

# Basi di dati

**Maurizio Lenzerini**

**Dipartimento di Informatica e Sistemistica "Antonio Ruberti"  
Università di Roma "La Sapienza"**

Anno Accademico 2003/2004  
Canale A-L

<http://www.dis.uniroma1.it/~lenzerini/didattica/basididati/>

## Il corso di Basi di Dati è rivolto a

### ❑ Nuovo ordinamento (5 crediti, terzo anno)

- ❑ [Laurea in Ingegneria Informatica – canale A-L](#)
- ❑ Collegato a questo modulo è il corso "[Progetto di basi di dati](#)", che viene erogato nel secondo periodo didattico

### ❑ Vecchio ordinamento

- ❑ Il presente corso è uno di due moduli per la [Laurea in Ingegneria Informatica \(e altri corsi di laurea\) – canale A-L](#)
- ❑ Il secondo modulo è "[Sistemi di gestione di basi di dati](#)", che viene erogato nel secondo periodo didattico, e che è anche un corso della [Laurea Specialistica in Ingegneria Informatica](#) (nuovo ordinamento)
- ❑ L'esame si sostiene normalmente, e l'esito si combina con l'esame del secondo modulo per determinare il voto da verbalizzare per l'esame complessivo (la verbalizzazione è unica alla fine del secondo modulo)

## Obiettivi del corso

- ❑ *conoscenza dei DBMS (Sistemi di gestione di basi di dati) relazionali dal punto di vista degli utenti e degli sviluppatori di applicazioni*
- ❑ *conoscenza del modello relazionale e di SQL*
- ❑ *conoscenza del modello Entità-Relazione e di una metodologia di progettazione di basi di dati relazionali basata su tale modello*

Obiettivi del corso “Sistemi di gestione di basi di dati”:

- *conoscenza dei DBMS dal punto di vista di un amministratore di basi di dati e di un progettista di DBMS*
- *conoscenza di problematiche avanzate di gestione di dati in applicazioni informatiche*

## Aspetti organizzativi del corso

**Docente:** Maurizio Lenzerini (<http://www.dis.uniroma1.it/~lenzerini>)

**Ricevimento:**

- Martedì, ore 17:00
- Dipartimento di Informatica e Sistemistica,  
Via Salaria 113, 2° piano, stanza 201

**Sito del corso del Prof. Lenzerini**

<http://www.dis.uniroma1.it/~lenzerini/didattica/basididati/>

**Sito informativo sull'offerta didattica sulle basi di dati**

<http://www.dis.uniroma1.it/~lenzerini/didattica/basididati/offerta.html>

## Aspetti organizzativi del corso

### Lezioni (aula 1):

- Lunedì, ore 10:15 - 11:45
- Martedì, ore 10:15 - 11:45
- Venerdì, ore 12:00 - 13:30

### Esercitazioni in aula

### Esercitazioni al computer

- esercitazioni in laboratorio (verranno annunciate)
- si svolgono al Laboratorio Paolo Ercoli (Via Tiburtina 205)

### Esame composto da

- progettino individuale
- prova scritta
- prova orale

## Aspetti organizzativi del corso

### ❑ Materiale didattico e tutore

- ❑ Lucidi delle lezioni (con qualche giorno di anticipo)
- ❑ Atzeni, Ceri, Paraboschi, Torlone, *“Basi di dati: modelli e linguaggi di interrogazione”*, McGraw-Hill, 2002
- ❑ Tutore: [Domenico Lembo \(lembo@dis.uniroma1.it\)](mailto:lembo@dis.uniroma1.it)

### ❑ Materiale didattico

- ❑ Chi sa già che sosterrà il corso “Sistemi di Gestione di Basi dati”, può utilizzare il testo:  
Atzeni, Ceri, Paraboschi, Torlone, *“Basi di dati: linguaggi, sistemi e architetture”*, Seconda edizione, McGraw-Hill, 1999.

