

Basi di dati

Giuseppe De Giacomo

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

Anno Accademico 2005/2006
Canale M-Z

<http://www.dis.uniroma1.it/~degiacomo/didattica/basidati/>

Il corso di Basi di Dati è rivolto a

Nuovo ordinamento (5 crediti, terzo anno)

- Laurea in Ingegneria Informatica
 - canale M-Z: Prof. De Giacomo
 - canale A-L: Prof. Lenzerini
- Collegato a questo modulo è il corso “Progetto di basi di dati”, che viene erogato nel secondo periodo didattico

Vecchio ordinamento (Prof. Lenzerini)

- Il presente corso è uno di due moduli per la Laurea in Ingegneria Informatica (e altri corsi di laurea)
- 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, come unico esame che riguarda entrambi i moduli

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 dei corsi della laurea specialistica “Sistemi di gestione di basi di dati” e del corso “Gestione dei dati nei sistemi informativi”:

- *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: Giuseppe De Giacomo

<http://www.dis.uniroma1.it/~degiacomo>

Ricevimento:

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

Sito del corso del Prof. De Giacomo

<http://www.dis.uniroma1.it/~degiacomo/didattica/basidati/>

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):

- Martedì, ore 17:30 - 19:00
- Mercoledì, ore 17:30 - 19:00
- Venerdì, ore 14:00 - 15:30

Esercitazioni in aula

Esercitazioni al calcolatore

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

Esame composto da

- prova scritto
- prova orale

Aspetti organizzativi del corso

Materiale didattico e tutore

- Lucidi delle lezioni (nella pagina web con qualche giorno di anticipo)
- R. Ramakrishna, J Gehrke, “*Sistemi di basi di dati*”, McGraw-Hill, 2004

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

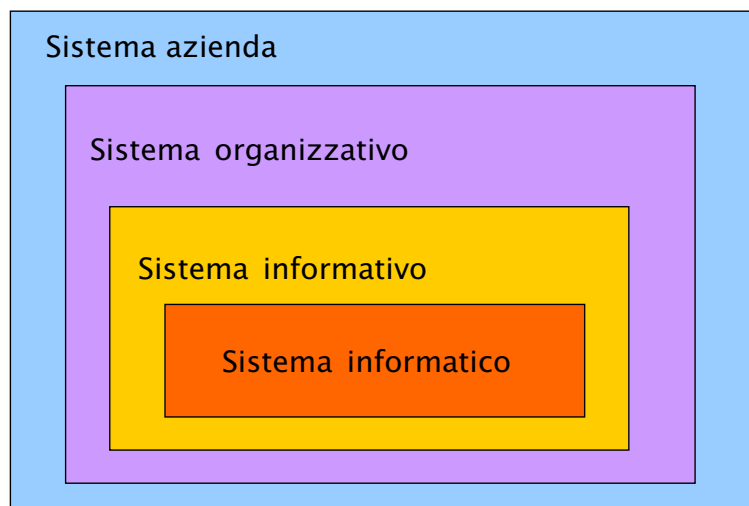
1. **il concetto di basi di dati**
2. introduzione ai sistemi di gestione

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 informatico

- 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

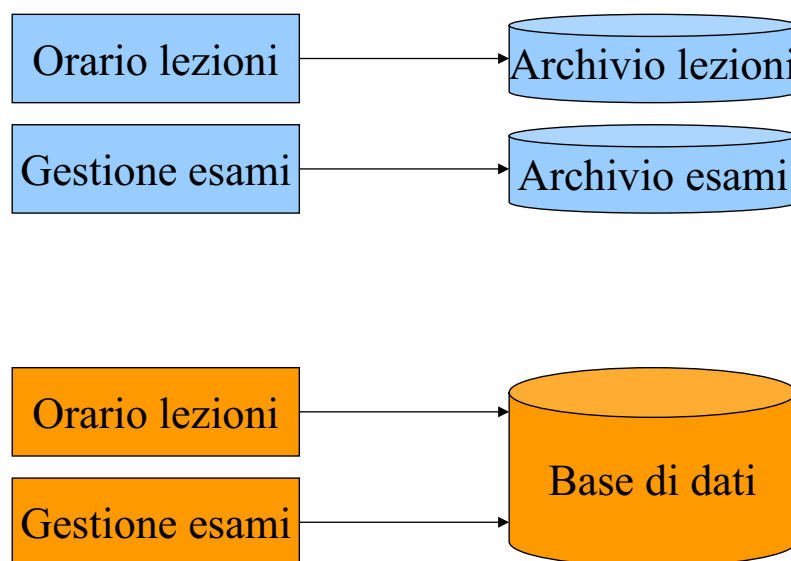
(accezione generica)

