**LAB-4:**

**TODO:1**

**Create Instructor Table with column id as a primary key(pk),Firstname,Lastname,Email,**

**Todo1.1:Create Forgien key column name as Instrcutor\_Details\_id in Instrcutor table as the reference id of InstrcutorDetails dB**

**Todo1.2:Create InstructorDetails Table with column id as a primary key(pk),Coursename,Review**

**Todo 1.3:Create Course Table with Column id as a primary key(pk),title,Instrcutor-id as forgien key reference to the id column of Instructor table**

**TODO:2**

**Todo2.1:Create com.hiberntae.practice,com.hibernate.practice.Entity package under src/main/java folder**

**Todo2.2:Create Instrctor.java,ImstrcutorDetails.java,Courses.java classes under com.hibernate.practice.Entity**

**Todo 2.3:CreateDemo.java,DeleteDemo.java classes under com.hibernate.practice**

**ToDo:3**

**Todo3.1:Create com.hibernate.utils package under src/main/resource folder**

**Todo 3.2:Create hibernate.cfg.xml file under src/main/resource folder**

**Todo 3.3:Do the necessary Configuration for DB**

**Instrcutor.java**

package com.hibernate.demo.entity;

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.OneToOne;

import javax.persistence.Table;

@Entity

@Table(name="instructor")

public class Instructor {

    // annotate the class as an entity and map to db table

    // define the fields

    // annotate the fields with db column names

    // \*\* set up mapping to InstructorDetail entity

    // create constructors

    // generate getter/setter methods

    // generate toString() method

    @Id

    @GeneratedValue(strategy=GenerationType.IDENTITY)

    @Column(name="id")

    private int id;

    @Column(name="first\_name")

    private String firstName;

    @Column(name="last\_name")

    private String lastName;

    @Column(name="email")

    private String email;

    @OneToOne(cascade=CascadeType.ALL)

    @JoinColumn(name="instructor\_detail\_id")

    private InstructorDetail instructorDetail;

     @OneToMany(mappedBy="instructor",

               cascade= {CascadeType.PERSIST, CascadeType.MERGE,

                         CascadeType.DETACH, CascadeType.REFRESH})

    private List<Course> courses;

    public Instructor() {

    }

    public Instructor(String firstName, String lastName, String email) {

        this.firstName = firstName;

        this.lastName = lastName;

        this.email = email;

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getFirstName() {

        return firstName;

    }

    public void setFirstName(String firstName) {

        this.firstName = firstName;

    }

    public String getLastName() {

        return lastName;

    }

    public void setLastName(String lastName) {

        this.lastName = lastName;

    }

    public String getEmail() {

        return email;

    }

    public void setEmail(String email) {

        this.email = email;

    }

    public InstructorDetail getInstructorDetail() {

        return instructorDetail;

    }

    public void setInstructorDetail(InstructorDetail instructorDetail) {

        this.instructorDetail = instructorDetail;

    }

    @Override

    public String toString() {

        return "Instructor [id=" + id + ", firstName=" + firstName + ", lastName=" + lastName + ", email=" + email

                + ", instructorDetail=" + instructorDetail + "]";

    }

    public List<Course> getCourses() {

        return courses;

    }

    public void setCourses(List<Course> courses) {

        this.courses = courses;

    }

    // add convenience methods for bi-directional relationship

    public void add(Course tempCourse) {

        if (courses == null) {

            courses = new ArrayList<>();

        }

        courses.add(tempCourse);

        tempCourse.setInstructor(this);

    }

}

**InstrcutorDetails.Java**

package com.hibernate.demo.entity;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name="instructor\_detail")

public class InstructorDetail {

    // annotate the class as an entity and map to db table

    // define the fields

    // annotate the fields with db column names

    // create constructors

    // generate getter/setter methods

    // generate toString() method

     @Id

    @GeneratedValue(strategy=GenerationType.IDENTITY)

    @Column(name="id")

    private int id;

    @Column(name="Course\_name")

    private String coursename;

    @Column(name="Review")

    private String review;

    public InstructorDetail() {

    }

    public InstructorDetail(String coursename, String review) {

        this.course\_name = coursename;

        this.review = review;

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getCoursename() {

        return coursename;

    }

    public void setcoursename(String coursename) {

        this.coursename = coursename;

    }

    public String getreview() {

        return review;

    }

    public void setreview(String review) {

        this.review = review;

    }

    @Override

    public String toString() {

        return "InstructorDetail [id=" + id + ", coursename=" + coursename + ", review=" + review + "]";

    }

}

**Course.java**

package com.hibernate.demo.entity;

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.ManyToOne;

import javax.persistence.Table;

@Entity

@Table(name="course")

public class Course {

    // define our fields

    // define constructors

    // define getter setters

    // define tostring

    // annotate fields

    @Id

    @GeneratedValue(strategy=GenerationType.IDENTITY)

    @Column(name="id")

    private int id;

    @Column(name="title")

    private String title;

    @ManyToOne(cascade= {CascadeType.PERSIST, CascadeType.MERGE,

                         CascadeType.DETACH, CascadeType.REFRESH})

    @JoinColumn(name="instructor\_id")

    private Instructor instructor;

    public Course() {

    }

    public Course(String title) {

        this.title = title;

    }

    public int getId() {

        return id;

    }

    public void setId(int id) {

        this.id = id;

    }

    public String getTitle() {

        return title;

    }

    public void setTitle(String title) {

        this.title = title;