### ❑ Ulteriore materiale disponibile sulla [pagina web](#)

- esercitazioni
- documentazione sul DBMS adottato
- esercizi di esame (anni accademici precedenti)

## Programma del corso di Basi di dati

1. Introduzione alle basi di dati
  - il concetto di basi di dati
  - introduzione ai sistemi di gestione
2. Il modello relazionale
  - basi di dati relazionali
  - algebra relazionale
3. Sistemi di gestione di basi di dati
  - definizione di una base di dati
  - utilizzo di una base di dati
  - il linguaggio SQL
4. Introduzione alla progettazione di basi di dati
5. La progettazione concettuale
  - modello entità-relazione
  - metodologia di progettazione concettuale
6. La progettazione logica-fisica
  - metodologia di progettazione logica
  - cenni alla progettazione fisica

## 1. Introduzione alle basi di dati

### 1.1 Il concetto di basi di Dati

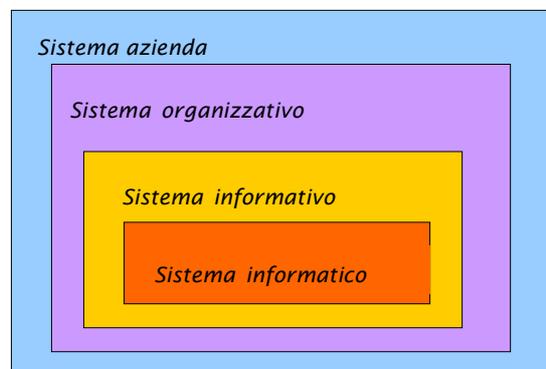
1. il concetto di basi di dati
2. introduzione ai sistemi di gestione

## Risorse di una organizzazione

- le risorse di una organizzazione:
  - persone
  - denaro
  - materiali
  - **dati e informazioni (sistema informativo)**
- funzioni di un sistema informativo
  - raccolta, acquisizione delle informazioni
  - archiviazione, conservazione delle informazioni
  - elaborazione delle informazioni
  - distribuzione, scambio di informazioni
  - il concetto di “sistema informativo” è indipendente da qualsiasi forma di automatizzazione

## Sistema Informativo

- porzione automatizzata del sistema informativo
- il sistema informatico è la parte del sistema informativo che gestisce informazioni per mezzo della tecnologia informatica



## Base di dati

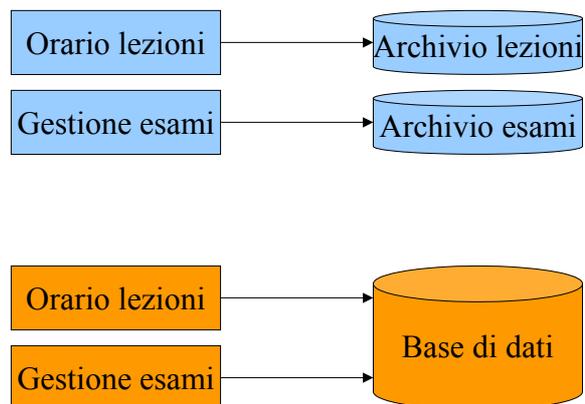
(accezione generica)

- **collezione di dati, utilizzati per rappresentare le informazioni di interesse per una o più applicazioni di una organizzazione.**

(accezione specifica)

- **collezione di dati in memoria secondaria gestita da un apposito sistema software, chiamato DBMS** (Sistema di Gestione di Basi di Dati).

## Nuova architettura del sistema informatico



# 1. Introduzione alle Basi di Dati

## 1.2 Introduzione ai sistemi di gestione

1. il concetto di basi di dati
2. introduzione ai sistemi di gestione

## Sistema di gestione di basi di dati Data Base Management System — DBMS

Sistema (**prodotto software**) in grado di gestire **collezioni di dati** che siano (anche):

- **grandi** (di dimensioni molto maggiori della memoria centrale dei sistemi di calcolo utilizzati)
- **persistenti** (con un periodo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano)
- **condivise** (utilizzate da applicazioni diverse)

garantendo:

