Day 7 hibernate Assignment

pom.xml

xml

```
<?xml version="1.0" encoding="UTF-8"?>
<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-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
        <hibernate.version>6.2.6.Final</hibernate.version>
        <mysql.version>8.0.33</mysql.version>
    </properties>

    <dependencies>
        <!-- Hibernate ORM -->
        <dependency>
            <groupId>org.hibernate.orm</groupId>
            <artifactId>hibernate-core</artifactId>
```

```
        <version>${hibernate.version}</version>
    </dependency>

    <!-- MySQL Connector -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>${mysql.version}</version>
    </dependency>
```

(hibernate.cfg.xml)

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.cj.jdbc.Driver</property>
    <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/citizen_passport_db</prope
rty>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password">password</property>

    <!-- JDBC connection pool settings -->
    <property name="hibernate.connection.pool_size">5</property>
```

```
        <!-- SQL dialect -->
        <property name="hibernate.dialect">org.hibernate.dialect.MySQL8Dialect</property>

        <!-- Echo all executed SQL to stdout -->
        <property name="hibernate.show_sql">true</property>
        <property name="hibernate.format_sql">true</property>

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

        <!-- Entity classes -->
        <mapping class="com.example.entity.Citizen"/>
        <mapping class="com.example.entity.Passport"/>
    </session-factory>
</hibernate-configuration>
```

Entity Classes

Citizen.java

java

```
package com.example.entity;

import jakarta.persistence.*;

@Entity
@Table(name = "citizens")
public class Citizen {

    @Id
```

```
@GeneratedValue(strategy = GenerationType.IDENTITY)
private Long id;

@Column(nullable = false)
private String name;

@OneToOne(cascade = CascadeType.ALL, fetch = FetchType.LAZY)
@JoinColumn(name = "passport_id", unique = true)
private Passport passport;

// Constructors
public Citizen() {}

public Citizen(String name) {
    this.name = name;
}

// Getters and Setters
public Long getId() {
    return id;
}

public void setId(Long 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;
        passport.setCitizen(this); // Maintain bidirectional relationship
    }

    @Override
    public String toString() {
        return "Citizen{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", passport=" + (passport != null ? passport.getPassportNumber() : "null") +
                '}';
    }
}
```

Passport.java

java

```
package com.example.entity;
```

```
import jakarta.persistence.*;

@Entity
@Table(name = "passports")
public class Passport {

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

    @Column(name = "passport_number", nullable = false, unique = true)
    private String passportNumber;

    @OneToOne(mappedBy = "passport", fetch = FetchType.LAZY)
    private Citizen citizen;

    // Constructors
    public Passport() {}

    public Passport(String passportNumber) {
        this.passportNumber = passportNumber;
    }

    // Getters and Setters
    public Long getId() {
        return id;
    }
```

```
    public void setId(Long 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;
    }

    @Override
    public String toString() {
        return "Passport{" +
                "id=" + id +
                ", passportNumber='" + passportNumber + '\'' +
                '}';
    }
}
```

HibernateUtil.java

java

```
package com.example.util;

import org.hibernate.SessionFactory;
import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

public class HibernateUtil {
    private static StandardServiceRegistry registry;
    private static SessionFactory sessionFactory;

    public static SessionFactory getSessionFactory() {
        if (sessionFactory == null) {
            try {
                // Create registry
                registry = new StandardServiceRegistryBuilder()
                        .configure("hibernate.cfg.xml")
                        .build();

                // Create MetadataSources
                MetadataSources sources = new MetadataSources(registry);

                // Create Metadata
                Metadata metadata = sources.getMetadataBuilder().build();
```

```
            // Create SessionFactory
            sessionFactory = metadata.getSessionFactoryBuilder().build();

        } catch (Exception e) {
            e.printStackTrace();
            if (registry != null) {
                StandardServiceRegistryBuilder.destroy(registry);
            }
        }
    }
    return sessionFactory;
  }

  public static void shutdown() {
    if (registry != null) {
        StandardServiceRegistryBuilder.destroy(registry);
    }
  }
}
```

App.java

java

```
package com.example.app;

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

```
import org.hibernate.Transaction;

public class App {
    public static void main(String[] args) {
        try {
            // Create a new citizen with passport
            Citizen citizen = new Citizen("John Doe");
            Passport passport = new Passport("A12345678");
            citizen.setPassport(passport);

            // Save the citizen (passport will be saved automatically due to cascade)
            Session session = HibernateUtil.getSessionFactory().openSession();
            Transaction transaction = session.beginTransaction();
            session.persist(citizen);
            transaction.commit();
            session.close();

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

            // Retrieve the citizen with passport
            session = HibernateUtil.getSessionFactory().openSession();
            Citizen retrievedCitizen = session.get(Citizen.class, citizen.getId());
            System.out.println("Retrieved Citizen: " + retrievedCitizen);
            System.out.println("Associated Passport: " + retrievedCitizen.getPassport());
            session.close();

        } catch (Exception e) {
            e.printStackTrace();
```

```
        } finally {
            HibernateUtil.shutdown();
        }
    }
}
```