**Hibernate.cfg.xml**

**<?xml version=*"1.0"* encoding=*"UTF-8"*?>**

**<!DOCTYPE hibernate-configuration SYSTEM**

**"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">**

**<hibernate-configuration>**

**<session-factory>**

**<property name=*"format\_sql"*>false</property>**

**<property name=*"connection.pool\_size"*>2</property>**

**<property name = *"hibernate.dialect"*>**

**org.hibernate.dialect.MySQL5Dialect**

**</property>**

**<property name = *"hibernate.connection.driver\_class"*>**

**com.mysql.cj.jdbc.Driver**

**</property>**

**<!-- Assume test is the database name -->**

**<property name = *"hibernate.connection.url"*>**

**jdbc:mysql://localhost:3306/practice**

**</property>**

**<property name = *"hibernate.connection.username"*>**

**root**

**</property>**

**<property name = *"hibernate.connection.password"*>**

**suriya**

**</property>**

**<!-- Disable 2nd level cache -->**

**<property name=*"cache.provider\_class"*>org.hibernate.cache.internal.NoCacheProvider</property>**

**<property name=*"show\_sql"*>true</property>**

**<property name=*"hbm2ddl.auto"*>update</property>**

**<!-- <property name="hbm2ddl.auto">create</property> -->**

**<mapping class=*"beans.Classes"*/>**

**<mapping class=*"beans.Subject"*/>**

**<mapping class=*"beans.Teacher"*/>**

**<mapping class=*"beans.Student"*/>**

**</session-factory>**

**</hibernate-configuration>**

**Index.jsp**

**<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"***

**pageEncoding=*"ISO-8859-1"*%>**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset=*"ISO-8859-1"*>**

**<title>Menu</title>**

**</head>**

**<body>**

**<a href=*"view.jsp"* style="color: *blue*;">view class information</a><br>**

**<a href=*"add\_class.jsp"* style="color: *green*;">Add class</a><br>**

**</body>**

**</html>**

**View.jsp**

**<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"***

**pageEncoding=*"ISO-8859-1"*%>**

**<%@ page import=*"java.util.List"***

**import=*"org.hibernate.Session"***

**import=*"org.hibernate.SessionFactory"***

**import=*"org.hibernate.cfg.Configuration"***

**import=*"java.util.Iterator"***

**import=*"beans.Classes"***

**import=*"beans.Subject"***

**import=*"beans.Teacher"***

**import=*"beans.Student"***

**import=*"java.util.Set"***

**import=*"javax.persistence.TypedQuery"*%>**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset=*"ISO-8859-1"*>**

**<title>Insert title here</title>**

**</head>**

**<body>**

**<table width=*"400"* border=*"1"*>**

**<tr><th>Class Id</th><th>Class</th><th>Subjects</th><th>Teachers</th><th>Students</th></tr>**

**<%**

**Configuration cf=new Configuration();**

**SessionFactory sf = cf.configure().buildSessionFactory();**

**Session session1 =sf.openSession();**

**//Using from Clause**

**@SuppressWarnings("unchecked")**

**List<Classes> list= session1.createQuery("from Classes").list();**

**for(Classes e : list) {**

**int id = e.getClassId();**

**String name =e.getClassName();**

**Set<Subject> subs =e.getSubjects();**

**Set<Teacher> teachers =e.getTeachers();**

**Set<Student> students =e.getStudents();**

**%>**

**<tr>**

**<td><%=id%></td>**

**<td><%=name%></td>**

**<td><%**

**for(Subject s: subs){**

**out.println(s.getSubjectName()+"<br>");**

**}**

**%>**

**</td>**

**<td><%**

**for(Teacher t: teachers){**

**out.println(t.getTeacherName()+"<br>");**

**}**

**%></td>**

**<td><%**

**for(Student st: students){**

**out.println(st.getStudentName()+"<br><br>");**

**}**

**%></td>**

**</tr>**

**<%**

**}**

**session1.close();**

**%>**

**</table>**

**</body>**

**</html>**

**Class:classes**

**package beans;**

**import java.util.HashSet;**

**import java.util.Set;**

**import javax.persistence.CascadeType;**

**import javax.persistence.Column;**

**import javax.persistence.Entity;**

**import javax.persistence.GeneratedValue;**

**import javax.persistence.GenerationType;**

**import javax.persistence.Id;**