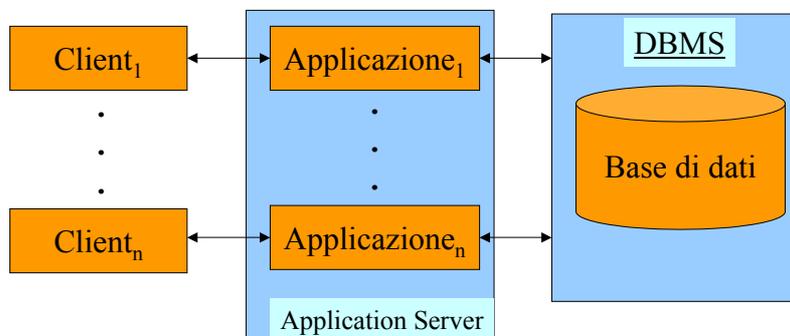
- **affidabilità** (resistenza a malfunzionamenti hardware e software)
- **privatizza** (con una disciplina e un controllo degli accessi),
- **efficienza** (utilizzare al meglio le risorse di spazio e tempo del sistema)
- **efficacia** (rendere produttive le attività dei suoi utilizzatori).

## Un po' di storia

- **Inizio anni '60:** Charles Bachman (General Electric) progetta il primo DBMS (Integrated Data Store), basato sul modello reticolare. Bachman vincerà il primo *ACM Turing Award* nel 1973.
- **Fine anni '60:** l'IBM sviluppa l'Information Management System (IMS), basato sul modello gerarchico e usato tutt'oggi.
- **1970:** Edgar Codd (IBM) propone il modello relazionale. Codd vincerà l'*ACM Turing Award* nel 1981.
- **Anni '80:** il modello relazionale vince sugli altri, e i DBMS basati su tale modello si diffondono. Il linguaggio SQL viene standardizzato come linguaggio per DBMS basati sul modello relazionale.
- **Anni '90:** sulla spinta di intense ricerche, i DBMS relazionali divengono sempre più sofisticati e diffusi (DB2, Oracle, Informix, ecc.). Nel 1999 James Gray vince l'*ACM Turing Award* per il suo contributo alla gestione delle transazioni.
- **Recentemente:** i DBMS si integrano con strumenti WEB, e ampliano il loro spettro di utilizzazione.

## Base di dati gestita dal DBMS

Il DBMS è l'unico responsabile della gestione della base di dati: i dati sono accessibili all'esterno solo tramite il DBMS



## Condivisione

- A ciascun settore o attività del sistema organizzativo corrisponde un (sotto-)sistema informativo, ed è comune che esistano sovrapposizioni fra i dati di interesse dei vari settori.
- Una base di dati è una risorsa **integrata**, condivisa fra i vari settori. L'integrazione e la condivisione permettono di ridurre la **ridondanza** (evitando ripetizioni) e, di conseguenza, le possibilità di incoerenza (o **inconsistenza**) fra i dati.
- Poiché la condivisione non è mai completa (o comunque non opportuna) i DBMS prevedono meccanismi di definizione della **privatezza** dei dati e di limitazioni all'accesso (**autorizzazioni**).
- La condivisione richiede un opportuno coordinamento degli accessi: **controllo della concorrenza**.

## Efficienza

- Si misura (come in tutti i sistemi informatici) in termini di **tempo di esecuzione** (tempo di elaborazione) e **spazio di memoria** (principale e secondaria).
- L'efficienza è il risultato della **qualità** del DBMS e delle applicazioni che lo utilizzano.
- I DBMS, a causa della varietà di funzioni, non sono necessariamente sempre più efficienti dei file system.

## DBMS vs file system

- La gestione di insiemi di dati grandi e persistenti è possibile anche attraverso sistemi più semplici — gli ordinari **file system** dei sistemi operativi, che permettono di realizzare anche rudimentali forme di condivisione.
- I DBMS estendono le funzionalità dei file system, fornendo più servizi ed in maniera integrata (cfr. **efficacia**).

