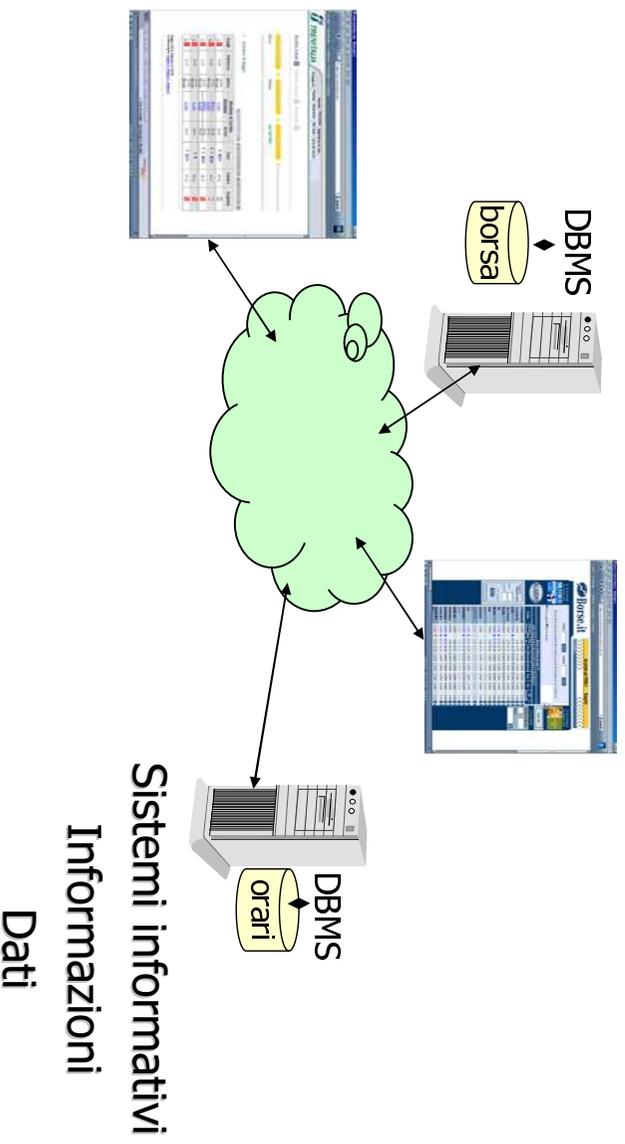


# Introduzione



1

# Sistema informativo

- Sottosistema di una organizzazione che gestisce
  - acquisizione
  - elaborazione
  - conservazione
  - produzionedelle **informazioni** di interesse
- Ogni organizzazione ha un sistema informativo
  - Le informazioni sono un "bene"
- L'esistenza del sistema informativo è indipendente dalla sua automazione (**sistema informatico**)

2

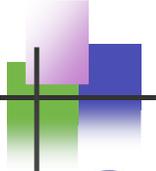


# Sistema informatico

- Gestisce un sistema informativo in modo automatizzato
- Garantisce che i dati siano conservati in modo permanente sui dispositivi di memorizzazione
- Permette un rapido aggiornamento dei dati per riflettere rapidamente le loro variazioni
- Rende i dati accessibili alle interrogazioni degli utenti
- Può essere distribuito sul territorio
- Anche prima di essere automatizzati, molti sistemi informativi si sono evoluti verso una

**razionalizzazione e standardizzazione delle procedure  
e dell'organizzazione delle informazioni**

3



# Gestione delle informazioni

- Modo di gestire e comunicare le informazioni
  - **accesso, elaborazione, trasmissione**
  - lingua scritta e parlata
  - disegni, grafici, schemi
  - codici e numeri
- Modo di memorizzare l'informazione
  - ricordata a memoria
  - copia cartacea
  - in formato elettronico

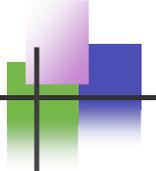
4



# Sistemi per la gestione di informazioni

- Nelle attività standardizzate dei sistemi informativi complessi, sono state introdotte col tempo  
**forme di organizzazione e codifica delle informazioni**
- Ad esempio, nei **servizi anagrafici** si è iniziato con registrazioni discorsive e poi
  - nome e cognome
  - estremi anagrafici
  - codice fiscale .... una **chiave** per tutti!

