

Introduzione ai sistemi di basi di dati

Angelo Montanari

Dipartimento di Scienze Matematiche,
Informatiche e Fisiche
Università degli Studi di Udine

Introduzione

Caratteristiche **distintive** di una base di dati:

- **persistenza;**
- **mole;**
- **globalità.**

Caratteristiche **generali** di un sistema informatico:

- facilità d'uso (**indipendenza** dei dati);
- efficienza delle operazioni di accesso ai dati (**strutture di indicizzazione e ottimizzazione** delle interrogazioni);
- **efficacia** (convenienza).

Persistenza dei dati

Per garantire la **persistenza** dei dati nel tempo, la base di dati

- risiede stabilmente nella **memoria secondaria**,
- **esiste indipendentemente** dai programmi che interagiscono con essa per ottenere le informazioni di interesse o per aggiornarla operando inserimenti, cancellazioni e modifiche.

Grandi quantità di dati

Gestione di **grandi quantità di dati**, dove il termine grande ha un significato ben preciso: una quantità di dati che supera la capacità della porzione di memoria primaria a disposizione.

Il sistema deve decidere quali dati caricare e mantenere nella memoria primaria e deve eseguire in modo efficiente le frequenti **operazioni di trasferimento dei dati** da memoria secondaria a memoria primaria, e viceversa.

Nota bene: nonostante la crescita della capacità delle memorie a stato solido usate come memoria primaria, il problema dei trasferimenti tra le memorie persiste, perché negli stessi anni è avvenuto un aumento delle dimensioni delle basi di dati, in buona misura determinato dalla natura sempre più complessa dei dati da gestire (immagini, audio, filmati, ...).

Globalità dei dati

Globalità dei dati: i dati sono di interesse per una pluralità di utenti/programmi e occorre, pertanto, disciplinarne l'utilizzo.

In particolare, devono essere specificati e gestiti i **diritti di accesso** ai dati e garantita la **privatezza** dei dati sensibili.

Inoltre, devono essere **risolti i possibili conflitti** fra utenti diversi che vogliano operare nello stesso momento sui medesimi dati.

Perché non usare il file system?

Due utili confronti: (i) sistemi di basi di dati *vs* programmi e (ii) sistemi di basi di dati *vs* **sistemi operativi**.

Esempio: un'applicazione in ambito bancario.

- ridondanza e inconsistenza dei dati;
- difficoltà di accesso ai dati (proliferazione di programmi ad hoc);
- disomogeneità dei dati;
- problemi di integrità dei dati;
- (non) atomicità delle operazioni di accesso ai dati;
- anomalie causate da accessi concorrenti ai dati;
- problemi di sicurezza/protezione dei dati.

Che cos'è una base di dati?

Base di dati: **collezione di dati** usata per rappresentare informazione di interesse per un dato sistema informativo (Definizione 1).

Base di dati: **collezione di dati** gestita da un DataBase Management System, abbreviato DBMS (Definizione 2).

DBMS: collezione di file interconnessi e insieme di programmi che consentono di accedere e modificare tali file.

Le basi di dati supportano **astrazioni sui dati**. Livelli di astrazione:

- il livello fisico;
- il livello logico/concettuale;
- il livello delle viste.

Modelli dei dati - 1

Un **modello dei dati** è una collezione di strumenti concettuali per descrivere i dati, le loro relazioni e i vincoli di consistenza sui dati.

In ordine cronologico...

- modelli reticolare e gerarchico;
- modelli relazionale ed entità/relazioni;
- modello orientato agli oggetti;
- modelli basati sulla (programmazione) logica;
- modelli ibridi relazionali e orientati agli oggetti;
- modelli dei dati (basati su) XML

Modelli dei dati - 2

Due possibili classificazioni:

- modelli basati sull'**identità degli oggetti** (reticolare, gerarchico, orientato agli oggetti) vs modelli basati sui **valori** (entità/relazioni, relazionale, XML);
- modelli basati sugli **oggetti** (entità/relazioni, orientato agli oggetti, XML) vs modelli basati sui **record** (reticolare, gerarchico, relazionale).

Schemi e istanze

Una distinzione fondamentale: **schema** e **istanza** di una base di dati.

Schemi a diversi **livelli di astrazione**: uno schema fisico, uno schema logico e più sottoschemi (viste).

Progettazione e **schema concettuale**.

Indipendenza dei dati:

- indipendenza fisica dei dati;
- indipendenza logica dei dati.

Linguaggi per basi di dati

Alla distinzione tra schemi e istanze corrisponde la distinzione tra DDL e DML:

- linguaggi per la **definizione dei dati** (DDL): la creazione del dizionario dei dati e l'eventuale definizione delle strutture di memorizzazione e dei metodi di accesso ai dati;
- linguaggi per la **manipolazione dei dati** (DML):
interrogazione, inserimento, cancellazione e modifica dei dati.
Linguaggi di interrogazione e di aggiornamento.
Linguaggi (di interrogazione) procedurali e non procedurali.

