**Name : Chinmay Makrand Pimpalgaonkar**

**Student ID : WDGET2024026**

**Email ID:** [**chinmaymp007@gmail.com**](mailto:chinmaymp007@gmail.com)

**Task 1 and Task 2 are Integrated in this PDF**

**Assignment on Hibernate**

--------------------------------------------------------------------------------------------------------------------------

**Task 1 :** 1**. (i) Create a student class with the following members regno,name and EmailId. Save the student object in the database using hibernate. (ii)Alter student class with marks data member. (iii) Update mark data (iii) Fetch the object from the table. (iv) delete the object.**

**Github URL :**

**Student.java**

**package** com.chinmay.entity;

**import** java.io.Serializable;

**public** **class** Student **implements** Serializable {

**private** **int** regno;

**private** String name;

**private** String email;

**private** **int** marks; // altered after

**public** **int** getMarks() {

**return** marks;

}

**public** **void** setMarks(**int** marks) {

**this**.marks = marks;

}

**public** **int** getRegno() {

**return** regno;

}

**public** **void** setRegno(**int** regno) {

**this**.regno = regno;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

}

**Student.hbm.xml**

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

<!-- Import DTD rules -->

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

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

<hibernate-mapping>

<class name=*"com.chinmay.entity.Student"* table=*"student"*>

<!-- Name = data type name column is table column -->

<id name = *"regno"* column=*"regno"*></id>

<property name=*"name"* column = *"name"*></property>

<property name=*"email"* column = *"email"*></property>

<property name=*"marks"* column = *"marks"*></property>

</class>

</hibernate-mapping>

**hibernate.cfg.xml**

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

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

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

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->

<property name=*"hibernate.connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"hibernate.connection.url"*>jdbc:mysql://localhost:3306/chinmay</property>

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

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

<mapping resource =*"com/chinmay/entity/Student.hbm.xml"* />

</session-factory>

</hibernate-configuration>

**MainJava.java**

package com.chinmay.test;

import org.hibernate.HibernateException;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.Configuration;

import com.chinmay.entity.Student;

import com.mysql.cj.Session;

public class MainJava {

public static void addObject(org.hibernate.Session session,SessionFactory sessionFactory, Transaction transaction) {

try {

session = sessionFactory.openSession();

transaction = session.beginTransaction();

Student sd= new Student();

sd.setRegno(45);

sd.setName("Chinmayyy");

sd.setEmail("chinmay@gmail.com");

sd.setMarks(100);

session.save(sd);

// Commit the transaction

transaction.commit();

System.out.println("Product saved successfully.");

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

} finally {

if (session != null) {

session.close();

}

if (sessionFactory != null) {

sessionFactory.close();

}

}

}

public static void updateMarks(org.hibernate.Session session,SessionFactory sessionFactory, Transaction transaction) {

try {

session = sessionFactory.openSession();

transaction = session.beginTransaction();

System.out.println("The REg No of student to add marks : 44");

Student student = session.get(Student.class, 45); // Assuming 123 is the regNo of the student you want to update

if (student != null) {

// Update marks field

student.setMarks(95);

session.update(student); // Update the object in the database

System.out.println("Student marks updated successfully.");

} else {

System.out.println("Student not found!");

}

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

} finally {

if (session != null) {

session.close();

}

if (sessionFactory != null) {

sessionFactory.close();

}

}

}

public static void fetchObject(org.hibernate.Session session,SessionFactory sessionFactory, Transaction transaction) {

try {

session = sessionFactory.openSession();

transaction = session.beginTransaction();

// Fetching object by primary key (id)

Student student = session.get(Student.class, 45);

if (student != null) {

System.out.println("Fetched Student: " + ", Reg No: " + student.getRegno() + student.getName() + ", Marks: " + student.getMarks() + ", Email: " + student.getEmail() );

} else {

System.out.println("Student not found!");

}

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

} finally {

if (session != null) {

session.close();

}

if (sessionFactory != null) {

sessionFactory.close();

}

}

}

public static void deleteObject(org.hibernate.Session session,SessionFactory sessionFactory, Transaction transaction) {

try {

session = sessionFactory.openSession();

transaction = session.beginTransaction();