5



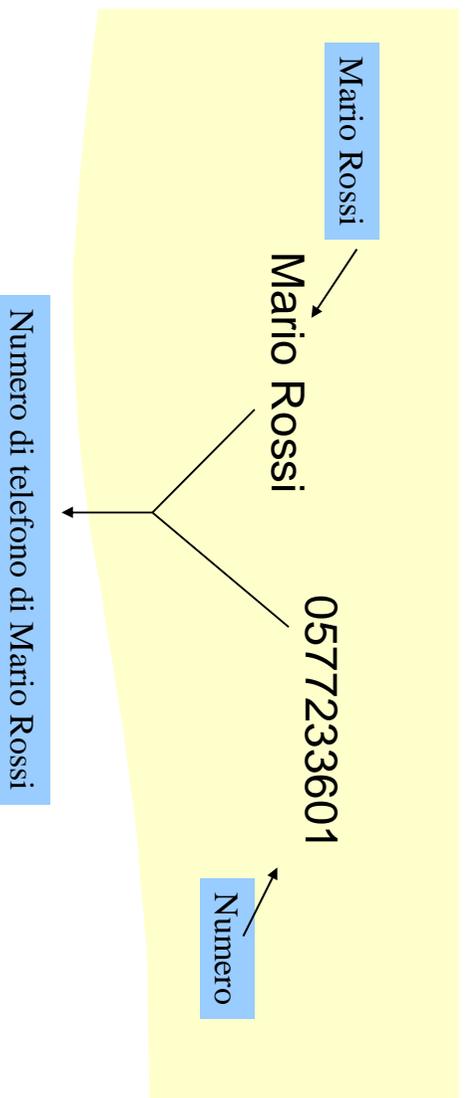
# Informazioni e dati

- Le informazioni vengono rappresentate e codificate in modo essenziale attraverso **dati**
- I dati devono essere interpretati per recuperare l'informazione
  - **informazione.** Notizia, dato o elemento che consente di avere conoscenza più o meno esatta di fatti, situazioni, modi di essere.
  - **dato.** Ciò che è immediatamente presente alla conoscenza, prima di ogni elaborazione; (in informatica) elementi di informazione costituiti da simboli che debbono essere elaborati.

6

# Dai dati all'informazione

- L'informazione è ottenuta interpretando e correlando fra loro i dati



7

# Un esempio: la rubrica telefonica

Dati

Nome Cognome Indirizzo Telefono

Mario Rossi Via Garibaldi 10 05773576373

Operazioni

- Cercare un numero dato il nome
- Inserire un nuovo numero
- Modificare un indirizzo
- .....

8



# Base di dati

- E' una **collezione di dati** utilizzata per rappresentare le informazioni di interesse in un sistema informativo
- L'elemento chiave è il dato
  - La rappresentazione precisa di forme più ricche di informazione e conoscenza è difficile
  - I dati costituiscono spesso una risorsa strategica, perché più **stabili nel tempo** di altre componenti (processi, tecnologie, ruoli umani)
  - I dati rimangono gli stessi nella **migrazione** da un sistema al successivo

9



# Basi di dati: quali applicazioni?

- Anagrafe - anagrafe studenti
- Servizi bancari - conti correnti - bancomat - carte di credito
- Elenchi delle utenze telefoniche
- Orari ferroviari
- Prenotazioni voli
- Catalogo prodotti (CD,...) - E-commerce
- Cataloghi di biblioteche
- Cartelle cliniche
- .....

10



## Basi di dati: come si accede?

- in modo distribuito (Sportelli bancomat, POS negozi...)
- da Internet (Orari FFSS, siti di e-commerce, cddb, .....
- da terminali dedicati (Agenzie di viaggio, biblioteche, anagrafe...)
- sul proprio personal (database personali)
- .....

11



## Gestione di basi di dati

- Utilizzare un linguaggio di programmazione generico (C, C++, COBOL, Java) memorizzando i dati su **file** su disco
  - Applicazioni specifiche
  - Duplicazione dei dati
  - Incoerenza dei dati condivisi
- Utilizzare un sistema di gestione di basi di dati (**Database Management System - DBMS**)
  - Sistema software progettato per gestire dati
  - Capacità di gestire collezioni di dati **grandi, condivise, persistenti**

