Concetti di partenza

#database #dbms #transazione #schema #istanze #viste #modello-relazionale #entity-relationship #ddl #dml

Basi di dati

Un insieme organizzato di dati utilizzati per il supporto allo svolgimento delle attività di un ente, gestiti da un DBMS (database management systems). Un sistema è qualcosa di complicato, non è semplicemente uno strumento ma sono tante funzioni unite.

Le basi di dati hanno alcune caratteristiche, una di queste è che sono *grandi*, parliamo di dati transazionali, dati decisionali, dati scientifici che raggiungono centinaia di Terabyte a volte.
es. basi di dati bancari, a transazioni sicure
es. dati scientifici, astronomici

Le basi di dati devono essere *persistenti*.
Sono *condivise*, più persone-processi condividono gli stessi dati.

Il modo di rappresentare i dati esternamente è diverso da quello interno, per poter mantenere l'*indipendenza dei dati* (modello dei dati).
Esistono modelli diversi di dati, ma ci occuperemo solamente del #modello-relazionale.

Si parla d'*indipendenza fisica* per tutti i DBMS,
si parla d'*indipendenza logica* delle tabelle di base.

DBMS

- privatezza
- affidabilità, mettendo in gioco politiche per recuperare dati per malfunzionamenti, con gestione delle *transazioni*
- efficienza, per spazio di memoria e tempo; DBMS con tante funzioni rischiano inefficienza
- efficaci, quello che deve fare viene fatto? risolve i problemi? se sì allora è un buon DBMS per in nostri usi
  I *file system* estendono quelli tradizionali, fornendo più servizi in maniera integrata ("tutto o niente").
  Esiste all'interno dello stesso una porzione che contiene *descrizione centralizzata dei dati*, che può essere utilizzata dai vari programmi
  (catalogo o dizionario), che serve come indice.

Transazione

O vanno tutte a buon fine, oppure vengono riprese.
Un insieme *indivisibile* (atomico) corretto anche in presenza di concorrenza e con effetti definitivi.
es. se al nostro conto corrente fosse addebitata una somma, ma per un errore del DMBS questa venisse duplicata, non sarebbe una bella cosa
Sono concorrenti, sono permanenti.

Schema, istanze e viste

`esempi di tabelle`

insegnamento	docente	aula	ora
analisi	luigi neri	n1	8:00
basi di dati	piero rossi	n2	9:45
fondamenti	luigi rossi	n1	10:00
...	...	...	...

aula	piano	dipartimento
n1	T	scienze matematiche
n2	T	informatica
...	...	...

- **schema**, intestazioni della tabella, descrivono la struttura invariante nel tempo
- **istanza**, i valori attuali che possono cambiare rapidamente, il corpo di ogni tabella

schema esterno → schema logico → schema interno → DATABASE

Una vista è l'aggancio tra due tabelle che hanno una relazione.

`esempio di vista`

insegnamento	docente	aula	ora	piano	dipartimento
analisi	luigi neri	n1	8:00	T	scienze matematiche
basi di dati	piero rossi	n2	9:45	T	informatica
...	...	...	...	...	...

Modelli dei dati

- modelli **logici**
  sono effettivamente utilizzati dagli utenti e programmi per interfacciarsi col DBMS
  #modello-relazionale
- modelli **concettuali**
  hanno scopo diverso, servono nelle fasi iniziali di progettazione del DBMS e servono più che altro alle persone per descrivere concetti del mondo reale
  #entity-relationship (ER)

Linguaggi per DATABASE

- linguaggio testuale interattivo (**SQL**)
- comandi SQL in linguaggio *ospite*
- comandi SQL per interagire con linguaggio ad hoc (per grafici, istogrammi, quello che c'è dentro alla mia base di dati)
- con interfacce grafiche

DDL (data definition language)

per gli schemi

```
# creiamo una tabella nel database 'database_esempio'
USE database_esempio;

CREATE TABLE orario (
        insegnamento CHAR(20),
        docente CHAR(20),
        aula CHAR(4),
        ora CHAR(5)
)
```

DML (data manipulation language)

per le istanze