

Basi di Dati (BD): Lezione 2

Concetti e Architetture di un Sistema di BD

Outline

Modelli dei Dati

Categorie

Schemi, Istanze e Stato di una BD

Architettura a Tre Livelli ed Indipendenza dei Dati

DBMS: Linguaggi ed Interfacce

Ambiente di un Sistema di BD

Architetture Centralizzata e Client/Server per i DBMS

Classificazione DBMS

Modelli dei Dati (I)

Modello dei Dati

Un **modello dei dati** e' un insieme di **concetti per descrivere** la **struttura di una BD** e le **operazioni** di manipolazione dei dati

Modello dei Dati: Struttura di una BD

per **struttura di una BD** si intendono i **tipi di dato**, le **associazioni** tra i dati, ed i **vincoli** che dovrebbero valere sui dati

Modello dei Dati (II)

Modello dei Dati

La **maggior parte** dei **modelli dei dati comprende anche** un insieme di **operazioni di base** per specificare reperimenti ed aggiornamenti sulla BD

- Oltre alle operazioni di base (inserimenti, aggiornamenti, cancellazioni ...) il **modello dei dati puo' includere inoltre concetti per specificare l'aspetto dinamico di una BD**
- Cio' consente al progettista della BD di specificare un insieme di **operazioni definite dall'utente** (ad esempio, operazione `calcola_media` applicata a `studente`)
- Nel modello relazionale dei dati esiste la possibilità di **associare il comportamento alle relazioni** (triggers, stored procedures)



Categorie di Modelli dei Dati

Modelli dei Dati di **Alto Livello** o **Concettuali**

Forniscono **concetti** che sono **vicini** alle **modalità** di **percezione** dei **dati** degli **utenti** finali

Modelli dei Dati di **Basso Livello** o **Fisici**

Forniscono **concetti** che **descrivono** **dettagli** sulla **memorizzazione** **fisica** dei dati

Modelli dei Dati di **Implementabili**

- Forniscono **concetti** che **possono** essere compresi dagli **utenti finali** ma che **non** sono troppo **lontani** dal modo in cui i **dati** sono organizzati all'interno del calcolatore
- **Nascondono** alcuni **dettagli** di **memorizzazione** dei dati, ma si **possono** implementare direttamente sul calcolatore



Schemi vs Istanze (I)

Qualsiasi sia il modello dei dati e' importante distinguere tra la descrizione della BD e la BD stessa.

Schema di una BD

- descrizione della BD
- viene specificata durante la fase di progettazione della BD

Diagramma di Schema

- rappresentazione grafica di una schema di BD
- descrive solo alcuni aspetti di uno schema

Costrutto di Schema

Ciascun oggetto dello schema (studente, corso, ...).

Schemi vs Istanze (II)

Stato di una BD

- dati della BD in un particolare istante di tempo
- si parla anche di **istanze** di BD

Nella BD, ciascun **costrutto** dello schema ha un proprio **insieme corrente di istanze**

- **Esempio:** il costrutto **studente** conterra' l'**insieme delle singole entita' (record) di ciascun studente** come sue **istanze**

Stato di una BD (continua)

Lo stato di una BD si riferisce al contenuto della BD in un particolare istante temporale

Stato Iniziale di una BD

- Si riferisce allo **stato di una BD**, nel momento in cui la **BD** viene per la prima volta popolata o caricata con i **dati iniziali**

Stato Valido di una BD

- stato della BD che **soddisfa la struttura ed i vincoli specificati nello schema** della BD.

Ancora su Schema di una BD vs Stato di una BD

- lo **schema** di una BD **non cambia frequentemente**
- lo **stato** di una BD **cambia ogni volta che la BD viene aggiornata**

- lo **schema** viene anche detto **intensione**
- lo **stato** viene anche detto **estensione**



Example (Stato di una BD)

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2

A database that stores student and course information.

Example (Diagramma di Schema di BD)

STUDENT

Name	Student_number	Class	Major
------	----------------	-------	-------

COURSE

Course_name	Course_number	Credit_hours	Department
-------------	---------------	--------------	------------

PREREQUISITE

Course_number	Prerequisite_number
---------------	---------------------

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
--------------------	---------------	----------	------	------------

GRADE_REPORT

Student_number	Section_identifier	Grade
----------------	--------------------	-------

Figure 2.1

Schema diagram for the database in Figure 1.2.

