**Spring -data-JPA-HandsOn**

**Hands on 1**

**Spring Data JPA - Quick Example**

**Database:**

create database ormlearn;

use ormlearn;

create table country (

co\_code varchar(2) primary key,

co\_name varchar(50)

);

insert into country values ('IN', 'India');

insert into country values ('US', 'United States of America');

**application.properties**

logging.level.org.springframework=info

logging.level.com.cognizant=debug

logging.level.org.hibernate.SQL=trace

logging.level.org.hibernate.type.descriptor.sql=trace

# Console log format

logging.pattern.console=%d{dd-MM-yy} %d{HH:mm:ss.SSS} %-20.20thread %5p %-25.25logger{25} %25M %4L %m%n

# Database configuration

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn

spring.datasource.username=root

spring.datasource.password=root

# Hibernate config

spring.jpa.hibernate.ddl-auto=validate

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

**Country.java**

package com.cognizant.ormlearn.model;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "country")

public class Country {

@Id

@Column(name = "co\_code")

private String code;

@Column(name = "co\_name")

private String name;

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

**CountryRepository.java**

package com.cognizant.ormlearn.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

import com.cognizant.ormlearn.model.Country;

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

}

**CountryService.java**

package com.cognizant.ormlearn.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.repository.CountryRepository;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

}

**OrmLearnApplication.java**

package com.cognizant.ormlearn;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import com.cognizant.ormlearn.model.Country;

import com.cognizant.ormlearn.service.CountryService;

@SpringBootApplication

public class OrmLearnApplication {

private static CountryService countryService;

private static final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);

LOGGER.info("Inside main");

countryService = context.getBean(CountryService.class);

testGetAllCountries();

}

private static void testGetAllCountries() {

LOGGER.info("Start");

List<Country> countries = countryService.getAllCountries();

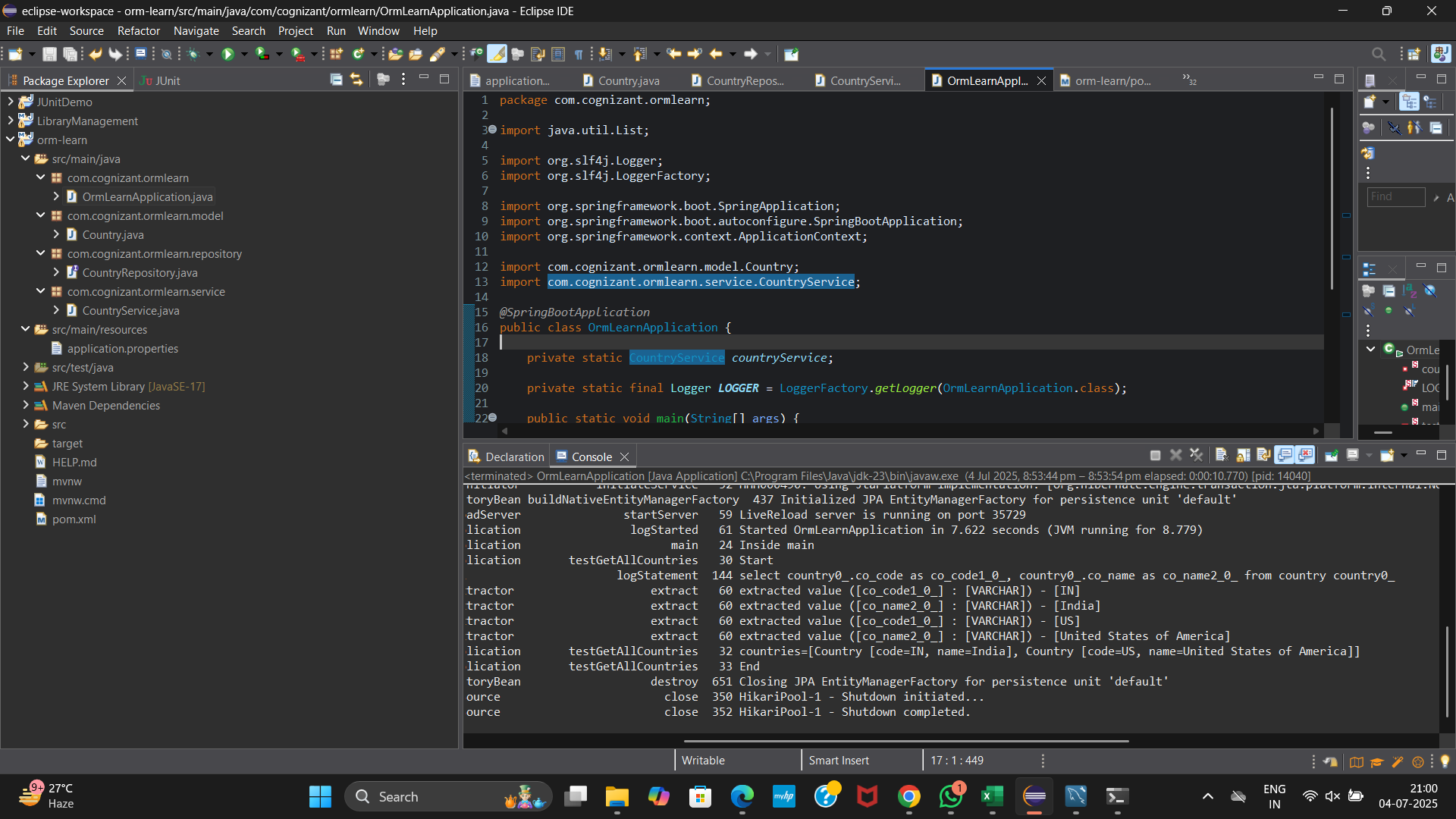
LOGGER.debug("countries={}", countries);

