```java
package model;

import jakarta.persistence.*;

@Entity
@Table(name="USER")
public class User {
    @Column(name="USER_ID")
    @Id
    @GeneratedValue(strategy = GenerationType.
IDENTITY)
    private Integer id;
    private String fullName;
    private String email;
    private String password;
    private int age;
    private double salary;
    private String city;

    public User(String fullName, String email, String
 password, int age, double salary, String city) {
        this.fullName = fullName;
        this.email = email;
        this.password = password;
        this.age = age;
        this.salary = salary;
        this.city = city;
    }

    public User(){

    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getFullName() {
        return fullName;
    }

    public void setFullName(String fullName) {
        this.fullName = fullName;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    public String getCity() {
        return city;
    }

    public void setCity(String city) {
        this.city = city;
    }
}

```

```java
package controller;

import jakarta.persistence.TypedQuery;
import model.User;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;

import java.util.List;

public class UserController {
    public static void main(String[] args) {
        SessionFactory factory = new Configuration().configure().buildSessionFactory();
        Session session = factory.openSession();
//        addUser(factory,session);
//        findUser(factory,session,3);
//
//        updateUser(session,3);
//
//        deleteUser(session,4);

//        findUserHql(factory, session);
//        getRecordbyId(factory, session);

//        getRecords(session);
//        getMaxSalary(session);
        namedQueryExample(session);
        factory.close();


        session.close();
    }
    public static void namedQueryExample(Session session){
        String hql = "FROM User u WHERE u.id = :id";
        TypedQuery<User> query = session.createQuery(hql, User.class);
        query.setParameter("id", 2);
        List<User> result = query.getResultList();

        System.out.printf("%s%13s%17s%34s%21s%n", "|User Id", "|Full name", "|Email", "|Password", "|Salary");
        for(User u: result){
            System.out.printf(" %-10d %-20s %-30s %-23s %s %n", u.getId(), u.getFullName(), u.getEmail(), u.getPassword(), u.getSalary());
        }
    }
    public static void findUserHql(SessionFactory factory, Session session){
        String hqlFrom = "FROM User";
        String hqlSelect = "SELECT u FROM User u";
        TypedQuery<User> query = session.createQuery(hqlFrom, User.class);
        List<User> results = query.getResultList();

        System.out.printf("%s%13s%17s%34s%n","|User Id","|Full name","|Email","|Password");

        for(User u: results){
            System.out.printf(" %-10d %-20s %-30s %s %n", u.getId(), u.getFullName(), u.getEmail(), u.getPassword());
        }
    }

    public static void addUser(SessionFactory factory, Session session) {
        Transaction transaction = session.beginTransaction();
        User uOne = new User();
        uOne.setEmail("haseeb@gmail.com");
        uOne.setFullName("Moh Haseeb");
        uOne.setPassword("has123");
        uOne.setSalary(2000.69);
        uOne.setAge(20);
        uOne.setCity("NYC");

        User uTwo = new User();
        uTwo.setEmail("James@gmail.com");
        uTwo.setFullName("James Santana");
        uTwo.setPassword("James123");
        uTwo.setSalary(2060.69);
        uTwo.setAge(25);
        uTwo.setCity("Dallas");

        User uThree = new User();
        uThree.setEmail("Shahparan@gmail.com");
        uThree.setFullName("AH Shahparan");
        uThree.setPassword("Shahparan123");
        uThree.setSalary(3060.69);
        uThree.setAge(30);
        uThree.setCity("Chicago");

        User uFour = new User("Christ", "christ@gmail.com", "147852", 35, 35000.3, "NJ");
        User uFive = new User("Sid", "Sid", "s258", 29, 4000.36, "LA");

        session.persist(uOne);
        session.persist(uTwo);
        session.persist(uThree);
        session.persist(uFour);
        session.persist(uFive);

        transaction.commit();
        System.out.println("successfully saved");
        factory.close();
        session.close();


    }

    public static void findUser(SessionFactory factory, Session session, int userId) {
        Transaction tx = session.beginTransaction();

        User u = session.get(User.class, userId);
        System.out.println("FullName: " + u.getFullName());
        System.out.println("Email: " + u.getEmail());
        System.out.println("password: " + u.getPassword());
        tx.commit();
        factory.close();
        session.close();
    }

    public static void updateUser(Session session, int userId) {
        // Todo comment out findUser method and uncomment updateUser method

        Transaction tx = session.beginTransaction();
        User u = new User();
        u.setId(userId);
        u.setEmail("mhaseeb@perscholas");
        u.setFullName("M Haseeb");
        u.setPassword("123456");
        session.merge(u);
        session.getTransaction().commit();
        session.close();
    }

    public static void deleteUser(Session session, int userId) {
        // Todo comment out updateUser method and uncomment deleteUser method

        SessionFactory factory = new Configuration().configure().buildSessionFactory();
        Transaction tx = session.beginTransaction();
        User u = new User();
        u.setId(userId);
        session.remove(u);
        tx.commit();
        session.close();
        factory.close();
    }

```

```java
//    public static void findUserHql(SessionFactory factory, Session session) {
//        String hqlForm = "FROM User";
//        String hqlSelect = "SELECT u FROM User u";
//        TypedQuery<User> query = session.createQuery(hqlSelect, User.class);
//        List<User> results = query.getResultList();
//        System.out.printf("%s%13s%17s%34s%n", "|User Id", "|Full name", "|Email", "|Password");
//        for (User u : results) {
//            System.out.printf(" %-10d %-20s %-30s %s %n", u.getId(), u.getFullName(), u.getEmail(), u.getPassword());
//        }
//    }

    public static void getRecordbyId(SessionFactory factory, Session session) {
        String hql = "FROM User u WHERE u.id > 2 ORDER BY u.salary DESC";
        TypedQuery<User> query = session.createQuery(hql, User.class);
        List<User> results = query.getResultList();
        System.out.printf("%s%13s%17s%34s%21s%n", "|User Id", "|Full name", "|Email", "|Password", "|Salary");
        for (User u : results) {
            System.out.printf(" %-10d %-20s %-30s %-23s %s %n", u.getId(), u.getFullName(), u.getEmail(),
                    u.getPassword(), u.getSalary());
        }
    }

    public static void getRecords(Session session) {
        TypedQuery<Object[]> query = session.createQuery("SELECT U.salary, U.fullName FROM User AS U", Object[].class);
        List<Object[]> results = query.getResultList();
        System.out.printf("%s%13s%n","Salary","City");
        for(Object[] a: results){
            System.out.printf("%-16s%s%n",a[0],a[1]);
        }
    }

    public static void getMaxSalary(Session session){
        String hql = "SELECT max(U.salary) FROM User U";
        TypedQuery<Object> query = session.createQuery(hql, Object.class);
        Object result = query.getSingleResult();
        System.out.printf("%s%s","Maximum Salary:",result);
    }


}

```

```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>
        <!-- Drop and re-create the database on startup -->
        <property name="hibernate.hbm2ddl.auto">update </property>

        <!-- Database connection settings -->
        <property name="connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property name="connection.url">jdbc:mysql://localhost:3306/usersDb?createDatabaseIfNotExist=true</property>
        <property name="connection.username">root</property>
        <property name="connection.password">SzhengSQL123</property>

        <!-- MySQL DB dialect -->
        <property name="dialect">org.hibernate.dialect.MySQLDialect</property>
        <!-- print all executed SQL on console -->
        <property name="hibernate.show_sql" >true </property>
        <property name="hibernate.format_sql" >true </property>

        <!--   Mapping entity file -->
        <mapping class="model.User"/>

    </session-factory>
</hibernate-configuration>


```