

Assignment Question

■ Case Study: Citizen and Passport Management System ■ Objective: To implement a Hibernate-based Java application using One-to-One mapping between two entities: Citizen and Passport. Each Citizen must have exactly one Passport and vice versa. ■ Entities: 1. Citizen (id, name, passport) 2. Passport (id, passportNumber, citizen) ■ Features: • Save Citizen with Passport using cascade. • Retrieve Citizen and Passport. • Maintain referential integrity. ■ Stack: • Hibernate 6+ • MySQL • JPA 3.1+ • Maven

1. 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
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>CitizenPassportHibernate</artifactId>
  <version>1.0</version>

  <dependencies>
    <dependency>
      <groupId>org.hibernate.orm</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>6.4.0.Final</version>
    </dependency>
    <dependency>
      <groupId>com.mysql</groupId>
      <artifactId>mysql-connector-j</artifactId>
      <version>8.0.33</version>
    </dependency>
    <dependency>
      <groupId>jakarta.persistence</groupId>
      <artifactId>jakarta.persistence-api</artifactId>
      <version>3.1.0</version>
    </dependency>
  </dependencies>
</project>
```

2. hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
  "-//Hibernate/Hibernate Configuration DTD//EN"
  "http://hibernate.org/dtd/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/yourdb</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password">root</property>
    <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>
    <property name="hibernate.hbm2ddl.auto">update</property>
    <property name="hibernate.show_sql">true</property>
    <mapping class="com.example.entity.Citizen"/>
    <mapping class="com.example.entity.Passport"/>
  </session-factory>
</hibernate-configuration>
```

3. HibernateUtil.java

```
package com.example.util;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {
  private static final SessionFactory sessionFactory;
  static {
```

```

        try {
            sessionFactory = new Configuration()
                .configure("hibernate.cfg.xml")
                .buildSessionFactory();
        } catch (Throwable ex) {
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}

```

4. Passport.java

```

package com.example.entity;

import jakarta.persistence.*;

@Entity
public class Passport {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    private String passportNumber;

    @OneToOne(mappedBy = "passport")
    private Citizen citizen;

    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

    public String getPassportNumber() { return passportNumber; }
    public void setPassportNumber(String passportNumber) { this.passportNumber = passportNumber; }

    public Citizen getCitizen() { return citizen; }
    public void setCitizen(Citizen citizen) { this.citizen = citizen; }
}

```

5. Citizen.java

```

package com.example.entity;

import jakarta.persistence.*;

@Entity
public class Citizen {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int id;

    private String name;

    @OneToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "passport_id")
    private Passport passport;

    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 Passport getPassport() { return passport; }
    public void setPassport(Passport passport) { this.passport = passport; }
}

```

6. App.java

```
package com.example.app;

import com.example.entity.Citizen;
import com.example.entity.Passport;
import com.example.util.HibernateUtil;
import org.hibernate.Session;

public class App {
    public static void main(String[] args) {
        Passport passport = new Passport();
        passport.setPassportNumber("P123456789");

        Citizen citizen = new Citizen();
        citizen.setName("John Doe");
        citizen.setPassport(passport);

        Session session = HibernateUtil.getSessionFactory().openSession();
        session.beginTransaction();

        session.persist(citizen);

        session.getTransaction().commit();
        session.close();

        System.out.println("Citizen and Passport saved successfully.");
    }
}
```