Architettura a Tre Livelli

- Proposta per **supportare** le caratteristiche di un DBMS di:
 - **Indipendenza dei dati**
 - **Viste multiple** sui dati
- **utile per illustrare e spiegare l'organizzazione di un sistema di base di dati**



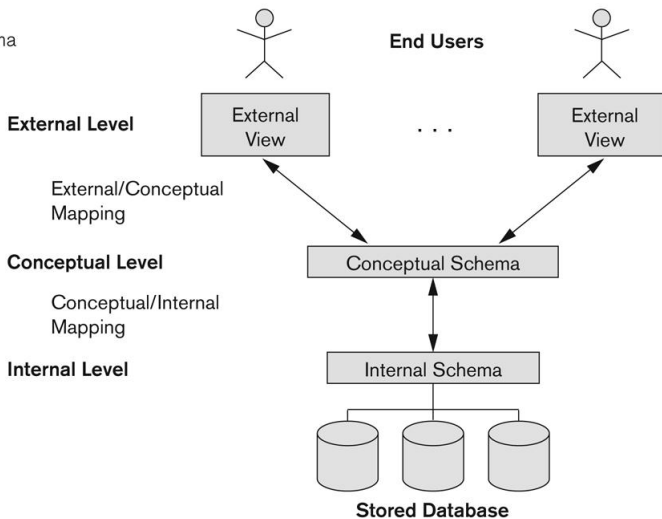
Architettura a Tre Livelli

Definisce **schema DBMS** in **tre livelli**:

1. **Schema Interno**: livello interno—per descrivere la **memorizzazione fisica** dei dati e le strutture di accesso (ad esempio, gli indici)
 - Usa tipicamente un modello dei dati fisico
2. **Schema Concettuale**: livello concettuale—per **descrivere le strutture ed i vincoli sulla BD** per una classe di utenti
 - Usa un modello dei dati concettuale oppure implementabile
3. **Schema Esterno**: livello esterno—**per descrivere le varie viste degli utenti**
 - Si utilizzano gli stessi modelli dei dati usati per lo schema concettuale

Figure 2.2

The three-schema architecture.





Architettura a Tre Livelli

Opera di **mappatura** necessaria per trasformare le **richieste ed i dati** tra i livelli di schema

- Programmi fanno riferimento a schema esterno, e sono mappati dal DBMS verso lo schema interno per essere eseguiti
- Dati estratti dal livello del DBMS interno vengono riformattati per corrispondere alle viste esterne degli utenti (ad esempio, formattazione del risultato di una query SQL per una pagina WEB)

Indipendenza Dati

Indipendenza Dati Logica

Capacita' apporre **cambiamenti a schema concettuale senza dover cambiare gli schemi esterni ed i programmi applicativi associati**

Indipendenza Dati Fisica

- Capacita' di apporre **cambiamenti allo schema interno senza dover cambiare lo schema concettuale**
- Ad esempio, lo schema interno potrebbe essere modificato in seguito alla creazione di nuovi indici per ottimizzare le performance del DBMS

Indipendenza Dati

In un DBMS che supporta l'indipendenza dei dati:

- Quando uno **schema viene modificato ad un livello più basso**, è necessario **modificare soltanto il mapping** con i livelli di **schema più alti**;
- i livelli di schema più alti rimangono invece inalterati.
- Ciò consente di preservare intatti anche i programmi che fanno riferimento agli schemi esterni

Linguaggi DBMS

Data Definition Language (DDL)

- Utilizzato dai DBA e dai progettisti della BD **per specificare lo schema concettuale della BD**
- In molti DBMS, il DDL viene utilizzato anche per definire schemi interni ed esterni
- In alcuni DBMS, vi sono linguaggi speciali per definire schemi interni (storage definition language—SDL) e schemi esterni (view definition language—VDL)

Linguaggi DBMS

Data Manipulation Language (DML)

- Utilizzato per specificare **interrogazioni ed aggiornamenti**
- I comandi del DML possono essere applicati direttamente alla BD (query language)
- Alternativamente, i comandi del DML possono essere integrati in un linguaggio di programmazione (linguaggio ospite), come C, C++, o Java
- E' possibile anche avere a disposizione apposite librerie per accedere ad UN DBMS da un linguaggio di programmazione

Tipi di DML

Linguaggi di Alto Livello o Non-Procedurali

- Ad esempio **SQL**
- Sono **dichiarativi**, ovvero specificano quali dati reperire piuttosto che come procedere all'interrogazione della BD
- set-oriented

Linguaggi di Basso Livello o Procedurali

- reperiscono i dati procedendo record per record
- Sono necessari costrutti di loop e puntatori per reperire insiemi di record

Interfacce DBMS

Interfacce

Diversi Tipi:

- Stand-alone query language interface
- Interfacce per l'utilizzo del DML nei linguaggi di programmazione
- Interfacce user-friendly (menu-based, graphics-based, ...)