## DBMS vs file system (coerenza dei dati)

I **file system** non prevedono verifiche sulla coerenza dei dati:

- diversi file possono essere creati, cancellati e modificati indipendentemente l'uno dall'altro
- questo può creare incoerenze rispetto alle informazioni che il contenuto dei file rappresentano

Nei **DBMS**, sono presenti meccanismi che permettono di garantire che venga mantenuta la **coerenza reciproca dei dati**.

## DBMS vs file system (condivisione dei dati)

I **file system** prevedono forme di condivisione:

- accesso condiviso in lettura
- accesso esclusivo in scrittura: se è in corso un'operazione di scrittura su un file, altri non possono accedere affatto al file

Nei **DBMS**, c'è maggiore **flessibilità** e **granularità**:

- si può accedere contemporaneamente a record diversi di uno stesso file o addirittura allo stesso record (in lettura)

## DBMS vs file system (catalogo dei dati)

Nei programmi tradizionali che accedono a file:

- ogni programma contiene una descrizione della struttura del file stesso
- nascono rischi di incoerenza fra le descrizioni (ripetute in ciascun programma) e i file stessi

Nei **DBMS**, esiste una porzione della base di dati (il **catalogo** o **dizionario**) che contiene una **descrizione centralizzata dei dati**, che può essere utilizzata dai vari programmi.

## DBMS vs file system (indipendenza dei dati)

- I DBMS consentono di utilizzare diversi livelli di descrizione e rappresentazione dei dati.
- I programmi fanno riferimento all'organizzazione logica della base di dati (livello alto).
- L'organizzazione fisica dei dati (livello basso) può essere modificata senza necessità di modifica dei programmi.
- In questo modo il DBMS garantisce l'**indipendenza dei dati** dalla rappresentazione fisica.

## DBMS vs file system (interrogazione dei dati)

Nei programmi tradizionali che accedono a file ogni programma deve realizzare:

- i meccanismi di accesso ai dati nei file
- i metodi di composizione di tali dati per ottenere i risultati richiesti

I **DBMS** mettono a disposizione un **linguaggio di interrogazione**, che permette di accedere in modo semplice e flessibile ai dati memorizzati e di comporre i dati elementari in modo opportuno.

## Modello dei dati

- insieme di costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica
- componente fondamentale: **meccanismi di strutturazione** (o **costruttori di tipo**)
- come nei linguaggi di programmazione esistono meccanismi che permettono di definire nuovi tipi, così ogni modello dei dati prevede alcuni costruttori
- ad esempio, il **modello relazionale** prevede il costruttore **relazione**, che permette di definire insiemi di record omogenei

## Tabelle: rappresentazione di relazioni

CORSI	Corso	Docente	Aula
	Basi di dati	Rossi	DS3
	Sistemi	Neri	N3
	Reti	Bruni	N3
	Controlli	Bruni	G

AULE	Nome	Edificio	Piano
	DS1	Ex-OMI	Terra
	N3	Ex-OMI	Terra
	G	Pincherle	Primo

## Schemi e istanze

In ogni base di dati si distinguono:

- lo **schema**, sostanzialmente invariante nel tempo, che ne descrive la struttura (aspetto **intensionale**); nell'esempio, le intestazioni delle tabelle

*Esempio:* CORSI(Corso, Docente, Aula)  
AULE(Nome, Edificio, Piano)

- l'**istanza**, costituita dai valori attuali, che possono cambiare molto e rapidamente (aspetto **estensionale**); nell'esempio, il "corpo" di ciascuna tabella