LOGGER.info("End");

}

}

**OUTPUT**



**Hands on 2**

**Hibernate XML Config implementation walk through**

**Pom.xml**

<dependencies>

<!-- Hibernate Core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.6.15.Final</version>

</dependency>

<!-- MySQL Connector -->

<dependency>

<groupId>mysql</groupId>

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

<version>8.0.33</version>

</dependency>

<!-- JUnit for testing (optional) -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

**Database**

CREATE DATABASE hibernatedb;

USE hibernatedb;

CREATE TABLE STUDENT (

ID INT PRIMARY KEY,

NAME VARCHAR(100),

AGE INT

);

**Student.java**

package com.example.model;

public class Student {

private int id;

private String name;

private int age;

// Getters & Setters

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 int getAge() { return age; }

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

@Override

public String toString() {

return "Student [ID=" + id + ", Name=" + name + ", Age=" + age + "]";

}

}

**Student.hbm.xml**

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

<!DOCTYPE hibernate-mapping PUBLIC

"-//Hibernate/Hibernate Mapping DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>

<class name="com.example.model.Student" table="STUDENT">

<id name="id" column="ID"/>

<property name="name" column="NAME"/>

<property name="age" column="AGE"/>

</class>

</hibernate-mapping>

**hibernate.cfg.xml**

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

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 3.0//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

<session-factory>

<!-- JDBC Database connection -->

<property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/hibernatedb</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password">root</property>

<!-- Hibernate properties -->

<property name="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</property>

<property name="hibernate.show\_sql">true</property>

<!-- Mapping file -->

<mapping resource="Student.hbm.xml"/>

</session-factory>

</hibernate-configuration>

**HibernateApp.java**

package com.example.main;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

import org.hibernate.cfg.Configuration;

import com.example.model.Student;

import java.util.List;

public class HibernateApp {

public static void main(String[] args) {

// Configure and build session factory

Configuration cfg = new Configuration();

cfg.configure("hibernate.cfg.xml");

SessionFactory factory = cfg.buildSessionFactory();

Session session = factory.openSession();

Transaction tx = session.beginTransaction();

// Insert data

Student student = new Student();

student.setId(1);

student.setName("Alice");

student.setAge(21);

session.save(student);

// Retrieve data

List<Student> list = session.createQuery("from Student").list();

for (Student s : list) {

System.out.println(s);

}

// Delete student

Student s1 = session.get(Student.class, 1);

session.delete(s1);