12



# Caratteristiche dei DBMS

- **grandi dimensioni**
  - Possono avere dimensioni di Terabyte
  - Fanno uso della memoria secondaria
- **condivisione**
  - accesso di più utenti/applicazioni a dati comuni
  - si riduce la ridondanza dei dati e la possibilità di inconsistenze
  - Il DBMS deve disporre del **controllo di concorrenza**
- **persistenza**
  - i dati hanno una vita che va oltre quella dell'esecuzione delle applicazioni che li utilizzano

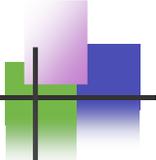
13



# Requisiti dei DBMS

- **affidabilità**
  - capacità di non perdere i dati in caso di malfunzionamento hardware o software (**backup** e **recovery**)
- **privatizza**
  - ciascun utente è riconosciuto da un *username* e una *password*
  - un utente ha autorizzazione solo per certe operazioni
- **efficienza**
  - capacità di svolgere le operazioni in tempo ragionevole e con risorse di calcolo e memoria accettabili
  - dipende dall'implementazione
  - può imporre dei vincoli sull'hardware

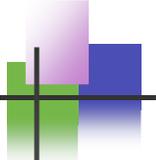
14



## DBMS vs filesystem [1]

- La gestione di insiemi di dati grandi e persistenti è possibile anche attraverso sistemi più semplici  
il **file system** del sistema operativo
- I file system prevedono forme di condivisione *tutto o niente*. Nei DBMS c'è maggiore flessibilità
- I DBMS estendono le funzionalità del file system, fornendo più servizi ed in maniera integrata
- Comunque, i DBMS, a causa della varietà di funzioni, non sono necessariamente più efficienti dei file system

15



## DBMS vs filesystem [2]

- Nei programmi tradizionali che accedono a file, ogni programma contiene una **descrizione della struttura del file** stesso, con i conseguenti 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.

16



# Descrizione dei dati nei DBMS

- Livelli diversi di descrizione e rappresentazione dei dati
  - permettono l'**indipendenza dei dati** dalla rappresentazione fisica
  - i programmi fanno riferimento alla struttura a livello più alto
  - le rappresentazioni sottostanti possono essere modificate senza necessità di modifica dei programmi
- Si introduce il concetto di **modello dei dati**
  - modalità usata per organizzare i dati e descriverne la struttura
  - offre meccanismi di strutturazione simili ai costruttori di tipo dei linguaggi di programmazione

17



# I modelli dei dati

- **modelli logici** - utilizzati nei DBMS esistenti per l'organizzazione dei dati
  - utilizzati dai programmi
  - indipendenti dalle strutture fisiche
- **modelli concettuali** - permettono di rappresentare i dati in modo indipendente da ogni sistema
  - cercano di descrivere i concetti del mondo reale
  - sono utilizzati nelle fasi preliminari di progettazione

## Modello Entity-Relationship (E-R)

18

# I modelli logici dei dati

- Esistono vari modelli (logici) di dati proposti
  - **modello gerarchico ('60)**  
Basato sull'uso di strutture ad albero (gerarchie).
  - **modello reticolare - CODASYL ('70)**  
Basato sull'uso di grafi.
  - **modello relazionale ('70-'80)**  
Basato sull'uso del concetto di relazione.
  - **modello ad oggetti ('80)**  
Evoluzione del modello relazionale. Basato sulle idee della programmazione ad oggetti.

19

# Il modello relazionale

- Permette di organizzare in insiemi di **record a struttura fissa** (concetto di relazione)
- Una relazione è spesso rappresentata per mezzo di una **tabella**  
campo

<b>Nome</b>	<b>Cognome</b>	<b>Indirizzo</b>	<b>Telefono</b>
Mario	Rossi	Via Verdi 5	0577234567
Marco	Bianchi	Via Righi 73	0577456783
Anna	Dati	Via Romeo 4	0578345234

record →

relazione

20

# Schemi ed istanze

- **Schema** - Parte sostanzialmente invariabile dei dati  
(componente **intensionale**)

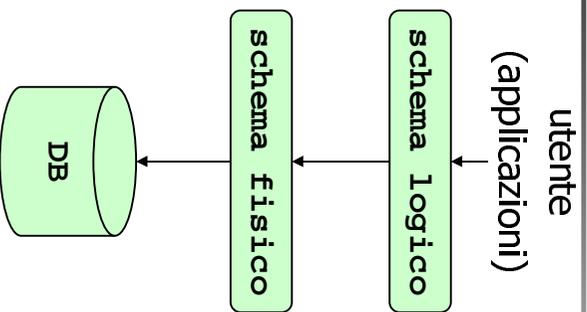
RUBRICA(Nome,Cognome,Indirizzo,Telefono)