- **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 (Data Base Management Systems, o 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 normalmente)
- **persistenti** (con un periodo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano)
- **condivise** (utilizzate da applicazioni diverse)

garantendo:

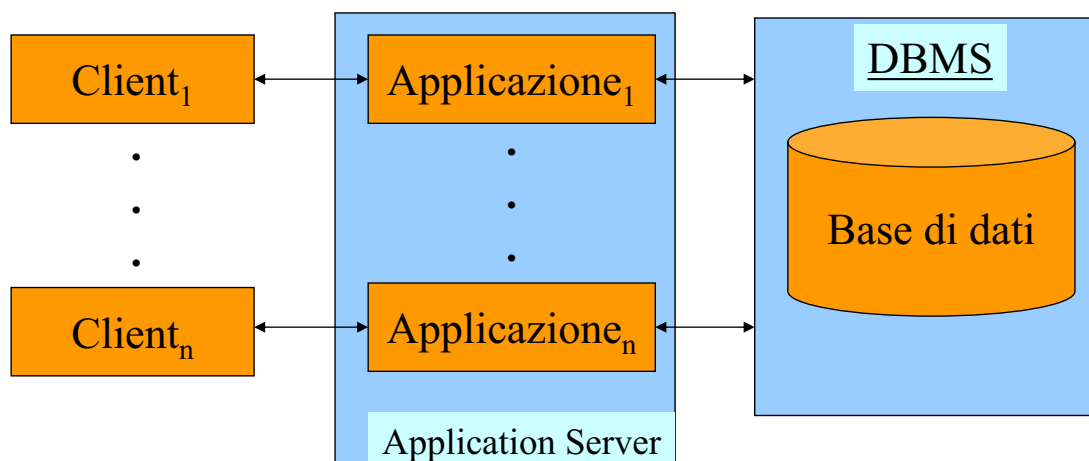
- **affidabilità** (resistenza a malfunzionamenti hardware e software)
- **privatezza** (con una disciplina e un controllo degli accessi),
- **efficienza** (utilizzare al meglio le risorse di spazio e tempo del sistema)
- **efficacia** (rendere produttive le attività dei suoi utilizzatori).

Un pò 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 il contesto generale dello sviluppo del software e 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



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;

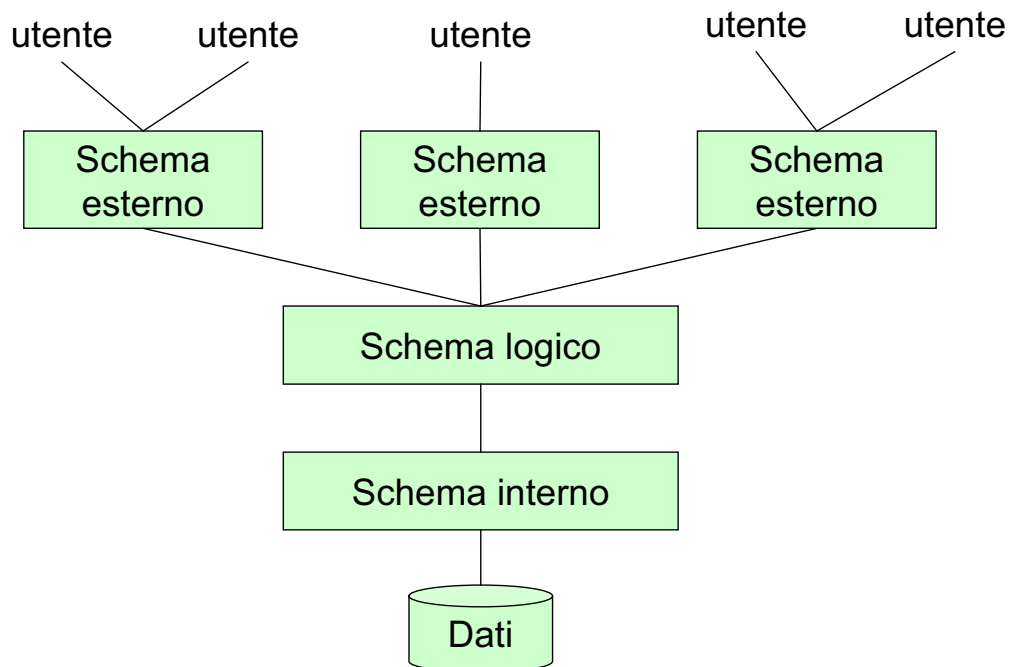
esempi: **relazionale**, reticolare, gerarchico, a oggetti