// Commit and close

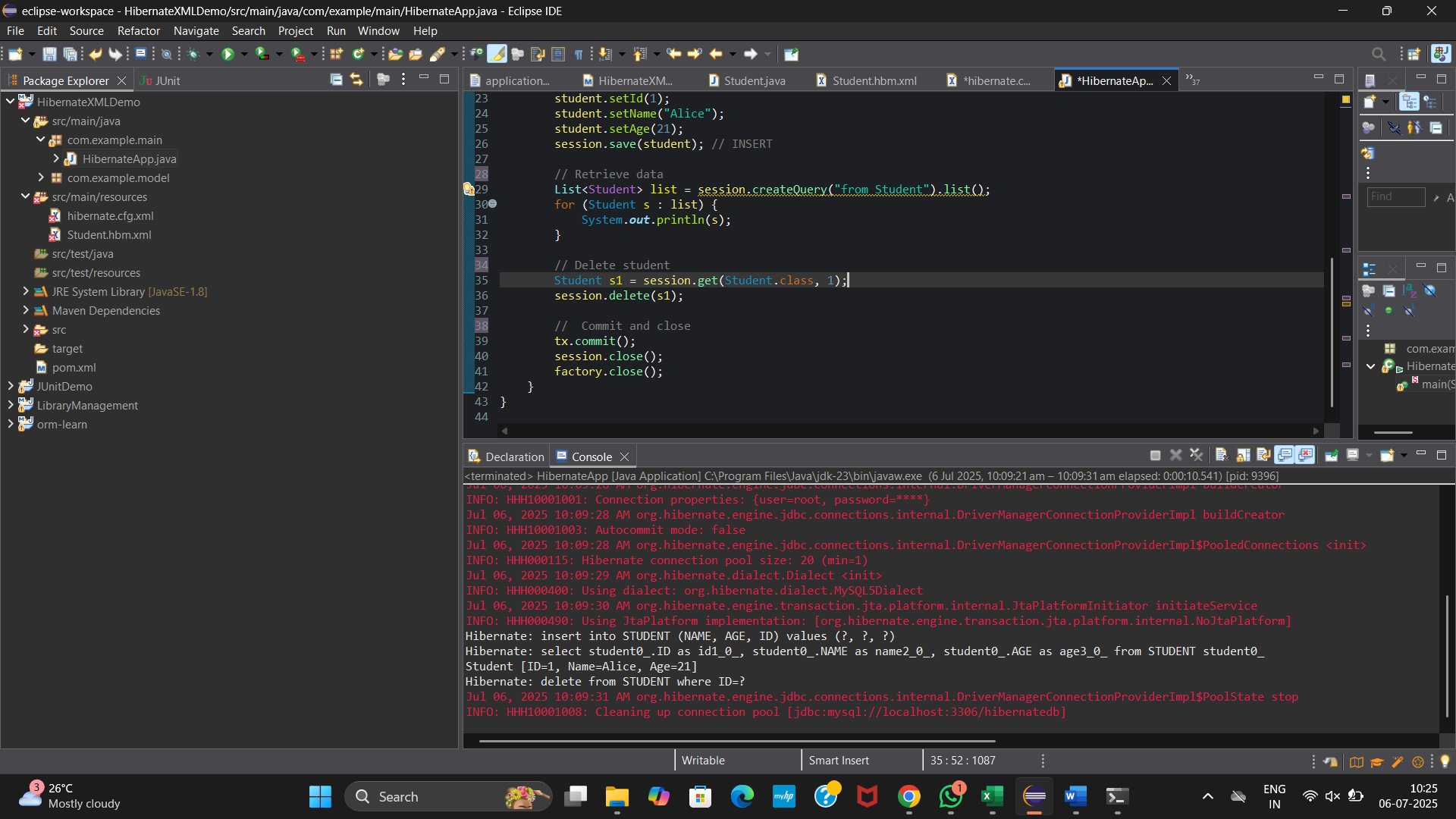
tx.commit();

session.close();

factory.close();

}

}



**Hands on 4**

**Difference between JPA, Hibernate and Spring Data JPA**

**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.example</groupId>

<artifactId>employee-app</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<!-- Hibernate Core -->

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.6.15.Final</version>

</dependency>

<!-- MySQL Connector -->

<dependency>

<groupId>mysql</groupId>

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

<version>8.0.33</version>

</dependency>

<!-- Spring Boot Starter Data JPA -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

<version>2.7.18</version>

</dependency>

<!-- Spring Boot Starter -->

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

<version>2.7.18</version>

</dependency>

</dependencies>

</project>

**Employee.java**

package com.example.model;

import javax.persistence.\*;

*@Entity*

*@Table*(name = "employee")

public class Employee {

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

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 + "]";

}

}

**HibernateUtil.java**

package com.example.util;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class HibernateUtil {

private static final SessionFactory ***sessionFactory*** =

new Configuration().configure("hibernate.cfg.xml").buildSessionFactory();

public static SessionFactory getSessionFactory() {

return ***sessionFactory***;

}

}

**hibernate.cfg.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>

<property name="hibernate.connection.driver\_class">com.mysql.cj.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/hibernatedb</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password">root</property>