⌘ **Istanza** - valori effettivi contenuti nella base di dati (stato)  
(componente **estensionale**)

Mario	Rossi	Via Verdi 5	0577234567
Marco	Bianchi	Via Righi 73	0577456783
Anna	Dati	Via Romeo	0578345234

21

# Schemi e astrazione



descrizione della base di dati  
nel modello logico  
(es, la struttura della tabella)

—————>  
rappresentazione dello schema logico  
per mezzo di strutture memorizzate  
(file)

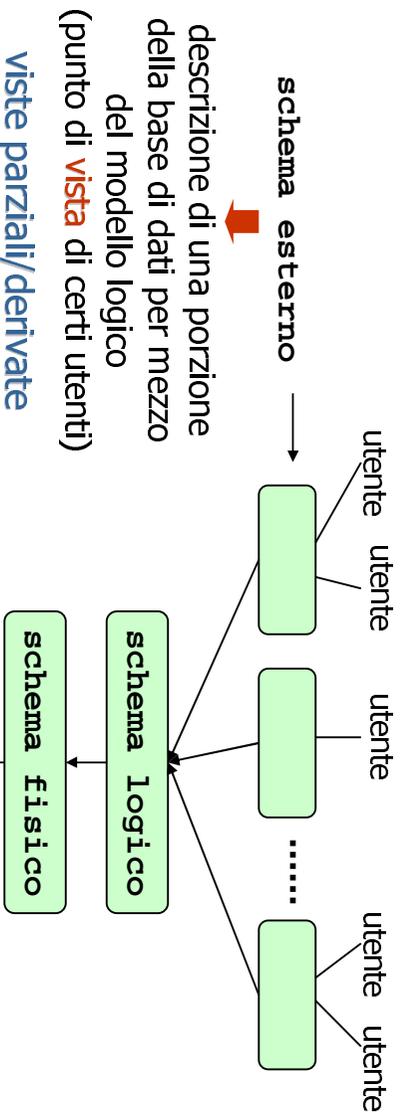


il livello logico è indipendente da quello fisico  
una tabella è utilizzata nello stesso modo  
qualunque sia la sua realizzazione fisica  
(che può anche cambiare nel tempo)