**import javax.persistence.JoinColumn;**

**import javax.persistence.JoinTable;**

**import javax.persistence.ManyToMany;**

**import javax.persistence.OneToMany;**

**import javax.persistence.SequenceGenerator;**

**import javax.persistence.Table;**

**@Entity**

**@Table**

**public class Classes {**

**@Id**

**@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "cls\_gen")**

**@SequenceGenerator(name="cls\_gen", sequenceName = "cls\_seq")**

**private int classId;**

**@Column(length = 20)**

**private String className;**

**@ManyToMany(cascade = CascadeType.ALL)**

**@JoinTable(name="class\_subject", joinColumns = {@JoinColumn(name = "classId")}, inverseJoinColumns = {@JoinColumn(name = "subjectId")})**

**Set<Subject> subjects = new HashSet<Subject>();**

**@ManyToMany(cascade = CascadeType.ALL)**

**@JoinTable(name="class\_teacher", joinColumns = {@JoinColumn(name = "classId")}, inverseJoinColumns = {@JoinColumn(name = "teacherId")})**

**Set<Teacher> teachers = new HashSet<Teacher>();**

**@OneToMany(cascade = CascadeType.ALL)**

**@JoinTable(name="class\_student", joinColumns = {@JoinColumn(name = "classId")}, inverseJoinColumns = {@JoinColumn(name = "studentId")})**

**Set<Student> students = new HashSet<Student>();**

**public Set<Student> getStudents() {**

**return students;**

**}**

**public void setStudents(Set<Student> students) {**

**this.students = students;**

**}**

**public int getClassId() {**

**return classId;**

**}**

**public void setClassId(int classId) {**

**this.classId = classId;**

**}**

**public String getClassName() {**

**return className;**

**}**

**public void setClassName(String className) {**

**this.className = className;**

**}**

**public Set<Subject> getSubjects() {**

**return subjects;**

**}**

**public void setSubjects(Set<Subject> subjects) {**

**this.subjects = subjects;**

**}**

**public Set<Teacher> getTeachers() {**

**return teachers;**

**}**

**public void setTeachers(Set<Teacher> teachers) {**

**this.teachers = teachers;**

**}**

**}**

**Class:student**

**package beans;**

**import javax.persistence.Entity;**

**import javax.persistence.GeneratedValue;**

**import javax.persistence.GenerationType;**

**import javax.persistence.Id;**

**import javax.persistence.SequenceGenerator;**

**import javax.persistence.Table;**

**@Entity**

**@Table**

**public class Student {**

**@Id**

**@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "stu\_gen")**

**@SequenceGenerator(name="stu\_gen", sequenceName = "stu\_seq")**

**private int studentId;**

**private String studentName;**

**public int getStudentId() {**

**return studentId;**

**}**

**public void setStudentId(int studentId) {**

**this.studentId = studentId;**

**}**

**public String getStudentName() {**

**return studentName;**

**}**

**public void setStudentName(String string) {**

**this.studentName = string;**

**}**

**}**

**Class:teacher**

**package beans;**

**import java.util.HashSet;**

**import java.util.Set;**

**import javax.persistence.Entity;**

**import javax.persistence.GeneratedValue;**

**import javax.persistence.GenerationType;**

**import javax.persistence.Id;**

**import javax.persistence.JoinColumn;**

**import javax.persistence.JoinTable;**

**import javax.persistence.ManyToMany;**

**import javax.persistence.OneToMany;**

**import javax.persistence.SequenceGenerator;**

**import javax.persistence.Table;**

**@Entity**

**@Table**

**public class Teacher {**

**@Id**

**@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "teach\_gen")**

**@SequenceGenerator(name="teach\_gen", sequenceName = "teach\_seq")**

**private int teacherId;**

**private String teacherName;**

**@ManyToMany(mappedBy = "teachers")**

**Set<Classes> classes = new HashSet<Classes>();**

**@OneToMany**

**@JoinTable(name="subject\_teacher", joinColumns = {@JoinColumn(name = "teacherId")}, inverseJoinColumns = {@JoinColumn(name = "subjectId")})**

