

Esercizio 1

a) Scrivere le istruzioni SQL per la creazione delle seguenti tabelle esplicitando le chiavi primarie, eventuali valori non nulli e il vincolo di integrità referenziale **Opera(Autore) → Scrittore(Id)**

Opera(Codice, Titolo, Autore, NumeroAtti)

Autore(Id, Nome, Cognome, DataNascita, DataMorte, NumOpere)

b) Scrivere le istruzioni per creare le seguenti tabelle esplicitando le chiavi primarie, eventuali valori non nulli e il vincolo di integrità referenziale indicato:

Opera(ISBN VARCHAR(15),
Titolo VARCHAR(50),
Genere INTEGER, →
PrezzoIntero FLOAT,
PrezzoScontato FLOAT,
CasaEditrice VARCHAR(50))

Foreign Key: Tabella: *Genere*
Attributo: *CodiceGenere*
Politiche di reazione:
ON UPDATE CASCADE
ON DELETE SET NULL

Genere(CodiceGenere INTEGER,
Descrizione VARCHAR(50))

Esercizio 2

a) Data la base di dati “dmlabxxx” su SQLServer, contenente la tabella

Opera(Codice, Nome, Autore, NumeroAtti)

scrivere la servlet che consente all’utente di richiedere i dati di tutte le opere di Verdi. La servlet si conatterà alla base di dati, eseguirà l’interrogazione e visualizzerà i dati richiesti come lista non ordinata.

b) Data la base di dati “Esercitazioni” su SQLServer, contenente la tabella

Disco(Codice, Titolo, Autore, NumeroBrani)

scrivere la servlet che consente all’utente di richiedere tutti i dischi di un certo autore (passato come parametro). La servlet si conatterà alla base di dati, eseguirà l’interrogazione e visualizzerà i dati richiesti dentro una tabella.

c) Data la base di dati “dmlabxxx” su SQLServer, contenente la tabella:

Spettacolo(Codice, Titolo, NomeRegista, CognomeRegista, NomeTeatro, Stagione)

scrivere:

c1) La Servlet **ServletSpettacolo.java** che:

si connette alla base di dati

esegue l’interrogazione che consente di trovare titolo e nome teatro di tutti gli spettacoli di una certa stagione (la stagione viene passata come parametro dalla FORM del punto (c2))
visualizza la lista degli spettacoli come lista ordinata

c2) La FORM che invoca la Servlet **ServletSpettacolo.java** passando come parametro il nome e il cognome del regista richiesto dall’utente. Si supponga che la Servlet compilata si trovi nel context **ROOT**.

Esercizio 3

a) Scrivere un documento XML valido rispetto al seguente DTD:

```
<!ELEMENT PROGRAMMAZIONE_TV (CANALE+)>
<!ELEMENT CANALE (NOME,GIORNO+)>
<!ELEMENT NOME (#PCDATA)>
<!ELEMENT GIORNO (DATA,SLOT_PROGRAMMA+)+>
<!ELEMENT DATA (#PCDATA)>
<!ELEMENT SLOT_PROGRAMMA (ORA,TITOLO,DESCRIZIONE?)>
<!ELEMENT ORA (#PCDATA)>
<!ELEMENT TITOLO (#PCDATA)>
<!ELEMENT DESCRIZIONE (#PCDATA)>
<!ATTLIST PROGRAMMAZIONE_TV  NOME          CDATA          #IMPLIED>
<!ATTLIST CANALE              NUMERO        CDATA          #REQUIRED>
<!ATTLIST TITOLO              VOTO         CDATA          #REQUIRED>
<!ATTLIST TITOLO              LINGUA       CDATA          #IMPLIED>
```

b) Dare una rappresentazione alternativa che contiene le stesse informazioni della seguente ma che usa attributi invece di elementi. Dare anche il DTD per questa rappresentazione.

```
<banca>
  <conto>
    <numero_conto>101</numero_conto>
    <agenzia> citta'</agenzia>
    <saldo> 500 </saldo>
  </conto>
  <conto>
    <numero_conto>102</numero_conto>
    <agenzia> roncadelle</agenzia>
    <saldo> 900 </saldo>
  </conto>
  <conto>
    <numero_conto>201</numero_conto>
    <agenzia> negrar</agenzia>
    <saldo> 1500 </saldo>
  </conto>
  <cliente>
    <nome_cliente>Bianchi</nome_cliente>
    <via_cliente> Mazzini</via_cliente>
    <citta'_cliente> Verona </citta'_cliente>
  </cliente>
  <cliente>
    <nome_cliente>Verdi</nome_cliente>
    <via_cliente> Garibaldi</via_cliente>
    <citta'_cliente> Garda </citta'_cliente>
  </cliente>
</banca>
```