*Esempio:*

Basi di Dati	Rossi	DS3
Sistemi	Neri	N3
Reti	Bruni	N3
Controlli	Bruni	G

## Due tipi (principali) di modelli

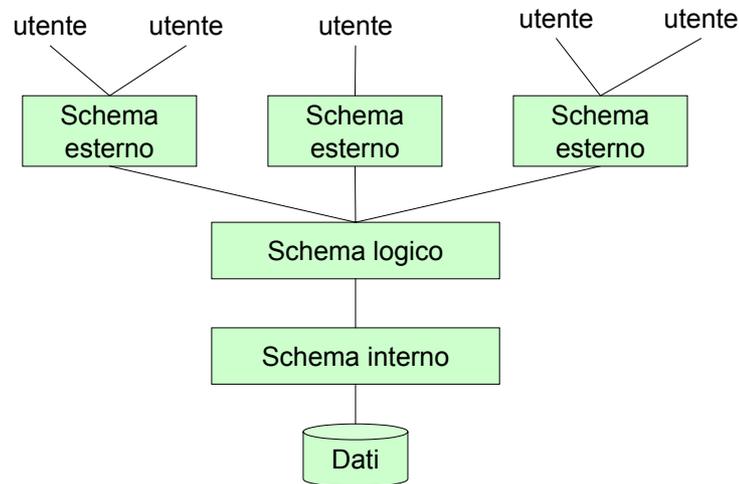
**modelli logici:** utilizzati nei DBMS esistenti per l'organizzazione dei dati; ad essi fanno riferimento i programmi; sono indipendenti dalle strutture fisiche;

*esempi:* **relazionale**, reticolare, gerarchico, a oggetti

**modelli concettuali:** permettono di rappresentare i dati in modo indipendente da ogni sistema, cercando di descrivere i concetti del mondo reale; sono utilizzati nelle fasi preliminari di progettazione;

*esempi:* il più noto è il modello **Entità-Relazione**

## Architettura standard (ANSI/SPARC) a tre livelli per DBMS



Maurizio Lenzerini

Basi di Dati

Introduzione - 29

## Architettura ANSI/SPARC: schemi

**schema logico:** descrizione dell'intera base di dati nel modello logico adottato dal DBMS

**schema esterno:** descrizione di una porzione della base di dati di interesse in un modello logico ("viste" parziali, derivate, anche in modelli diversi)

**schema interno (o fisico):** rappresentazione dello schema logico per mezzo di strutture fisiche di memorizzazione

Maurizio Lenzerini

Basi di Dati

Introduzione - 30

## Una vista

### Corsi

Corso	Docente	Aula
Basi di dati	Rossi	DS3
Sistemi	Neri	N3
Reti	Bruni	N3
Controlli	Bruni	G

### Aule

Nome	Edificio	Piano
DS1	Ex-OMI	Terra
N3	Ex-OMI	Terra
G	Pincherle	Primo

### CorsiSedi

Corso	Aula	Edificio	Piano
Sistemi	N3	Ex-OMI	Terra
Reti	N3	Ex-OMI	Terra
Controlli	G	Pincherle	Primo

## Indipendenza dei dati

Conseguenza della articolazione in livelli: l'accesso avviene solo tramite il livello esterno (che può coincidere con il livello logico)

Due forme di indipendenza dei dati:

**fisica:** il livello logico e quello esterno sono indipendenti da quello fisico; una relazione è utilizzata nello stesso modo qualunque sia la sua realizzazione fisica (che può anche cambiare nel tempo senza che debbano essere modificate le forme di utilizzo)

**logica:** il livello esterno è indipendente da quello logico

- aggiunte o modifiche alle viste non richiedono modifiche al livello logico
- modifiche allo schema logico che lascino inalterato lo schema esterno sono trasparenti

## Linguaggi per basi di dati

Un altro contributo all'efficacia è la disponibilità di vari linguaggi e di interfacce diverse.

