**Exercise 3 -Hands on 3 -Hibernate Annotation Config implementation walk through**

**CODE:**

**Employee Class: -**

package com.cognizant.handson3.model;  
import jakarta.persistence.\*;  
  
@Entity  
@Table(name = "employee")  
public class Employee {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name = "id")  
 private int id;  
  
 @Column(name = "name")  
 private String name;  
  
 @Column(name = "salary")  
 private double salary;  
  
 public int getId() {  
 return id;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public double getSalary() {  
 return salary;  
 }  
  
 public void setSalary(double salary) {  
 this.salary = salary;  
 }  
  
 @Override  
 public String toString() {  
 return "Employee [id=" + id + ", name=" + name + ", salary=" + salary + "]";  
 }  
}

**hibernate.cfg.xml file: -**

<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
 "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">  
  
<hibernate-configuration>  
 <session-factory>  
 <property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>  
 <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/ormlearn</property>  
 <property name="hibernate.connection.username">root</property>  
 <property name="hibernate.connection.password">Sreddy@11</property>  
  
 <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>  
  
 <property name="show\_sql">true</property>  
  
 <property name="hbm2ddl.auto">update</property>  
 <mapping class="com.cognizant.handson3.model.Employee"/>  
 </session-factory>  
</hibernate-configuration>

**HibernateUtil Class: -**

package com.cognizant.handson3.util;  
import org.hibernate.SessionFactory;  
import org.hibernate.cfg.Configuration;  
  
public class HibernateUtil {  
 private static SessionFactory *sessionFactory*;  
  
 static {  
 try {  
 *sessionFactory* = new Configuration().configure().buildSessionFactory();  
 } catch (Throwable ex) {  
 throw new ExceptionInInitializerError("Initial SessionFactory creation failed: " + ex);  
 }  
 }  
  
 public static SessionFactory getSessionFactory() {  
 return *sessionFactory*;  
 }  
  
 public static void shutdown() {  
 *getSessionFactory*().close();  
 }  
}

**OrmLearnApplication Class(Main class): -**

package com.cognizant.handson3;  
  
import com.cognizant. ormLearnApplication.model.Employee;  
import com.cognizant. ormLearnApplication.util.HibernateUtil;  
import org.hibernate.Session;  
import org.hibernate.Transaction;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class OrmLearnApplication {  
  
 public static void main(String[] args) {  
 Session session = HibernateUtil.*getSessionFactory*().openSession();  
 Transaction tx = null;  
  
 try {  
 tx = session.beginTransaction();  
  
 Employee emp = new Employee();  
 emp.setName("Sushma");  
 emp.setSalary(55000);  
 session.save(emp);  
  
 Employee fetched = session.get(Employee.class, emp.getId());  
 System.*out*.println("Fetched: " + fetched);  
  
 fetched.setSalary(60000);  
 session.update(fetched);  
  
 session.delete(fetched);  
  
 tx.commit();  
 } catch (Exception e) {  
 if (tx != null) tx.rollback();  
 e.printStackTrace();  
 } finally {  
 session.close();  
 }  
  
 HibernateUtil.*shutdown*();  
 }  
  
}

**OUTPUT:**