**Set<Subject> subjects = new HashSet<Subject>();**

**public int getTeacherId() {**

**return teacherId;**

**}**

**public void setTeacherId(int teacherId) {**

**this.teacherId = teacherId;**

**}**

**public String getTeacherName() {**

**return teacherName;**

**}**

**public void setTeacherName(String teacherName) {**

**this.teacherName = teacherName;**

**}**

**public Set<Classes> getClasses() {**

**return classes;**

**}**

**public void setClasses(Set<Classes> classes) {**

**this.classes = classes;**

**}**

**public Set<Subject> getSubjects() {**

**return subjects;**

**}**

**public void setSubjects(Set<Subject> subjects) {**

**this.subjects = subjects;**

**}**

**}**

**Class :subject**

**package beans;**

**import java.util.HashSet;**

**import java.util.Set;**

**import javax.persistence.CascadeType;**

**import javax.persistence.Entity;**

**import javax.persistence.GeneratedValue;**

**import javax.persistence.GenerationType;**

**import javax.persistence.Id;**

**import javax.persistence.JoinColumn;**

**import javax.persistence.JoinTable;**

**import javax.persistence.ManyToMany;**

**import javax.persistence.ManyToOne;**

**import javax.persistence.OneToMany;**

**import javax.persistence.SequenceGenerator;**

**import javax.persistence.Table;**

**@Entity**

**@Table**

**public class Subject {**

**@Id**

**@GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "sub\_gen")**

**@SequenceGenerator(name="sub\_gen", sequenceName = "sub\_seq")**

**private int subjectId;**

**private String subjectName;**

**@ManyToMany(mappedBy = "subjects")**

**Set<Classes> classes = new HashSet<Classes>();;**

**public int getSubjectId() {**

**return subjectId;**

**}**

**public void setSubjectId(int subjectId) {**

**this.subjectId = subjectId;**

**}**

**public String getSubjectName() {**

**return subjectName;**

**}**

**public void setSubjectName(String subjectName) {**

**this.subjectName = subjectName;**

**}**

**public Set<Classes> getClasses() {**

**return classes;**

**}**

**public void setClasses(Set<Classes> classes) {**

**this.classes = classes;**

**}**

**}**

**Servlet :AddclassServlet**

**package services;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import javax.servlet.http.HttpSession;**

**import org.hibernate.Session;**

**import org.hibernate.SessionFactory;**

**import org.hibernate.cfg.Configuration;**

**import beans.Classes;**

**import beans.Student;**

**import beans.Subject;**

**import beans.Teacher;**

**/\*\***

**\* Servlet implementation class AddClassServlet**

**\*/**

**public class AddClassServlet extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**@SuppressWarnings("unchecked")**