<property name="hibernate.dialect">org.hibernate.dialect.MySQL5Dialect</property>

<property name="hibernate.hbm2ddl.auto">update</property>

<mapping class="com.example.model.Employee"/>

</session-factory>

</hibernate-configuration>

**HibernateExample.java**

package com.example.hibernate;

import com.example.model.Employee;

import com.example.util.HibernateUtil;

import org.hibernate.\*;

public class HibernateExample {

public static void main(String[] args) {

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

Transaction tx = null;

try {

tx = session.beginTransaction();

Employee emp = new Employee();

emp.setName("Alice");

emp.setSalary(30000);

session.save(emp);

tx.commit();

System.***out***.println("Hibernate Save Successful");

} catch (Exception e) {

if (tx != null) tx.rollback();

e.printStackTrace();

} finally {

session.close();

}

}

}

**EmployeeRepository.java**

package com.example.springdatajpa;

import com.example.model.Employee;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**EmployeeService.java**

package com.example.springdatajpa;

import com.example.model.Employee;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import javax.transaction.Transactional;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee emp) {

employeeRepository.save(emp);

}

}

**SpringDataJpaApp.java**

package com.example.springdatajpa;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

import org.springframework.boot.autoconfigure.domain.EntityScan;

import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import com.example.model.Employee;

import com.example.springdatajpa.EmployeeService;

@SpringBootApplication(scanBasePackages = "com.example")

@EntityScan(basePackages = "com.example.model")

@EnableJpaRepositories(basePackages = "com.example.springdatajpa")

public class SpringDataJpaApp {

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(SpringDataJpaApp.class, args);

EmployeeService service = context.getBean(EmployeeService.class);

Employee emp = new Employee();

emp.setName("Bob");

emp.setSalary(40000);

service.addEmployee(emp);

System.out.println("Spring Data JPA Save Successful");

}

}

**application.properties**

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/hibernatedb

spring.datasource.username=root

spring.datasource.password=root

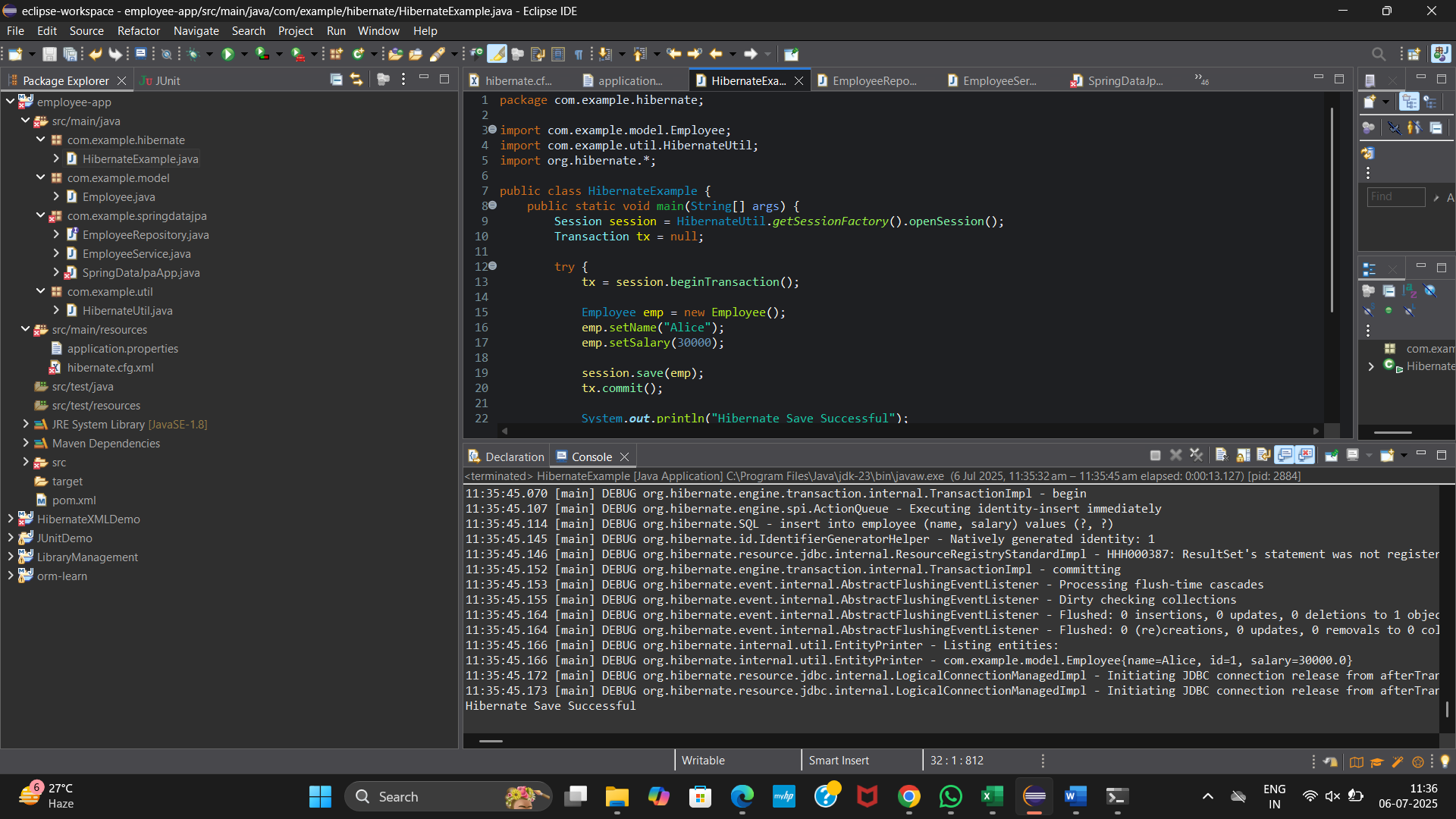
spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

