**Pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.cdac</groupId>

<artifactId>Simple-Hibernate-Maven-Project</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Simple-Hibernate-Maven-Project</name>

<description>Simple Hibernate Maven Project</description>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.32</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>6.1.7.Final</version>

</dependency>

</dependencies>

<build>

<sourceDirectory>src/main/java</sourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.5.1</version>

<configuration>

<source>17</source>

<target>17</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

**Student.java**

package com.cdac.model;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

*@Entity*

*@Table*(name = "table\_student1")

public class student {

*@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;

public Student() {

}

public Student(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;

}

*@Override*

public String toString() {

return "Student [id=" + id + ", firstName=" + firstName + ", lastName=" + lastName + ", email=" + email + "]";

}

}

**StudentDao.java**

package com.cdac.dao;

import java.util.List;

import org.hibernate.Session;

import org.hibernate.Transaction;

import com.cdac.model.Student;

import com.cdac.util.HibernateUtil;

public class StudentDao {

public void saveStudent(Student student) {

Transaction transaction = null;

try (Session session = HibernateUtil.*getSessionFactory*().openSession()) {

// start a transaction

transaction = session.beginTransaction();

// save the student object

session.~~save~~(student);

// commit transaction

transaction.commit();

} catch (Exception e) {

if (transaction != null) {

transaction.rollback();

}

e.printStackTrace();

}

}

public List<Student> getStudents() {

try (Session session = HibernateUtil.*getSessionFactory*().openSession()) {

return session.createQuery("from Student", Student.class).list();

}

}

**HibernateUtil.java**

package com.cdac.util;

import java.util.Properties;

import org.hibernate.SessionFactory;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

import org.hibernate.cfg.Configuration;

import org.hibernate.cfg.Environment;

import org.hibernate.service.ServiceRegistry;

import com.cdac.model.Student;

public class HibernateUtil {

private static SessionFactory *sessionFactory*;

public static SessionFactory getSessionFactory() {

if (*sessionFactory* == null) {

try {

Configuration configuration = new Configuration();

// Hibernate settings equivalent to hibernate.cfg.xml's properties

Properties settings = new Properties();

settings.put(Environment.***DRIVER***, "com.mysql.cj.jdbc.Driver");

settings.put(Environment.***URL***, "jdbc:mysql://localhost:3306/pune?useSSL=false");

settings.put(Environment.***USER***, "root");

settings.put(Environment.***PASS***, "cdac");

settings.put(Environment.***SHOW\_SQL***, "true");

settings.put(Environment.***CURRENT\_SESSION\_CONTEXT\_CLASS***, "thread");

settings.put(Environment.***HBM2DDL\_AUTO***, "create-drop");

configuration.setProperties(settings);

configuration.addAnnotatedClass(Student.class);

ServiceRegistry serviceRegistry = new StandardServiceRegistryBuilder()

.applySettings(configuration.getProperties()).build();

*sessionFactory* = configuration.buildSessionFactory(serviceRegistry);

} catch (Exception e) {

e.printStackTrace();

}

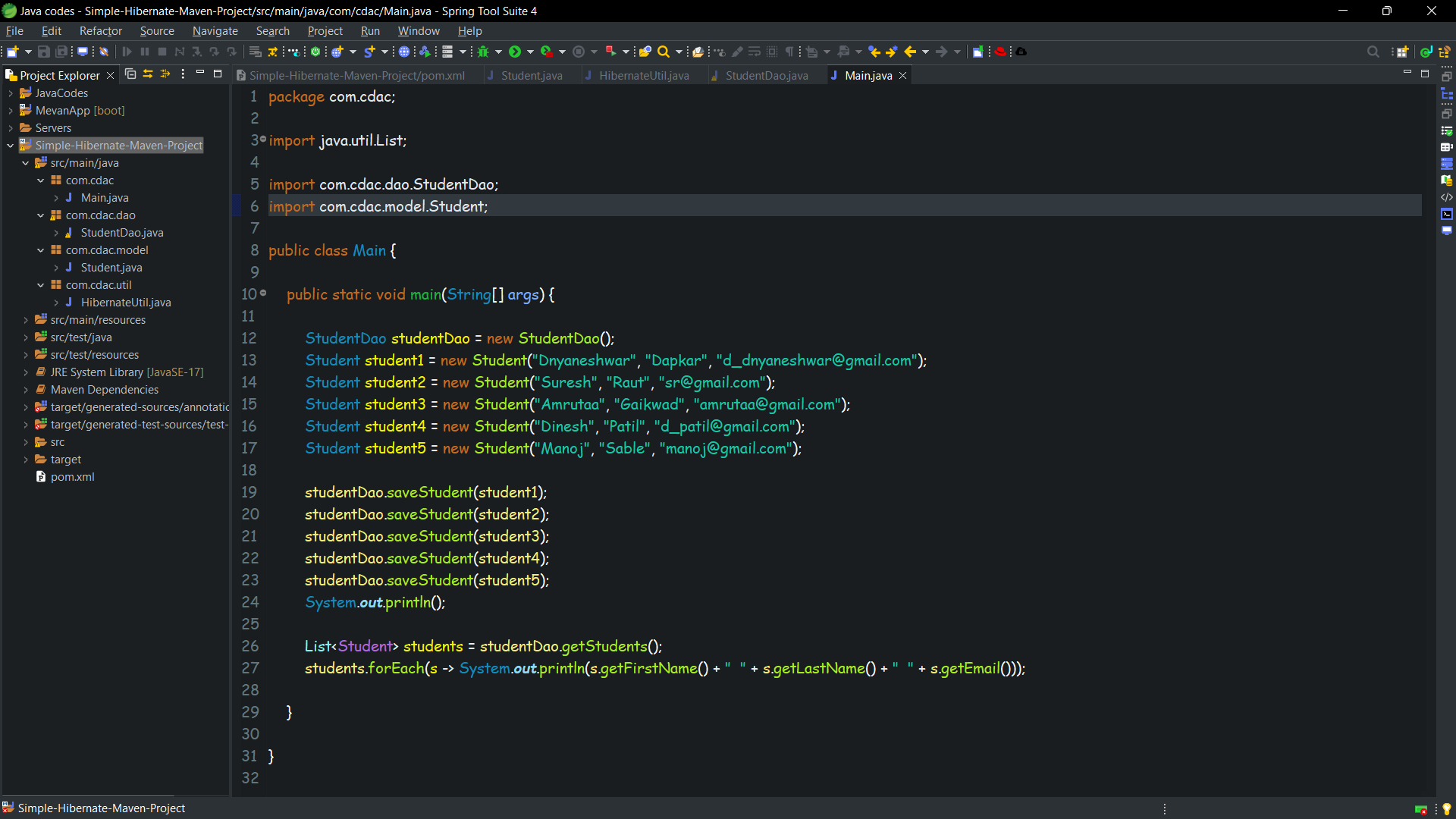
}

return *sessionFactory*;

}

}

**Project Structure**



**Database Screenshots**

