Matricola: _____
Cognome:_____
Nome:_____
Insegnamento: LAB Basi di dati ❑
LAB Basi di dati e web ❑
Basi di dati e web 4 CFU ❑

Laboratorio di Basi di dati (Laboratorio di Basi di dati e Web)

Basi di dati e Web (4 crediti)

Prova scritta del 23 febbraio 2011

Avvertenze: e' severamente vietato consultare libri e appunti; chiunque verrà trovato in possesso di materiale attinente al corso vedrà annullata la propria prova

Durata 1h 40m

Cognome e nome:_____ Matricola:_____

1. (4) Illustrare i metodi principali della classe HttpServletResponse (classe dell'oggetto response parametro dei metodi doGet/doPost/Service di una servlet).

Data la base di dati "Anagrafe" su SQLServer, contenente le seguenti tabelle:

COMUNE(CodiceISTAT, Nome, Superficie, DataIstituzione, Provincia, Capoluogo: {sì,no})
PERSONA(CodiceFis, Cognome, Nome, DataNascita)
RESIDENZA(Comune, Residente, DataInizio, DataFine*)
Vincoli di integrità referenziale:
RESIDENZA.Comune → COMUNE,
RESIDENZA.Residente → PERSONA.

2. (8) Progettare, secondo la metodologia basata sulla specifica di `page-schema`, lo schema logico di un sito web che presenti le informazioni contenute nella base di dati Anagrafe. In particolare:
   - Nella ***homePage*** si presenti l'elenco dei comuni riportando: il codice ISTAT e il nome del comune, il numero di residenti alla data attuale (il codice è un link verso lo schema di pagina ***comunePage***).
   - Nello schema di pagina ***comunePage*** si presentano le seguenti informazioni:
     - Il nome e codice ISTAT del comune, la superficie e la data di istituzione.
     - Si riporta inoltre il nome della provincia a cui il comune appartiene, indicando se è il capoluogo o no.
     - Infine si indica l'elenco delle persone attualmente residenti riportando per ogni persona: il nome, il cognome e la data di inizio della residenza (il nome della persona è un link verso lo schema di pagina ***personaPage***).
   - Nello schema di pagina ***personaPage*** si presentano tutte le informazioni che descrivono una persona inclusa la lista delle sue residenze indicando in ordine decrescente di data inizio: nome del comune, data inizio ed eventuale data fine della residenza.

   Lo studente progetti sia gli schemi di pagina (`page-schema`) che le interrogazioni SQL (`DB to page schema`) che li alimentano.

Per questa parte rispondere sul testo e usare eventualmente il foglio protocollo come brutta copia

3. (4) Scrivere il comando SQL per la creazione della tabella RESIDENZA supponendo già create le tabelle COMUNE e PERSONA. Si richiede di precisare la chiave primaria e i vincoli di integrità referenziale specificando la politica "cascade" in caso di aggiornamento delle tabelle "master".

4. (7) Partendo dallo schema logico progettato sopra, si completi la servlet Main.java mostrata nei fogli successivi con il codice necessario all'attivazione della JSP che implementa lo schema di pagina ***comunePage***. Si supponga presente un parametro "ps" che indica la JSP da attivare. Se il parametro è assente viene invocata la JSP che implementa la ***homePage***. Vanno scelti i parametri aggiuntivi (che si ipotizzano presenti nella richiesta HTTP) necessari per le interrogazioni che alimentano lo schema di pagina ***comunePage***.

5. (10) Si completi il codice come di seguito descritto:

   - Nella classe JAVA DBMS contenente i metodi:
     i. *public ComuneBean getComune(String codice),*
     ii. *public Vector getResidenti(String codiceComune),*

     si implementi il metodo *getResidenti(String codiceComune),* che restituisce un vettore di oggetti della classe PersonaBean da passare alla JSP comunePage per produrre l'elenco dei residenti presso il comune.
   - Si implementino i metodi get e un costruttore per il Java Data Bean: *PersonaBean* (proprietà: *codiceFiscale, nome, cognome, dataNascita*); è disponibile e non va implementato il Java Data Bean: *ComuneBean* (proprietà: *nome, codice, superficie, dataIstituzione, provincia, capoluogo*).
   - Si implementi infine la JSP ***comunePage*** che presenta le informazioni come richiesto dalla specifica dello schema di pagina corrispondente.

DOMANDA OBBLIGATORIA SOLO PER BASI DI DATI E WEB (4 CREDITI)

6. (4) Descrivere le caratteristiche principali delle JSP e mostrare in particolare i tag (elementi XML) disponibili per gestire oggetti Java conformi allo standard Java Data Bean.

Cognome e nome:_____ Matricola:_____

main.java

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import beanComuni.*;

public class main extends HttpServlet {
      public void doGet(HttpServletRequest request, HttpServletResponse response)
         throws IOException, ServletException {
            //Definizione e recupero dell'eventuale parametro ps della servlet
            String ps = "";
            RequestDispatcher rd = null;

            if (request.getParameter("ps") != null) {
                  ps =
            }
            try {
                  // Oggetto per l'interazione con il Database


                  if (ps.equals("")) {
                        // Parametro ps assente o vuoto, viene attivata la
                        // home page del sito. Non implementare

                  } else if
                        // Implementare quanto è necessario per l'attivazione
                        // della jsp comunePage.jsp

            } catch(Exception e) {
                  e.printStackTrace();
                  }
      }
}
```

Cognome e nome:_____ Matricola:_____

comunePage.jsp

```
<%@page import="java.io.*"%>
<%@page import="java.util.*"%>
<%@page import="beanComuni.*"%>
<html>
<%
//Dichiaro un bean che conterrà i dati del comune
ComuneBean comune;
//Recupero il bean passato come attributo contenente i dati del comune

comune =

//Recupero il vector passato come attributo contenente la lista dei residenti

Vector residenti =

%>

<!--Inizio la pagina HTML-->
<head>
<title>Pagina che descrive un comune e i suoi residenti</title>
</head>
<body>









</body>
</html>
```

Cognome e nome:_____ Matricola:_____

```
public class PersonaBean {


// Implementare un costruttore


// Implementare i metodi GET









// NON implementare i metodi SET
}

public class ComuneBean {
…
} NON IMPLEMENTARE!
```

DBMS.java

```
import java.sql.*;
import java.util.*;
import beanComuni.*

public class DBMS {
      // Dati di identificazione dell'utente

      // URL per la connessione alla base di dati

      // Driver da utilizzare per la connessione JDBC


      // Definizione dell'interrogazione SQL per il metodo getResidenti()
```

Cognome e nome:_____ Matricola:_____

```
	//Implementare il costruttore della classe DBMS




	//Metodi per la creazione di un bean a partire dal record attuale
	//del ResultSet dato come parametro
	private ComuneBean makeComuneBean (
	} NON IMPLEMENTARE!

	private PersonaBean makeResidenteBean (
	} NON IMPLEMENTARE!

	public Vector getResidenti(                                          ) {
		// Dichiarazione delle variabili necessarie
		Connection con = null;
		PreparedStatement pstmt = null;
		ResultSet rs = null;
		Vector result = new Vector();
		try {









	}

	public ComuneBean getComune (String nome) {
	} // NON IMPLEMENTARE!
} // classe DBMS.java
```