Il gestore di una base di dati

Il gestore di una base di dati (**database manager**) è il modulo software che realizza l'interfaccia con i dati fisicamente presenti nella base di dati (low-level data).

Esso supporta le seguenti **funzionalità**:

- interazione con il file system;
- gestione dei vincoli di integrità;
- gestione della sicurezza dei dati;
- backup e recovery;
- controllo della concorrenza.

L'amministratore di una base di dati

L'amministratore di una base di dati (**database administrator**) è il responsabile dei dati e delle relative procedure di accesso.

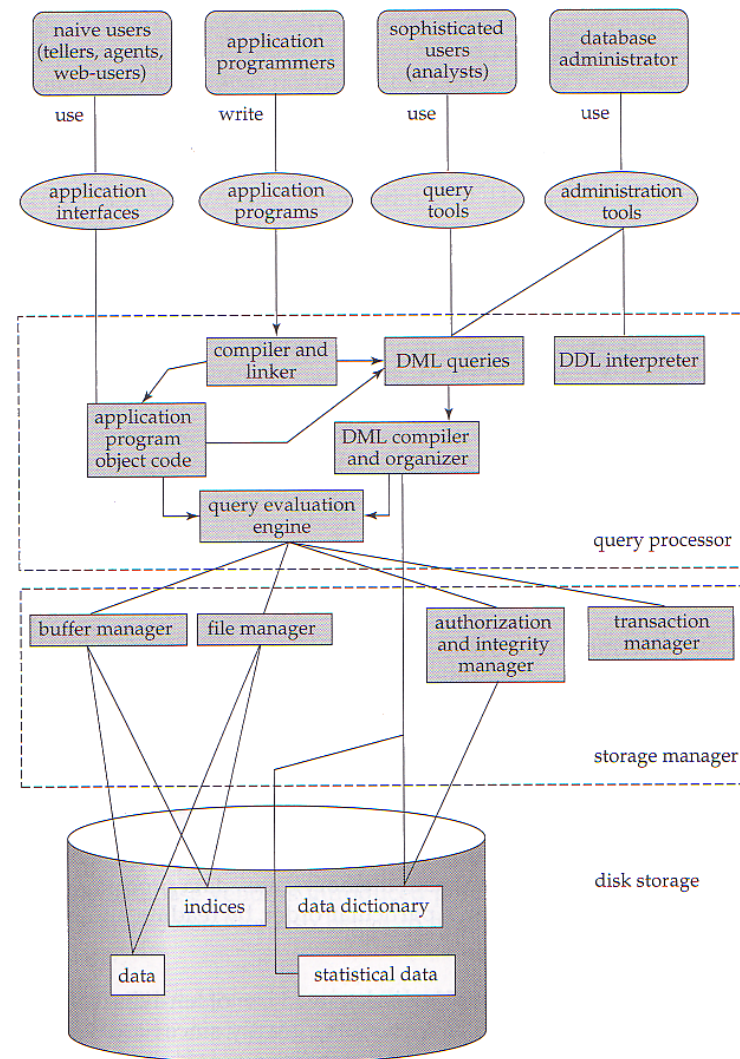
Tale figura svolge le seguenti **funzioni**:

- definizione dello schema;
- definizione delle strutture di memorizzazione e dei relativi metodi di accesso;
- eventuale modifica dello schema logico e/o dell'organizzazione fisica dei dati;
- assegnazione delle autorizzazioni per l'accesso ai dati;
- specifica dei vincoli di integrità.

Utenti di una base di dati

Possiamo distinguere più classi di utenti di una base di dati, sulla base delle diverse possibili **modalità di interazione** col sistema:

- utenti “naïve” (interazione trasparente con la base di dati mediante opportune interfacce applicative);
- utenti che sviluppano programmi applicativi scritti in un linguaggio di alto livello, tipo Cobol, Pascal, C, Java, che necessitano di accedere alla base di dati (uso del DML all'interno di linguaggi host);
- utenti “sofisticati”, che utilizzano direttamente il DML.



La struttura di un DBMS (tratto da A. Silberschatz, H. F. Korth, S. Sudarshan. *Database System Concepts*, 7th Edition, McGraw-Hill, 2019).

Struttura di un sistema di basi di dati (DBMS)

Componenti funzionali:

- storage manager (gestione della memoria);
- query processor (gestione dell'accesso in lettura/scrittura ai dati, coniugando usabilità ed efficienza).

Strutture dati (gestione di migliaia/milioni di megabyte):

- file dei dati e **indici** (strutture ausiliarie che garantiscono un rapido accesso ai dati in corrispondenza di determinate operazioni);
- dizionario dei dati (metadati che definiscono la struttura della base di dati);
- dati statistici (profili delle relazioni, ..).

Il gestore della memoria - 1

Modulo che costituisce l'**interfaccia** tra i dati di basso livello, memorizzati nella base di dati, e i comandi/programmi dell'utente.

In particolare, gestisce l'interazione col gestore del file, traducendo i comandi formulati in un linguaggio per la manipolazione dei dati in comandi di basso livello per il file system (il gestore del file può essere visto come un componente del gestore della memoria).