Student student = session.get(Student.class, 45);

if (student != null) {

session.delete(student);

System.out.println("Student with id 44 deleted successfully.");

} else {

System.out.println("Student not found, deletion failed!");

}

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

} finally {

if (session != null) {

session.close();

}

if (sessionFactory != null) {

sessionFactory.close();

}

}

}

public static void main(String[] args) {

// TODO Auto-generated method stub

Configuration cfg = null;

SessionFactory sessionFactory = null;

Student sd = null;

org.hibernate.Session session = null;

Transaction tx = null;

cfg = new Configuration();

cfg.configure("com/chinmay/cfgs/hibernate.cfg.xml");

sessionFactory = cfg.buildSessionFactory();

// addObject(sd); // performed this before

// updating new marks

// fetching data for that particular student

fetchObject(session, sessionFactory, tx);

// // updating marks

//

addObject(session, sessionFactory, tx);

updateMarks(session, sessionFactory, tx);

//

//

// // fetching data for that particular student to check update

//

fetchObject(session, sessionFactory, tx);

// // deleting object finally

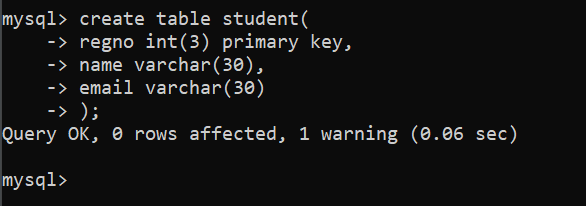
//

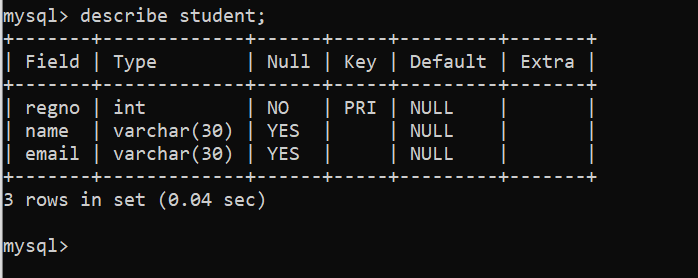
deleteObject(session, sessionFactory, tx);

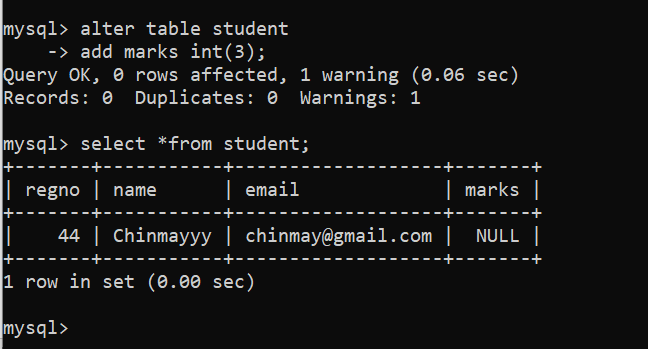
} }

**Outputs**

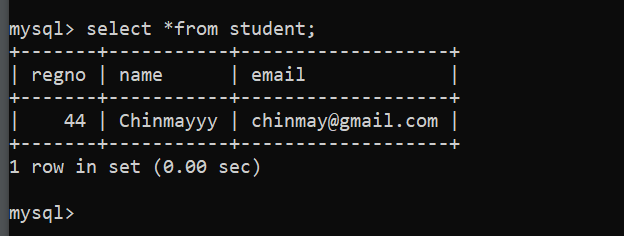
**(i) Create a student class with the following members regno,name and EmailId. Save the student object in the database using hibernate.**