    }

    public Instructor getInstructor() {

        return instructor;

    }

    public void setInstructor(Instructor instructor) {

        this.instructor = instructor;

    }

    @Override

    public String toString() {

        return "Course [id=" + id + ", title=" + title + "]";

    }

}

**CreateDemo.java**

package com.code.hibernate.demo;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import com.hibernate.demo.entity.Instructor;

import com.hibernate.demo.entity.InstructorDetail;

import com.hibernate.demo.entity.Student;

public class CreateDemo {

    public static void main(String[] args) {

        // create session factory

        SessionFactory factory = new Configuration()

                                .configure("hibernate.cfg.xml")

                                .addAnnotatedClass(Instructor.class)

                                .addAnnotatedClass(InstructorDetail.class)

                                .buildSessionFactory();

        // create session

        Session session = factory.getCurrentSession();

        try {

            // create the objects

            /\*

            Instructor tempInstructor =

                    new Instructor("Santhiya” “kowsi” “santhiya@yahoo.com");

            InstructorDetail tempInstructorDetail =

                    new InstructorDetail(“1”

                            "Java",

                            "5");

            \*/

            Instructor tempInstructor =

                    new Instructor("Madhu", "Patel", "madhu@gmail.com");

            InstructorDetail tempInstructorDetail =

                    new InstructorDetail( “2”

                            ".NET",

                            "4.97");

            // associate the objects

            tempInstructor.setInstructorDetail(tempInstructorDetail);

            // start a transaction

            session.beginTransaction();

            // save the instructor

            //

            // Note: this will ALSO save the details object

            // because of CascadeType.ALL

            //

            System.out.println("Saving instructor: " + tempInstructor);

            session.save(tempInstructor);

            // commit transaction

            session.getTransaction().commit();

            System.out.println("Done!");

        }

        finally {

            factory.close();

        }

    }

}

**Delete\_Demo.Java (App.java/Main.java)**

package com.hibernate.demo;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import com.hibernate.demo.entity.Instructor;

import com.hibernate.demo.entity.InstructorDetail;

public class DeleteDemo {

    public static void main(String[] args) {

        // create session factory

        SessionFactory factory = new Configuration()

                                .configure("hibernate.cfg.xml")

                                .addAnnotatedClass(Instructor.class)

                                .addAnnotatedClass(InstructorDetail.class)

                                .buildSessionFactory();

        // create session

        Session session = factory.getCurrentSession();

        try {

            // start a transaction

            session.beginTransaction();

            // get instructor by primary key / id

            int theId = 1;

            Instructor tempInstructor =

                    session.get(Instructor.class, theId);

            System.out.println("Found instructor: " + tempInstructor);

            // delete the instructors

            if (tempInstructor != null) {

                System.out.println("Deleting: " + tempInstructor);

                // Note: will ALSO delete associated "details" object

                // because of CascadeType.ALL

                //

                session.delete(tempInstructor);

            }

            // commit transaction

            session.getTransaction().commit();

            System.out.println("Done!");

        }

        finally {

            factory.close();

        }

    }

}

**CreateCourse.java**

package com.hibernate.demo;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import com.hibernate.demo.entity.Course;

import com.hibernate.demo.entity.Instructor;

import com.hibernate.demo.entity.InstructorDetail;

public class CreateCoursesDemo {

    public static void main(String[] args) {

        // create session factory

        SessionFactory factory = new Configuration()

                                .configure("hibernate.cfg.xml")

                                .addAnnotatedClass(Instructor.class)

                                .addAnnotatedClass(InstructorDetail.class)

                                .addAnnotatedClass(Course.class)

                                .buildSessionFactory();

        // create session

        Session session = factory.getCurrentSession();

        try {

            // start a transaction

            session.beginTransaction();

            // get the instructor from db

            int theId = 1;

            Instructor tempInstructor = session.get(Instructor.class, theId);

            // create some courses

            Course tempCourse1 = new Course("JAVA");

            Course tempCourse2 = new Course("SAP-ABAP");

            // add courses to instructor

            tempInstructor.add(tempCourse1);

            tempInstructor.add(tempCourse2);

            // save the courses

            session.save(tempCourse1);

            session.save(tempCourse2);

            // commit transaction

            session.getTransaction().commit();

            System.out.println("Done!");

        }

        finally {

            // add clean up code

            session.close();

            factory.close();

        }

    }

}

**DeleteCourse.java**

package com.hibernate.demo;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

import com.hibernate.demo.entity.Course;

import com.hibernate.demo.entity.Instructor;

import com.hibernate.demo.entity.InstructorDetail;

public class DeleteCourseDemo {

    public static void main(String[] args) {

        // create session factory

        SessionFactory factory = new Configuration()

                                .configure("hibernate.cfg.xml")

                                .addAnnotatedClass(Instructor.class)

                                .addAnnotatedClass(InstructorDetail.class)

                                .addAnnotatedClass(Course.class)

                                .buildSessionFactory();

        // create session

        Session session = factory.getCurrentSession();

        try {

            // start a transaction

            session.beginTransaction();

            // get a course

            int theId = 10;

            Course tempCourse = session.get(Course.class, theId);

            // delete course

            System.out.println("Deleting course: " + tempCourse);

            session.delete(tempCourse);

            // commit transaction

            session.getTransaction().commit();

            System.out.println("Done!");

        }

        finally {

            // add clean up code

            session.close();

            factory.close();

        }

    }

}