22

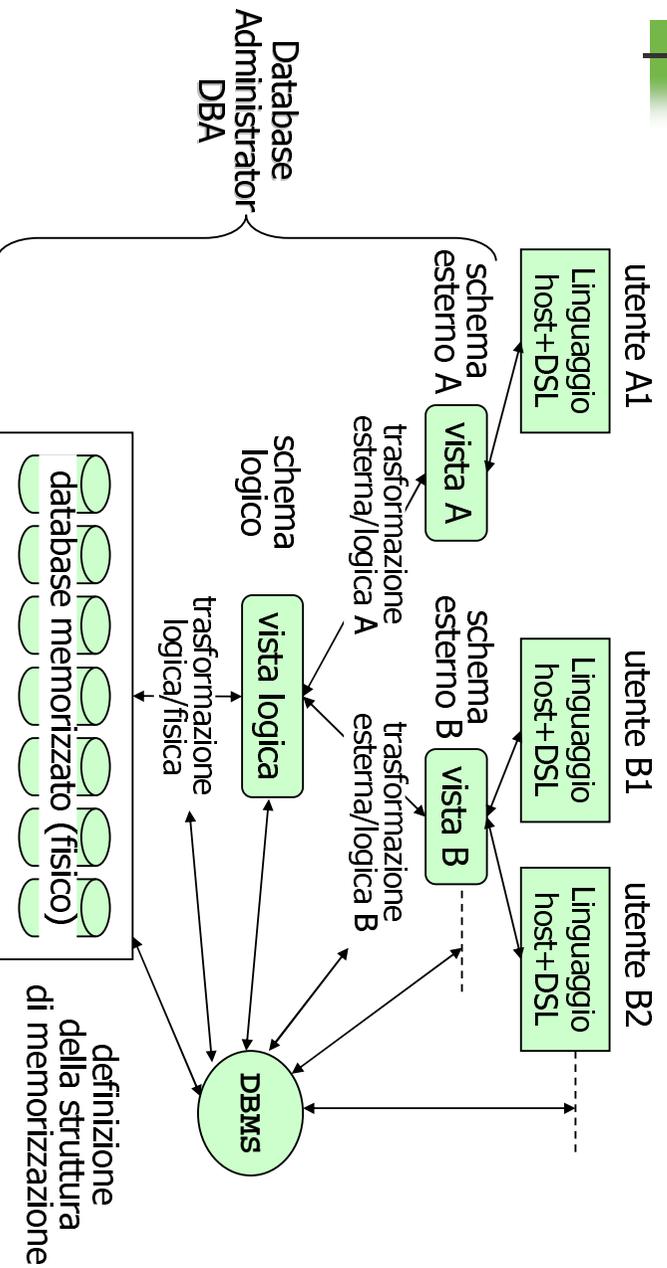
# Architettura ANSI/SPARC

Architettura a **3 livelli** proposta dall'ANSI/SPARC Study Group on DBMS



23

# Architettura dettagliata



24

# Le viste esterne

Vista A  
(segreteria Ingegneria)

Matricola	Nome

selezione dei record con  
Corso = Ingegneria  
e uso solo dei campi  
Matricola e Nome

Vista B  
(amministrazione)

Corso	Entrate

collezione dei record  
per valore del campo Corso  
e calcolo di Entrate come somma  
del campo Tasse in ogni gruppo

Schema logico

Matricola	Corso	Anno	Nome	Tasse

25

# Il DataBase Administrator

- E' la persona (o gruppo di persone) responsabile di
  - progettazione (logica/fisica)
  - definizione delle politiche di sicurezza (autorizzazioni)
  - amministrazione (backup, monitoraggio delle prestazioni) della base di dati
- Media le esigenze degli utenti mantenendo un controllo centralizzato sui dati
- Configura il DBMS per garantire sufficienti prestazioni
- Assicura l'affidabilità del sistema
- Gestisce le autorizzazioni di accesso.

26



# Utenti di un DBMS

## ■ Progettisti e programmatori di applicazioni

- realizzazione di programmi che accedono alla base di dati
- uso di linguaggi convenzionali (C, Java, PHP, ..) o linguaggi proprietari specifici del sistema con funzionalità specifiche (generazione di grafici, stampe, maschere sul video)
- uso di strumenti di sviluppo per la creazione di interfacce verso la base di dati

## ■ Utenti finali

- interazione basata su **transazioni** (attività frequenti e predefinite) utilizzando form o programmi guidati da menu (applicazioni sviluppate ad hoc dai programmatori)
- oppure uso di linguaggi interattivi e interrogazioni *casuali* (non predefinite)

27



# Transazioni

- Programmi che realizzano attività frequenti e predefinite, con poche eccezioni, previste a priori

## ■ Esempi:

- versamento presso uno sportello bancario
- emissione di certificato anagrafico
- prenotazione aerea

- Le transazioni sono di solito realizzate con programmi in linguaggio ospite (tradizionale o ad hoc)

- Il termine **transazione** ha un'altra accezione, più specifica: sequenza indivisibile di operazioni (o vengono eseguite tutte o nessuna)

28

# Linguaggi per basi di dati

- Linguaggi per la **definizione dei dati**  
**Data Definition Languages - DDL**  
definizione degli schemi logici, fisici e delle autorizzazioni di accesso
- Linguaggi di **manipolazione dei dati**  
**Data Manipulation Languages - DML**

interrogazione e aggiornamento delle basi di dati

Alcuni linguaggi come **SQL** (Structured Query Language) presentano integrate funzioni di entrambe le categorie

29

# SQL come DDL

Definizione di una tabella (relazione)

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

insegnamento	docente	aula	ora

orario

30

# SQL come DML

- Interrogazione

```
SELECT Insegnamento, Docente
```

```
FROM Orario
```

```
WHERE insegnamento = 'Basi di dati'
```

Produce una tabella

(relazione) che soddisfa

il criterio (clausola WHERE)



insegnamento	docente
Basi di dati	Maggini

31

# Linguaggi per basi di dati

- Disponibilità di vari linguaggi, interfacce diverse e supporto alla scrittura di programmi
  - linguaggi testuali interattivi (**Command Line Interface CLI**, ad esempio basate su SQL)
  - comandi immersi in un linguaggio **ospite** (es. **SQLJ** java embedded SQL)
  - disponibilità di librerie di interfaccia col DBMS per linguaggi di programmazione generici (es. **JDBC** - Java DataBase Connectivity)
  - comandi in un linguaggio ad hoc, anche con funzionalità specifiche (es. per grafici o stampe strutturate)
  - con interfacce amichevoli (senza linguaggio testuale)