L'accesso ai dati può avvenire:

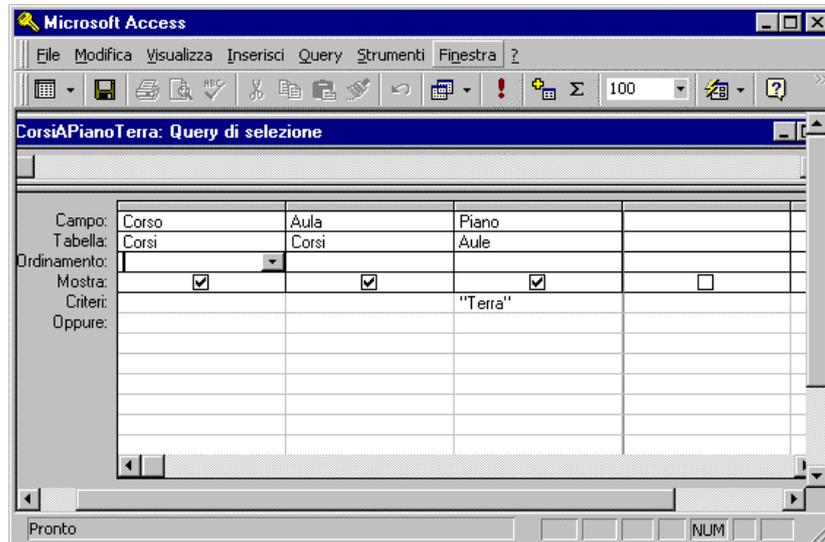
1. con **linguaggi testuali interattivi** (ad es. SQL)
2. con comandi (come quelli del linguaggio interattivo) immersi in un **linguaggio ospite** (Java, C, Cobol, etc.)
3. con comandi (come quelli del linguaggio interattivo) immersi in un **linguaggio ad hoc**, con anche altre funzionalità (ad es. per grafici o stampe strutturate), anche con l'ausilio di strumenti di sviluppo (ad es. per la gestione di maschere)
4. con **interfacce amichevoli** (senza linguaggio testuale)

## SQL, un linguaggio interattivo

```
SELECT Corso, Aula, Piano
FROM Aule, Corsi
WHERE Aule.Nome = Corsi.Aula
      AND
      Aule.Piano = "Terra"
```

Corso	Aula	Piano
Reti	N3	Terra
Sistemi	N3	Terra

## Interazione non testuale (MS Access)



Maurizio Lenzerini

Basi di Dati

Introduzione - 35

## Una distinzione terminologica (separazione fra dati e programmi)

### Data Definition Language (DDL):

per la definizione di schemi (logici, esterni, fisici) e altre operazioni generali

### Data Manipulation Language (DML):

per l'interrogazione e l'aggiornamento di (istanze di) basi di dati

Maurizio Lenzerini

Basi di Dati

Introduzione - 36

## Personaggi e interpreti

- progettisti e realizzatori di DBMS
- progettisti della base di dati e amministratori della base di dati (DBA)
- progettisti e programmatori di applicazioni
- utenti
  - utenti finali (terminalisti): eseguono applicazioni predefinite (**transazioni**)
  - utenti casuali: eseguono operazioni non previste a priori, usando linguaggi interattivi

## Database administrator (DBA)

- Persona o gruppo di persone responsabile del controllo centralizzato e della gestione del sistema, delle prestazioni, dell'affidabilità, delle autorizzazioni.
- Le funzioni del DBA includono anche (parte di) quelle di progettazione.

## Transazioni

- programmi che realizzano attività frequenti e predefinite sui dati, con poche eccezioni
- *Esempi:*
  - versamento presso uno sportello bancario
  - emissione di certificato anagrafico
  - dichiarazione presso l'ufficio di stato civile
  - prenotazione aerea
- Le transazioni sono di solito realizzate con programmi in linguaggio ospite (tradizionale o ad hoc).
- **N. B.:** il termine **transazione** ha un'altra accezione, più tecnica: sequenza indivisibile di operazioni (o vengono eseguite tutte o nessuna).

## Vantaggi e svantaggi dei DBMS

### Pro

- dati come risorsa comune, schema dei dati come modello della realtà
- gestione centralizzata con possibilità di standardizzazione ed "economia di scala"
- disponibilità di servizi integrati
- riduzione di ridondanze e incoerenze
- indipendenza dei dati (favorisce lo sviluppo e la manutenzione delle applicazioni)

### Contro

- costo dei prodotti e della transizione verso di essi
- non scorponabilità delle funzionalità (con riduzione di efficienza)