Interfacce a DBMS per Linguaggi di Programmazione

Interfacce per programmatori per utilizzare il DML nei linguaggi di programmazione:

- **Approccio Embedded**: ad esempio embedded SQL (per C, C++, ...), SQLJ (per Java)
- **API**: e.g. JDBC per Java, ODBC per altri linguaggi di programmazione
- **Approccio basato su linguaggi nativi**: ad esempio PL/SQL per ORACLE, PL/pgSQL per PostgreSQL. SQL + programmazione strutturata.

Interfacce User-Friendly

- menu-based, popolare per il WEB
- forms-based, definite per utenti naive
- graphics-based (point and click, drag and drop, ...)
- facenti uso di iinguaggi naturali
- ... e varie combinazioni: Ad esempio, menu+form in interfacce WEB a BD

Altre Interfacce

- Input ed output vocale
- Web Browser
- Interfacce parametriche
- Interfacce per DBA:
 - Creare account utenti, gestire le autorizzazioni
 - Definire i parametri di sistema
 - Modificare schemi/ cammini di accesso ai dati

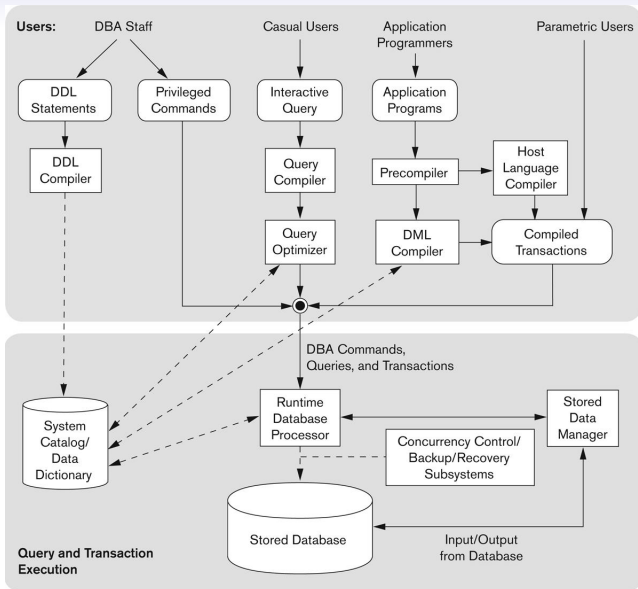


Figure 2.3

Component modules of a DBMS and their interactions.

DBMS: Architettura Centralizzata

Architettura Centralizzata

- Un **singolo sistema** per **DBMS sw, hw, programmi applicativi, gestione interfacce utenti**
- Gli utenti possono anche connettersi in remoto, ma tutta la computazione avviene a livello centralizzato

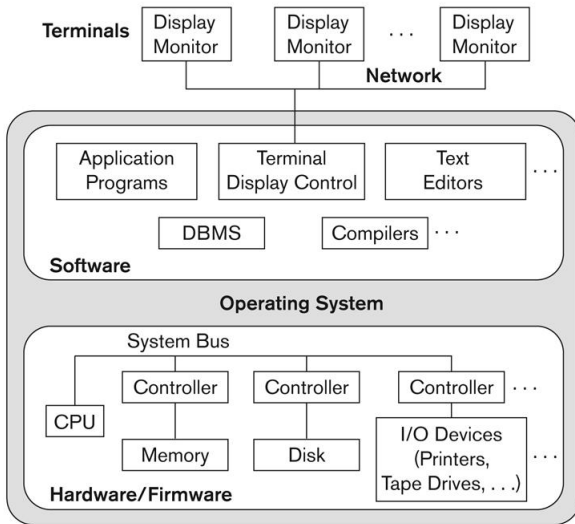


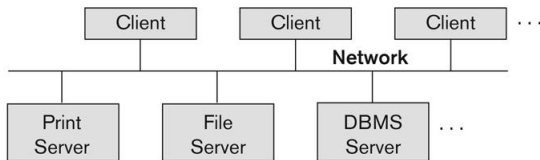
Figure 2.4
A physical centralized
architecture.

Architettura Client-Server

- **Server specializzati** per gestire **funzionalità** specifiche
 - Server di Stampa
 - File Server
 - DBMS Server
 - Web Server
 - Email Server
- Molteplici **macchine client** possono accedere alle risorse fornite da server specializzati
 - Clients forniscono all'utente le **interfacce appropriate** per utilizzare le varie tipologie di **server**
 - Clients possono essere workstation senza dischi, oppure workstation/personal computer con dischi su cui e' installato solo sw client
 - Connessi ai servers tramite una qualche forma di rete (LAN, wireless ...)