**@Override**

**protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**PrintWriter out = response.getWriter();**

**String className = request.getParameter("class-name");**

**String subject1 = request.getParameter("sub1-name");**

**String subject2 = request.getParameter("sub2-name");**

**String subject3 = request.getParameter("sub3-name");**

**String sub1\_teacher = request.getParameter("teacher1-name");**

**String sub2\_teacher = request.getParameter("teacher2-name");**

**String sub3\_teacher = request.getParameter("teacher3-name");**

**String std1 = request.getParameter("student1-name");**

**String std2 = request.getParameter("student2-name");**

**String std3 = request.getParameter("student3-name");**

**String std4 = request.getParameter("student4-name");**

**Classes classObj = new Classes();**

**classObj.setClassName(className);**

**Subject s1 = new Subject();**

**s1.setSubjectName(subject1);**

**Subject s2 = new Subject();**

**s2.setSubjectName(subject2);**

**Subject s3 = new Subject();**

**s3.setSubjectName(subject3);**

**Teacher t1 = new Teacher();**

**t1.setTeacherName(sub1\_teacher);**

**Teacher t2 = new Teacher();**

**t2.setTeacherName(sub2\_teacher);**

**Teacher t3 = new Teacher();**

**t3.setTeacherName(sub3\_teacher);**

**//students creation**

**Student st1 = new Student();**

**st1.setStudentName(std1);**

**Student st2 = new Student();**

**st2.setStudentName(std2);**

**Student st3 = new Student();**

**st3.setStudentName(std3);**

**Student st4 = new Student();**

**st4.setStudentName(std4);**

**classObj.getSubjects().add(s1);**

**classObj.getSubjects().add(s2);**

**classObj.getSubjects().add(s3);**

**s1.getClasses().add(classObj);**

**s2.getClasses().add(classObj);**

**s3.getClasses().add(classObj);**

**classObj.getTeachers().add(t1);**

**classObj.getTeachers().add(t2);**

**classObj.getTeachers().add(t3);**

**t1.getClasses().add(classObj);**

**t2.getClasses().add(classObj);**

**t3.getClasses().add(classObj);**

**t1.getSubjects().add(s1);**

**t2.getSubjects().add(s2);**

**t3.getSubjects().add(s3);**

**classObj.getStudents().add(st1);**

**classObj.getStudents().add(st2);**

**classObj.getStudents().add(st3);**

**classObj.getStudents().add(st4);**

**HttpSession httpSession = request.getSession();**

**SessionFactory sf = new Configuration().configure().buildSessionFactory();**

**Session session = sf.openSession();**

**session.beginTransaction();**

**session.saveOrUpdate(classObj);**

**session.getTransaction().commit();**

**}**

**}**

**Servlet:adminservlet**

**package services;**

**import java.io.IOException;**

**import java.io.PrintWriter;**

**import java.util.HashSet;**

**import java.util.Set;**

**import javax.servlet.ServletException;**

**import javax.servlet.http.HttpServlet;**

**import javax.servlet.http.HttpServletRequest;**

**import javax.servlet.http.HttpServletResponse;**

**import javax.servlet.http.HttpSession;**

**import org.hibernate.Query;**

**import org.hibernate.Session;**

**import org.hibernate.SessionFactory;**

**import org.hibernate.cfg.Configuration;**

**import beans.Classes;**

**import beans.Student;**

**import beans.Subject;**

**import beans.Teacher;**

**/\*\***

**\* Servlet implementation class AdminServlet**

**\*/**

**public class AdminServlet extends HttpServlet {**

**private static final long serialVersionUID = 1L;**

**@SuppressWarnings("null")**

**@Override**

**protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {**

**// response.setContentType("text/html");**

**PrintWriter out = response.getWriter();**

**//Common for all classes**

**Set<Classes> classes = new HashSet<Classes>();**

**//11th sets**

**Set<Subject> subjects = new HashSet<Subject>();**

**Set<Teacher> teachers = new HashSet<Teacher>();**

**Set<Student> students = new HashSet<Student>();**

**//12th sets**

**Set<Subject> twelth\_subjects = new HashSet<Subject>();**

**Set<Teacher> twelth\_teachers = new HashSet<Teacher>();**

**Set<Student> twelth\_students = new HashSet<Student>();**

**//10th**

**Set<Subject> tenth\_subjects = new HashSet<Subject>();**

**Set<Teacher> tenth\_teachers = new HashSet<Teacher>();**

**Set<Student> tenth\_students = new HashSet<Student>();**

**//11th subs**

**Subject maths = new Subject();**

**maths.setSubjectName("11th Mathematics");**

**Subject chemistry = new Subject();**

**chemistry.setSubjectName("11th Chemistry");**

**Subject physics = new Subject();**

**physics.setSubjectName("11th Physics");**

**//12th subs**

**Subject twelth\_maths = new Subject();**

**twelth\_maths.setSubjectName("12th Mathematics");**

**Subject twelth\_chemistry = new Subject();**

**twelth\_chemistry.setSubjectName("12th Chemistry");**

**Subject twelth\_physics = new Subject();**

**twelth\_physics.setSubjectName("12th Physics");**

**//10th