È responsabile della memorizzazione, del recupero e dell'aggiornamento dei dati.

Il gestore della memoria - 2

Le componenti:

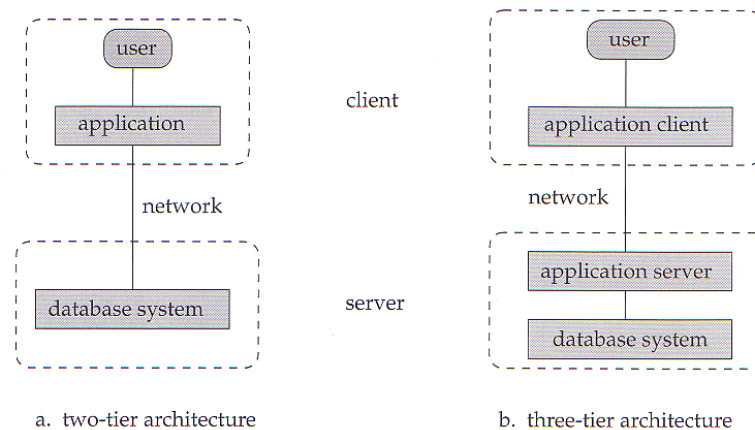
- gestore delle autorizzazioni e dell'integrità dei dati;
- gestore delle transazioni (garantisce le proprietà acide delle transazioni);
- gestore del buffer (responsabile della gestione della porzione di memoria principale riservata al DBMS);
- gestore dei file (responsabile della allocazione dello spazio in memoria secondaria e della gestione delle strutture dati utilizzate per la rappresentazione dell'informazione).

Il query processor

Le componenti:

- l'interprete dei comandi DDL per la definizione dei dati (registra le definizioni nei dizionario dei dati);
- il compilatore dei comandi DML per la manipolazione dei dati (tali comandi vengono tradotti in piani di valutazione costituiti da istruzioni di basso livello per il query evaluation engine; in generale, viene selezionato un piano da un insieme di piani alternativi - **ottimizzazione**);
- il query evaluation engine (esegue il piano di valutazione generato dal compilatore di comandi DML).

Architetture client/server a due e tre strati



Architettura delle applicazioni (tratto da A. Silberschatz, H. F. Korth, S. Sudarshan. *Database System Concepts*, 7th Edition, McGraw-Hill, 2019).

Architettura delle applicazioni - 1

Distinzione fondamentale tra macchine **client**, sulle quali operano gli utenti remoti della base di dati, e macchine **server**, sulle quali opera il sistema di basi di dati.

Architetture a due strati: l'applicazione di basi di dati viene suddivisa in due componenti. Una (l'applicazione in senso stretto) risiede sulla macchina client e invoca le funzionalità dell'altra (il sistema di basi di dati), che risiede sulla macchina server, attraverso opportuni comandi. L'interazione tra client e server viene effettuata per mezzo di interfacce standard per programmi applicativi (ODBC, JDBC).

Architettura delle applicazioni - 2

Architetture a tre strati: la macchina client agisce come semplice front-end e non effettua nessuna chiamata diretta al DBMS. Attraverso opportune form, comunica con un server dell'applicazione, responsabile dell'interazione diretta col DBMS.

La logica dell'applicazione (business logic), che specifica quali azioni eseguire sotto quali condizioni, è incorporata nel server dell'applicazione, invece di essere distribuita fra i vari possibili client.

È una buona soluzione per applicazioni di grandi dimensioni e per applicazioni web.

BIBLIOGRAFIA - 1

R. Elmasri, S.B. Navathe. *Fundamentals of Database Systems* (7th edition). Pearson International Education / Addison Wesley, 2016.

R. Elmasri, S.B. Navathe, *Sistemi di basi di dati. Fondamenti e complementi* (settima edizione). Pearson Italia, 2018.

P. Atzeni, S. Ceri, P. Fraternali, S. Paraboschi, R. Torlone. *Basi di dati* (quinta edizione). McGraw-Hill Education, Milano, 2018.

A. Silberschatz, H. F. Korth, S. Sudarshan. *Database System Concepts* (7th Edition), McGraw-Hill, 2019.

Postgresql (<https://www.postgresql.org>).

BIBLIOGRAFIA - 2

A. Albano, G. Ghelli, R. Orsini. *Fondamenti di basi di dati*. Zanichelli, Bologna 2005.

J.D. Ullman. *Principle of Database and Knowledge Base Systems I-II*. Computer Science Press, Rockville, Maryland, 1989. (Trad. it. *Basi di dati e di conoscenza*. Gruppo Editoriale Jackson, Milano, 1991.)

S.Abiteboul, R.Hull, V. Vianu. *Foundations of Databases*, Addison-Wesley, 1995.

C. J. Date. *An Introduction to Databases Systems* (7th edition), Addison-Wesley, 2000.

L. Welling, L. Thomson. *MySQL Tutorial*, Pearson Education Inc., 2004.