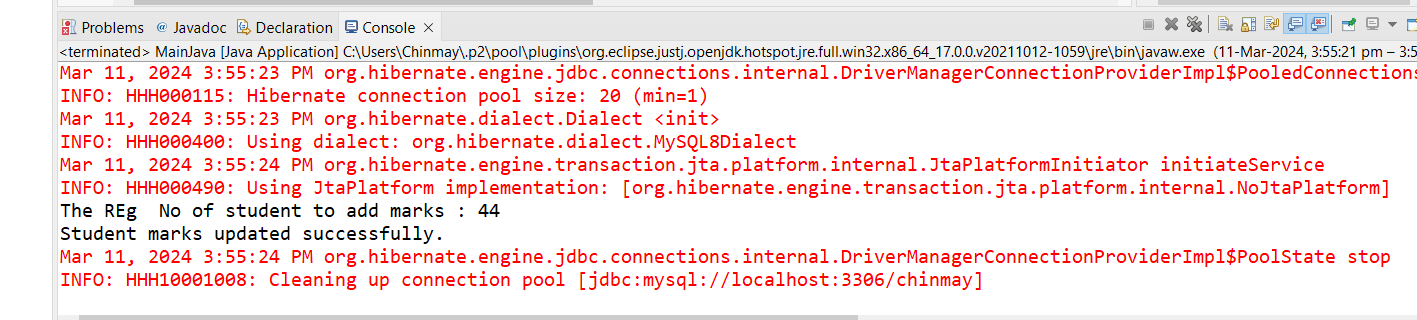




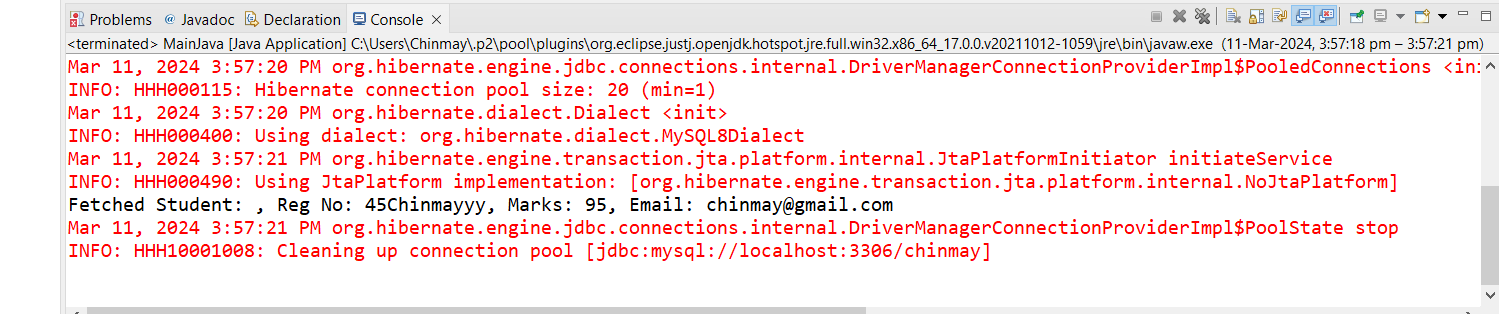
**(ii)Alter student class with marks data member.**



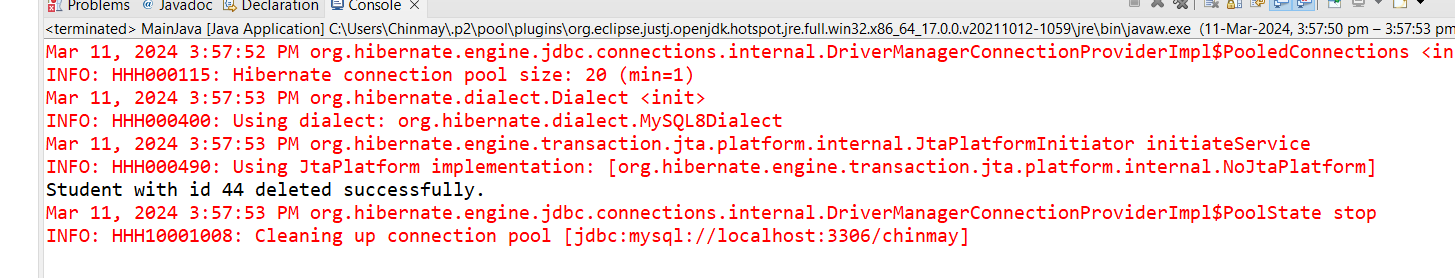
**(iii) Update mark data**



**(iii) Fetch the object from the table.**



**(iv) delete the object.**



**Task 2 : Along with student class, create mark class with three subject marks and perform CRUD operations using HQL.**