subs**

**Subject tenth\_science = new Subject();**

**tenth\_science.setSubjectName("10th Science");**

**Subject tenth\_maths = new Subject();**

**tenth\_maths.setSubjectName("10th Mathematics");**

**Subject tenth\_tamil = new Subject();**

**tenth\_tamil.setSubjectName("10th Tamil");**

**//teachers**

**Teacher t1 = new Teacher();**

**t1.setTeacherName("Malar");**

**Teacher t2 = new Teacher();**

**t2.setTeacherName("Ashok");**

**Teacher t3 = new Teacher();**

**t3.setTeacherName("sekar");**

**//10th students**

**Student s1 = new Student();**

**s1.setStudentName("Nandha");**

**Student s2 = new Student();**

**s2.setStudentName("Kumar");**

**Student s3 = new Student();**

**s3.setStudentName("Velu");**

**Student s4 = new Student();**

**s4.setStudentName("Harish");**

**//11th students**

**Student s5 = new Student();**

**s5.setStudentName("Nikil");**

**Student s6 = new Student();**

**s6.setStudentName("Nalini");**

**Student s7 = new Student();**

**s7.setStudentName("Seetha");**

**//12th students**

**Student s8 = new Student();**

**s8.setStudentName("Swetha");**

**Student s9 = new Student();**

**s9.setStudentName("Nandhini");**

**Student s10 = new Student();**

**s10.setStudentName("Kishore");**

**Student s11 = new Student();**

**s11.setStudentName("Rahul");**

**//classes**

**Classes eleventh = new Classes();**

**eleventh.setClassName("Eleventh standard");**

**Classes tenth = new Classes();**

**tenth.setClassName("Tenth standard");**

**Classes twelth = new Classes();**

**twelth.setClassName("Twelth standard");**

**subjects.add(maths);**

**subjects.add(physics);**

**subjects.add(chemistry);**

**twelth\_subjects.add(twelth\_physics);**

**twelth\_subjects.add(twelth\_maths);**

**twelth\_subjects.add(twelth\_chemistry);**

**tenth\_subjects.add(tenth\_tamil);**

**tenth\_subjects.add(tenth\_science);**

**tenth\_subjects.add(tenth\_maths);**

**// t1.setSubjects(subjects);**

**// t2.setSubjects(subjects);**

**// t3.setSubjects(subjects);**

**teachers.add(t1);**

**teachers.add(t2);**

**teachers.add(t3);**

**twelth\_teachers.add(t1);**

**twelth\_teachers.add(t2);**

**twelth\_teachers.add(t3);**

**tenth\_teachers.add(t1);**

**tenth\_teachers.add(t2);**

**tenth\_teachers.add(t3);**

**classes.add(eleventh);**

**classes.add(tenth);**

**classes.add(twelth);**

**t1.setClasses(classes);**

**t2.setClasses(classes);**

**t3.setClasses(classes);**

**maths.setClasses(classes);**

**physics.setClasses(classes);**

**chemistry.setClasses(classes);**

**twelth\_maths.setClasses(classes);**

**twelth\_physics.setClasses(classes);**

**twelth\_chemistry.setClasses(classes);**

**tenth\_maths.setClasses(classes);**

**tenth\_science.setClasses(classes);**

**tenth\_tamil.setClasses(classes);**

**eleventh.setTeachers(teachers);**

**eleventh.setSubjects(subjects);**

**tenth.setTeachers(tenth\_teachers);**

**tenth.setSubjects(tenth\_subjects);**

**twelth.setTeachers(twelth\_teachers);**

**twelth.setSubjects(twelth\_subjects);**

**tenth\_students.add(s1);**

**tenth\_students.add(s2);**

**tenth\_students.add(s3);**

**tenth\_students.add(s4);**

**students.add(s5);**

**students.add(s6);**

**students.add(s7);**

**twelth\_students.add(s8);**

**twelth\_students.add(s9);**

**twelth\_students.add(s10);**

**twelth\_students.add(s10);**

**eleventh.setStudents(students);**

**twelth.setStudents(twelth\_students);**

**tenth.setStudents(tenth\_students);**

**for(Teacher t: eleventh.getTeachers()) {**

**if(t.getTeacherName().equals("Malar")) {**

**t.getSubjects().add(physics);**

**}else if(t.getTeacherName().equals("Ashok")) {**

**t.getSubjects().add(maths);**

**}else {**

**t.getSubjects().add(chemistry);**

**}**

**}**

**for(Teacher t: twelth.getTeachers()) {**

**if(t.getTeacherName().equalsIgnoreCase("malar")) {**

**t.getSubjects().add(twelth\_maths);**

**}else if(t.getTeacherName().equals("Ashok")) {**

**t.getSubjects().add(twelth\_physics);**

**}else {**

**t.getSubjects().add(twelth\_chemistry);**

**}**

**}**

**for(Teacher t: tenth.getTeachers()) {**

**if(t.getTeacherName().equalsIgnoreCase("malar")) {**

**t.getSubjects().add(tenth\_maths);**

**}else if(t.getTeacherName().equals("Ashok")) {**

**t.getSubjects().add(tenth\_science);**

**}else {**

**t.getSubjects().add(tenth\_tamil);**

**}**

**}**

**HttpSession httpSession = request.getSession();**

**httpSession.setAttribute("classes", classes);**

**SessionFactory sf = new Configuration().configure().buildSessionFactory();**

**Session session = sf.openSession();**

**session.beginTransaction();**

**int i =0;**

**for(Classes cl: classes) {**

**i= (Integer)session.save(cl);**

**}**

**session.getTransaction().commit();**

**if(i>0) {**

**out.print("Record inserted");**

**}else {**

**out.print("Record insertion failed");**

**}**

**}**

**}**