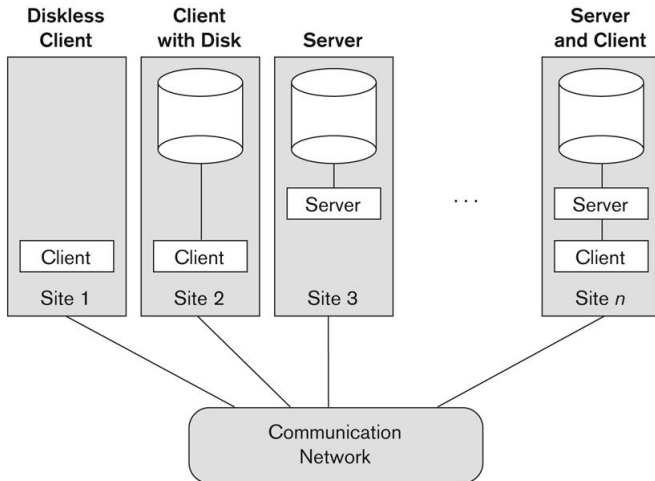
Figure 2.5

Logical two-tier
client/server
architecture.



**Figure 2.6**

Physical two-tier
client/server
architecture.



Architettura Client/Server a 2 Livelli per DBMS

- **DBMS Server:** Tipicamente Funzionalità transazionali e di interrogazione
- **RDBMS:** Si parla di **server SQL**, **server delle interrogazioni**, oppure **server delle transazioni**.
- **Programmi applicativi su client** utilizzano **API** (application programming interface) per **accedere a DBMS**
 - Interfacce standard:
 - **ODBC** (Open Database Connectivity Standard)
 - **JDBC**, standard per il linguaggio di programmazione **Java**

Architettura Client/Server a 2 Livelli per DBMS

- Altri approcci possibili per la divisione di sw tra clients e server
- Livello server puo' comprendere:
 - sw DBMS responsabile della memorizzazione dei dati su pagine del disco
 - sw responsabile del trasferimento in buffer e della memorizzazione nella cache delle pagine del disco
 - ...
- livello client puo' gestire funzionalita' quali:
 - dizionario dei dati
 - interazioni DBMS con i compilatori dei linguaggi di programmazione
 - ottimizzazione globale interrogazioni ...

Architettura Client/Server a 2 Livelli per DBMS

- Altri approcci possibili per la divisione di sw tra clients e server
- In alcuni DBMS orientati agli oggetti altre funzionalita' sono portate sul lato client:
 - dizionario dei dati
 - ottimizzazione globale interrogazioni
 - strutturazione oggetti complessi partendo dai dati nei buffer
 - ...

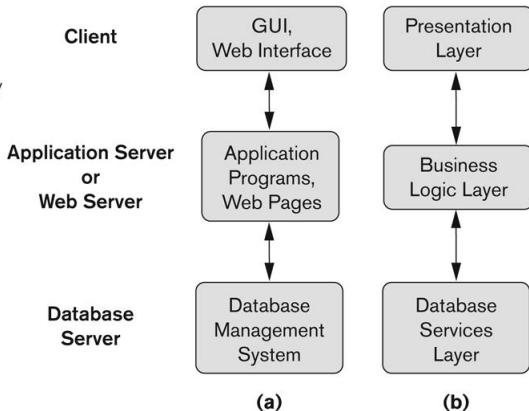
Architetture Client/Server a Tre Livelli

Architettura Client/Server a 3 Livelli

- Nelle applicazioni Web
- Livello intermedio viene detto server delle applicazioni o server Web
 - memorizza regole aziendali (procedure e vincoli) usate per accedere ai dati nel server della BD
 - tramite per passare i dati (parzialmente) elaborati tra BD server e clienti
- Architettura a 3 livelli puo' migliorare sicurezza BD
 - controllo credenziali clienti prima di inoltrare richiesta al server della BD

Figure 2.7

Logical three-tier client/server architecture, with a couple of commonly used nomenclatures.



Classificazione DBMS

Possibili Classificazioni

- In base al **modello dei dati**
 - Tradizionali: relazionali, gerarchico, reticolare
 - Emergenti: ad oggetti, , relazionali ad oggetti
- In base al numero di **siti sui quali e' distribuita la BD**:
centralizzati vs distribuiti
- **General purpose vs special purpose**
- In base a **criteri di costo**
- ...



