

# Tecnologie di Sviluppo per il Web

## Programmazione su Basi di Dati: Introduzione

versione 3.0

Questo lavoro è concesso in uso secondo i termini di una licenza Creative Commons  
(vedi ultima pagina)

G. Mecca – mecca@unibas.it – Università della Basilicata



Programmazione su BD: Introduzione >> Sommario



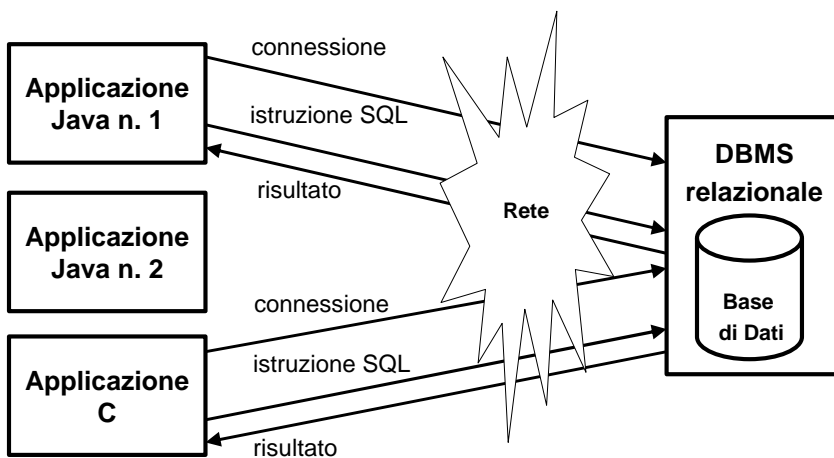
## Sommario

- Architettura
- Problemi Tecnologici
  - ⇒ Disaccoppiamento di Impedenza
- Progetto di Riferimento

# Architettura

- Obiettivo
  - ⇒ sviluppo di Sistemi Informativi
- Architettura
  - ⇒ client/server
  - ⇒ server: DBMS, base di dati condivisa
  - ⇒ client: applicazioni che realizzano le funzionalità del sistema informativo

# Architettura





## Architettura

- In sostanza
  - ⇒ cambia la tecnologia della persistenza, che è basata sul DBMS
- Sul client
  - ⇒ non cambiano molto interfaccia, controllo e modello
  - ⇒ cambia sostanzialmente lo strato di persistenza



## Problemi Tecnologici

- 1. Stabilire una connessione con il DBMS
  - ⇒ è necessario un opportuno protocollo
- 2. Specificare le istruzioni SQL nel codice
  - ⇒ regole per l'utilizzo di SQL nel programma
- 3. Gestire il risultato delle interrogazioni
  - ⇒ tipicamente una collezione di ennuple
- Il tutto
  - ⇒ effettuando le operazioni efficientemente



## Problemi Tecnologici

- Vantaggio

- ⇒ il DBMS offre in modo centralizzato servizi avanzati (concorrenza, ottimizzazione ecc.)

- Svantaggio

- ⇒ “disaccoppiamento di impedenza” tra DBMS linguaggio (in particolare a oggetti)

- ⇒ differenze nel modello logico dei dati

- ⇒ differenze nei linguaggi



## Disaccoppiamento di Impedenza

- Differenza nel modello logico dei dati

- Tipi di base

- ⇒ linguaggio: numeri, stringhe, booleani

- ⇒ DBMS: CHAR e VARCHAR, DATE, TIME...

- Costruttori di tipo

- ⇒ linguaggio a oggetti: classi, ereditarietà, polimorfismo

- ⇒ DBMS: tabelle ed ennuple



## Disaccoppiamento di Impedenza

- Differenza nel linguaggio
- Semantica delle operazioni
  - ⇒ linguaggio: operazioni su “variabili” singole
  - ⇒ DBMS: operazioni su collezioni di ennuple
- Tecniche algoritmiche
  - ⇒ linguaggio a oggetti: “navigazione” tra oggetti
  - ⇒ DBMS: assenza di puntatori, join su valori



## Progetto di Riferimento

- Il Sistema Informativo dell'ACI
  - ⇒ dati su automobili e proprietari
- I casi d'uso
  - ⇒ tipici casi d'uso di un'applicazione client-server
  - ⇒ corrispondono alle cosiddette operazioni CRUD sulle tabelle della base di dati



## Progetto di Riferimento

- Operazioni CRUD su una tabella
  - ⇒ “Create”: inserimento di una ennupla
  - ⇒ “Update”: aggiornamento di una ennupla
  - ⇒ “Delete”: eliminazione di una ennupla
  - ⇒ “Retrieve”: ricerca di ennuple nella tabella sulla base di un criterio di selezione
  - ⇒ le operazioni di tipo “retrieve” possono essere varie; es: ricerca per codice fiscale, ricerca per nome



## Progetto di Riferimento

- I casi d'uso dell'applicazione
  - ⇒ corrispondono alle operazioni CRUD su proprietari e automobili
- Casi d'uso relativi al proprietario
  - ⇒ ricerca proprietario per codice fiscale
  - ⇒ ricerca proprietario per nome
  - ⇒ inserimento proprietario
  - ⇒ aggiornamento proprietario
  - ⇒ eliminazione proprietario



## Progetto di Riferimento

- Casi d'uso relativi alle automobili
  - ⇒ ricerca automobile per targa
  - ⇒ inserimento nuova automobile
  - ⇒ aggiornamento dati automobile
  - ⇒ eliminazione automobile



## Progetto di Riferimento

- Funzionamento
  - ⇒ l'utente si identifica con nome utente e password (requisito tipico di sicurezza)
  - ⇒ non deve essere consentito ad utenti non autorizzati di effettuare operazioni sulla base di dati
  - ⇒ a seconda del suo ruolo (utente o amministratore) accede ad un menu che gli consente di effettuare varie operazioni



## Un Esempio

### ○ La base di dati

```
create table utenti (
  nomeUtente char(10)
    not null primary key,
  nome varchar(50),
  password varchar(10),
  ruolo varchar(50)
)
```

```
create table proprietari (
  codiceFiscale char(16)
    not null primary key,
  nome varchar(50) not null,
  cittaDiResidenza varchar(50),
  annoPatente integer
)
```

Supporteremo che la base di dati "automobili" sia già stata creata usando, per esempio, PostgreSQL

```
create table automobili (
  targa char(7) not null
    primary key,
  modello varchar(50),
  cilindrata integer,
  proprietario char(16)
    not null references
    Proprietari(codiceFiscale)
)
```



## Progetto di Riferimento

>> [it.unibas.aci](http://it.unibas.aci)

### ○ Architettura applicativa

- ⇒ interfaccia e controllo non sono significativamente diversi da quelli di altre applicazioni desktop viste
- ⇒ lo strato del modello corrisponde ai concetti della realtà di interesse
- ⇒ lo strato di persistenza è costituito di opportuni componenti di tipo DAO





## Riassumendo

- Architettura
- Problemi Tecnologici
  - ⇒ Disaccoppiamento di Impedenza
- Progetto di Riferimento



## Termini della Licenza

- This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- Questo lavoro viene concesso in uso secondo i termini della licenza "Attribution-ShareAlike" di Creative Commons. Per ottenere una copia della licenza, è possibile visitare <http://creativecommons.org/licenses/by-sa/1.0/> oppure inviare una lettera all'indirizzo Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.