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

*ANSI Standards Planning And Requirements Committee



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

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:

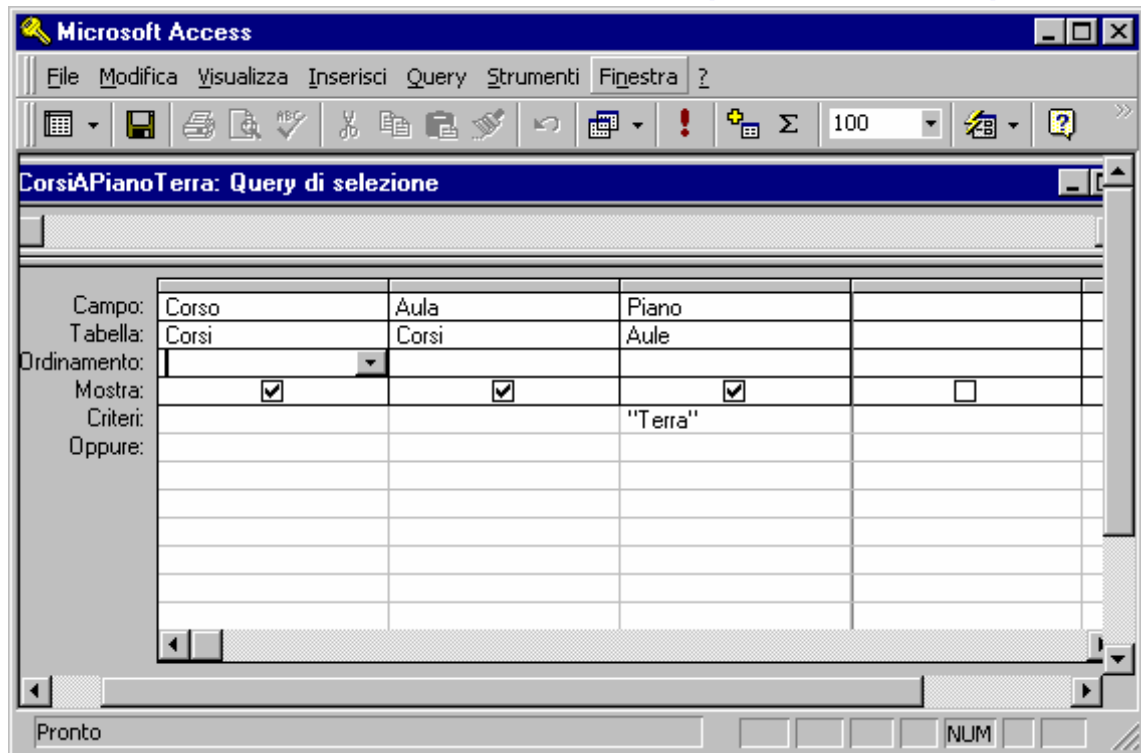
1. con **linguaggi testuali interattivi** (ad es. SQL)
2. con comandi (come quelli del linguaggio interattivo) immersi in un **linguaggio ospite** (Java, C, Cobol, etc.)
3. con comandi (come quelli del linguaggio interattivo) immersi in un **linguaggio ad hoc**, con anche altre funzionalità (ad es. per 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)



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

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

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 scorporabilità delle funzionalità (con potenziale riduzione di efficienza)