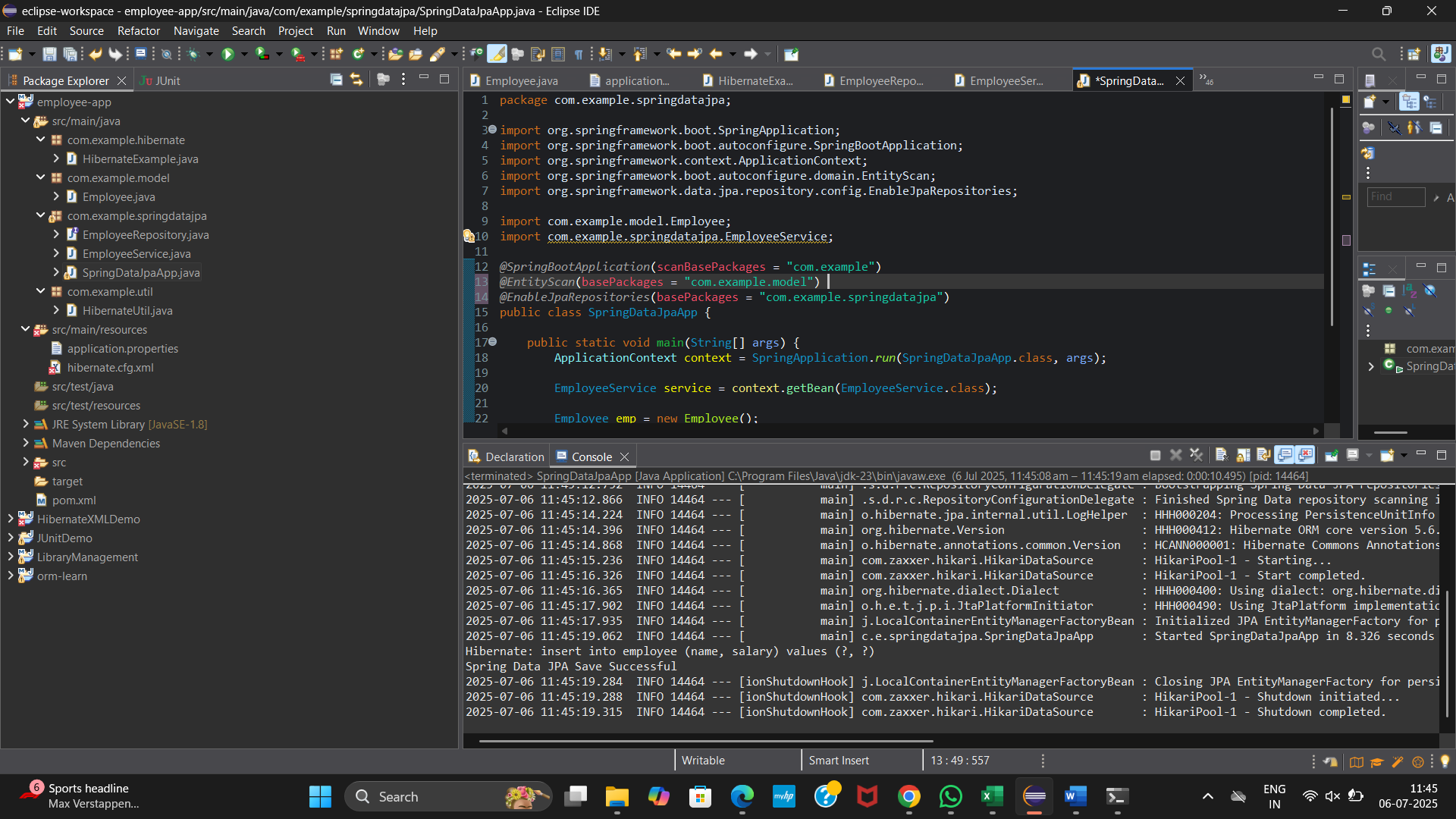
spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect

**OUTPUT**

**HIBERNATE**

****

**SPRING DATA JPA**

****

**Hands On : 5-9**

**Implement services for managing Country**

**pom.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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>3.5.3</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>com.cognizant.ormlearn</groupId>

<artifactId>country-service</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>country-service</name>

<description>Demo project for Spring Boot</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**Country.java**

package com.cognizant.ormlearn.country\_service.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

import jakarta.persistence.Column;

*@Entity*

*@Table*(name = "country")

public class Country {

*@Id*

*@Column*(name = "co\_code")

private String code;

*@Column*(name = "co\_name")

private String name;

public Country() {}

public Country(String code, String name) {

this.code = code;

this.name = name;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

*@Override*

public String toString() {

return "Country [code=" + code + ", name=" + name + "]";

}

}

**CountryRepository.java**

package com.cognizant.ormlearn.country\_service.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import com.cognizant.ormlearn.country\_service.model.Country;

public interface CountryRepository extends JpaRepository<Country, String> {

List<Country> findByNameContainingIgnoreCase(String keyword);

}

**CountryService.java**

package com.cognizant.ormlearn.country\_service.service;

import java.util.List;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.cognizant.ormlearn.country\_service.model.Country;

import com.cognizant.ormlearn.country\_service.repository.CountryRepository;

*@Service*

public class CountryService {

*@Autowired*

private CountryRepository countryRepository;

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

public Country findCountryByCode(String code) {

Optional<Country> result = countryRepository.findById(code);

return result.orElse(null);

}

public Country addCountry(Country country) {

return countryRepository.save(country);

}

public Country updateCountry(String code, String name) {

Optional<Country> result = countryRepository.findById(code);

if (result.isPresent()) {

Country country = result.get();

country.setName(name);

return countryRepository.save(country);

}

return null;

}

public void deleteCountry(String code) {

countryRepository.deleteById(code);

}

public List<Country> searchByName(String keyword) {

return countryRepository.findByNameContainingIgnoreCase(keyword);

}

}

**CountryServiceApplication.java**

package com.cognizant.ormlearn.country\_service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import com.cognizant.ormlearn.country\_service.model.Country;

import com.cognizant.ormlearn.country\_service.service.CountryService;

*@SpringBootApplication*

public class CountryServiceApplication implements CommandLineRunner {

*@Autowired*

private CountryService countryService;

public static void main(String[] args) {

SpringApplication.*run*(CountryServiceApplication.class, args);

}

*@Override*

public void run(String... args) throws Exception {

System.***out***.println("Getting all countries:");

List<Country> countries = countryService.getAllCountries();

countries.forEach(System.***out***::println);

System.***out***.println("\nAdding new country:");

Country c1 = new Country("XY", "Xyland");

countryService.addCountry(c1);

System.***out***.println("Added: " + c1);

System.***out***.println("\nUpdating country:");

countryService.updateCountry("XY", "Xylophone Land");

System.***out***.println("\nSearching by name:");

List<Country> searchResults = countryService.searchByName("land");

searchResults.forEach(System.***out***::println);

System.***out***.println("\nDeleting country:");

countryService.deleteCountry("XY");

}

}

**application.properties**

spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn

spring.datasource.username=root

spring.datasource.password=root

# JPA settings

spring.jpa.hibernate.ddl-auto=none

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

# Logging

logging.level.org.hibernate.SQL=DEBUG

logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE

**OUTPUT**