32

# SQL Java embedded SQL

Supportato da Oracle, IBM, Sybase, Tandem, JavaSoft, Microsoft, Informix, XDB

- Implementazione <http://www.oracle.com/st/products/jdbc/sqlj>
- È un linguaggio semplice e coinciso per inglobare comandi SQL in programmi Java
- Lo standard garantisce la portabilità fra diversi DBMS
- Si inseriscono i comandi SQL nel programma
  - i comandi SQL iniziano con "#sql" e terminano con ";"
  - si può far riferimento alle variabili Java usando il prefisso "%"
  - il testo SQL è posto fra parentesi graffe "{..}"

33

# SQL vs JDBC

JDBC (**Java Database Connectivity**) mette a disposizione una libreria di classi per accedere al DBMS eseguendo comandi SQL

```
// SQL      variabile
           Java
int n;
#sql {INSERT INTO emp
      VALUES (:n)};
           variabile
           Java
           nell' sqlj
```

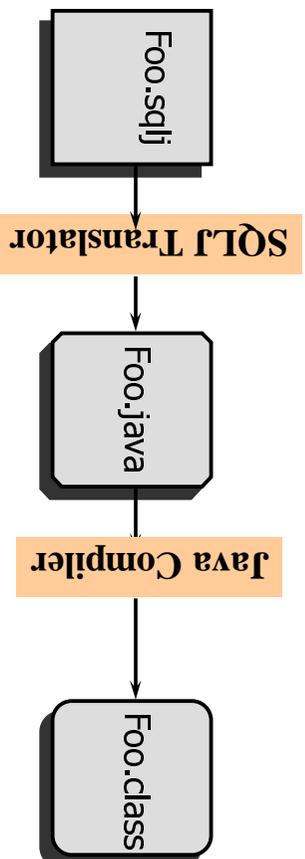
---

```
// JDBC
           classe della libreria
           java.sql
int n;
Statement stmt =
           conn.prepareStatement
           ("INSERT INTO emp VALUES (?)");
stmt.setInt(1,n);
stmt.execute ();
stmt.close();
```

34

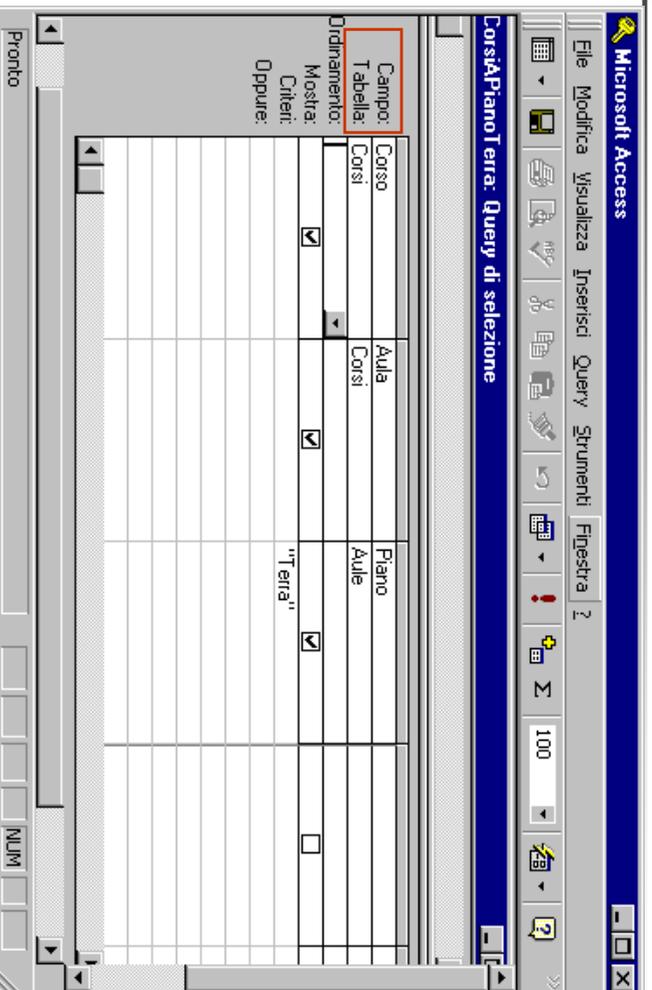
# Da SQL a Java

Il file sorgente SQL deve essere tradotto in un sorgente Java puro che può essere compilato in un file class



