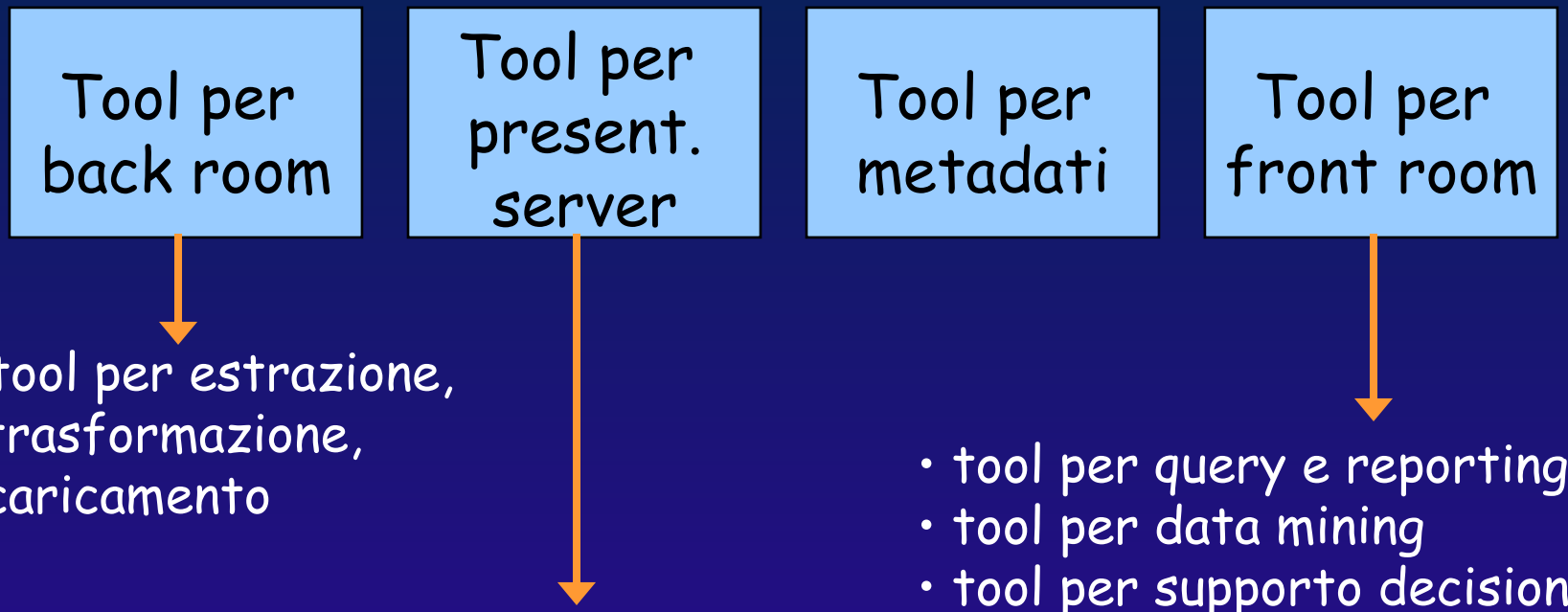
- Metadati che descrivono quali dati sono stati estratti e come devono essere presentati
- Esempi
  - nomi per colonne, tabelle, raggruppamenti
  - definizione dei report
  - documentazione
  - profili sicurezza
  - certificati di autenticazione
  - statistiche relative alla sicurezza
  - profili utente
  - statistiche relative alle query

Approccio


Catalogo metadati
Query stat.
Raggrupp. logici
(12) query stat.
Front-end tools
(9) descrizioni (tabelle, nomi colonne, ecc.)
DW
(10) richiesta
(11) dati

Rassegna dei tool di data warehousing

Classi di tool



In genere

- I produttori di DBMS forniscono in genere tool per ogni componente architetturale.

- Tra questi:
  - ORACLE
  - INFORMIX
  - MICROSOFT
  - SYBASE

La soluzione ORACLE