Modelli dei Dati: Una Prospettiva Storica

Possibili Classificazioni

- In base al **modello dei dati**
 - Tradizionali: Relazionali, gerarchico, reticolare
 - Emergenti: ad oggetti, , relazionale ad oggetti
- Modello **Reticolare**
- Modello **Gerarchico**
- Modello **Relazionale**
- **Orientato agli Oggetti**
- **Relazionale a Oggetti**

Modelli Gerarchico (I)

Modello Gerarchico

- Rappresenta i dati come strutture gerarchiche ad albero
- A partire da un dato padre, si accede ai dati figli da cui essi dipendono
- Definito durante la prima fase di sviluppo dei DBMS (anni 60) ed implementato da IBM e North American Rockwell intorno al 1965
- Non esiste linguaggio standard per il modello gerarchico. Un DML diffuso e' il linguaggio DL/1 del sistema IMS.

Modello Gerarchico (II)

Vantaggi

- Rispecchia natura gerarchica di una molteplicità di domini

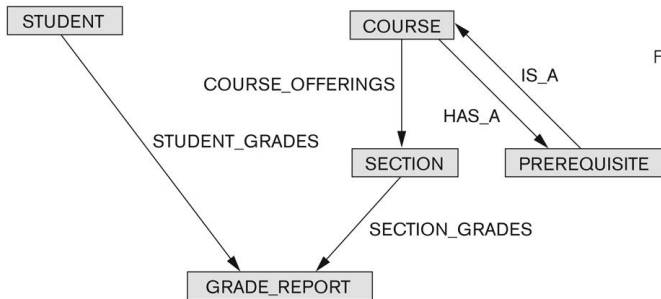
Svantaggi

- Struttura gerarchica impone regole rigide sull'esecuzione di aggiornamenti ed interrogazioni
- Scarso spazio per l'ottimizzazione automatica delle query
- Dipendenza dei programmi dalle strutture
- Non si presta a rappresentare in modo efficiente relazioni N:M
- La definizione di relazioni più generiche richiede l'introduzione di duplicati
- ...

Modell Reticolare (I)

Modello Reticolare

- Primo DBMS reticolare implementato da Honeywell nel 1965 (IDS System)
- Alla base di una varietà di sistemi in voga fino a metà degli anni '80: IDMS (Cullinet, oggi Computer Associates), DMS 1100 (Unisys), IMAGE (HP), VAX-DBMS (Digital Equipment Corporation, poi COMPAQ, oggi HP)
- Supportato dalla CODASYL (Conference on Data Systems Languages/CDASYL-DBTG Report del 1971)
- Rappresenta dati come tipi di record
- Record sono legati tra loro tramite puntatori che permettono all'utente di accedere ai dati più facilmente, senza i vincoli imposti dal modello gerarchico

**Figure 2.8**

The schema of Figure 2.1 in network model notation.

Modello Reticolare (II)

Vantaggi

- Un record puo' avere uno o piu' record padri, evitando problemi di ridondanza
- Ogni nodo puo' essere il punto di partenza per raggiungere un determinato campo
- Permette di modellare relazioni N:M

Svantaggi

- Complesso reticolo di puntatori nella BD
- Scarso spazio per l'ottimizzazione automatica delle query
- ...

Modello Relazionale

Modello Relazionale

- Proposto nel 1970 da E. F. Codd (IBM), primi sistemi commerciali nel 1981-82
- Oggi in molteplici prodotti commerciali (DB2, ORACLE, MS SQL Server, INFORMIX, SYBASE)
- Processo standardizzazione SQL: SQL-89 (SQL1), SQL-92 (SQL2), SQL99 ...
- Ad oggi il prodotto dominante nel mercato dello sviluppo di BD

Modello ad Oggetti

Modello ad Oggetti

- Definisce BD in termini di oggetti, delle loro proprietà, e delle operazioni associate
- Incorporano molte caratteristiche del paradigma ad oggetti (tipi di dati astratti, incapsulamento, ereditarietà ...)
- OODBMS iniziano a diffondersi alla fine degli anni '80
- Inizialmente considerate concorrenti alle BD relazionali, oggi la loro penetrazione complessiva nel mercato dei prodotti BD rimane sotto al 5



Modello Ibrido Relazionale ad Oggetti

Modello Relazionale ad Oggetti

- Trend piu' recente. Inizia con l'avvento di Informix Universal Server
- RDBMS incorporano concetti relativi ai BD ad oggetti, portando al modello relazionale ad oggetti
- Tra gli altri, ultime versioni di Oracle-10i, DB2, PostgreSQL ...
- Standard inclusi in SQL'99