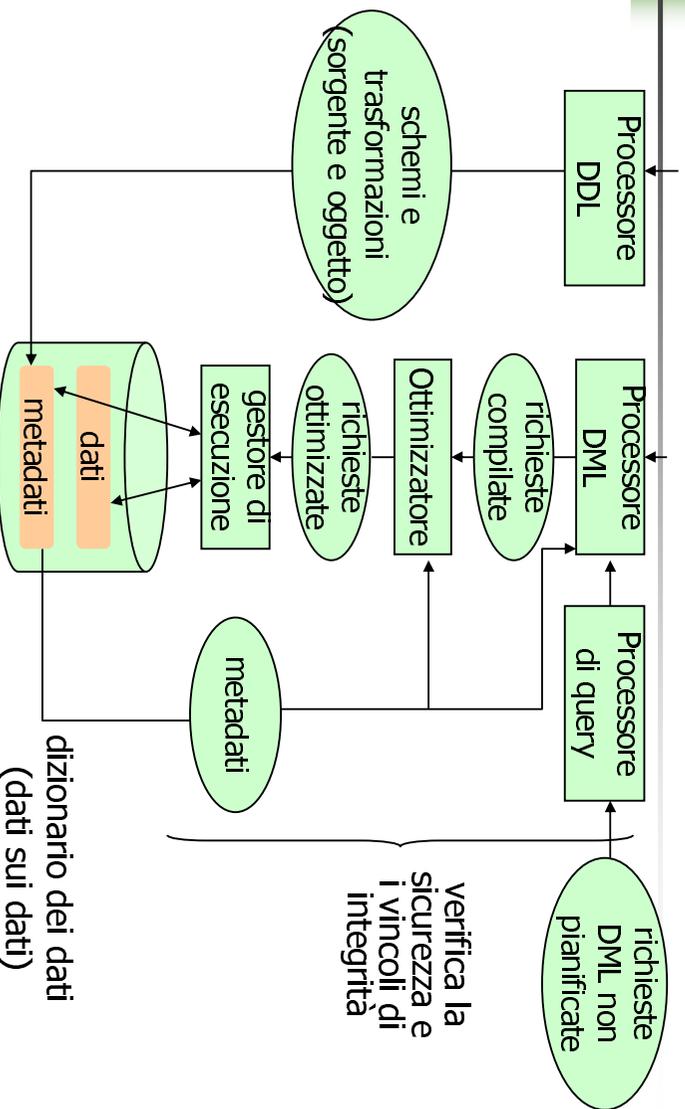
35

# Interazione grafica



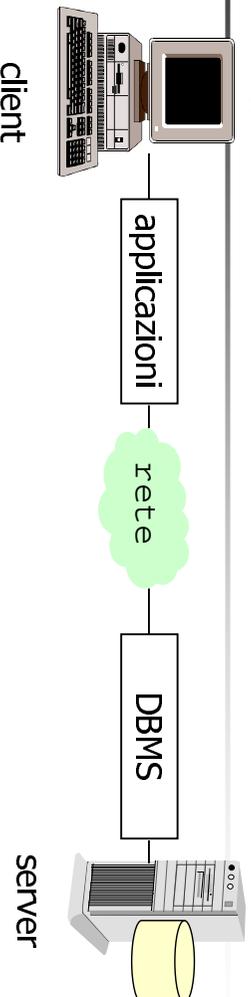
36

# Struttura di un DBMS



37

# Organizzazione distribuita



client

server

- L'applicazione database può essere organizzata col modello client/server
- Il client e il sever (DBMS) possono essere eseguiti su macchine diverse
- Uno stesso server può servire più client
- Un client può utilizzare più servers (database distribuito)

38



# Vantaggi e svantaggi dei DBMS

## Pro

- dati come risorsa comune a disposizione di tutta l'organizzazione
- gestione centralizzata con possibilità di standardizzazione
- disponibilità di servizi integrati
- riduzione di ridondanze e inconsistenze grazie alla condivisione
- indipendenza dei dati (favorisce lo sviluppo e la manutenzione delle applicazioni)

## Contro

- costo dei prodotti e della transizione verso di essi (strutture, hardware, personale)
- non scorponabilità di alcuni servizi non utili (con riduzione di efficienza)