**Exercise 1: Spring Data JPA - Quick Example**

**Application.properties:**

spring.application.name=country-demo

# Logging

logging.level.org.springframework=info

logging.level.com.example=debug

logging.level.org.hibernate.SQL=debug

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

# MySQL config

spring.datasource.url=jdbc:mysql://localhost:3306/country\_schema

spring.datasource.username=root

spring.datasource.password=root

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

# Hibernate config

spring.jpa.hibernate.ddl-auto=validate

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

**country info:**

package com.example.country\_demo.entity;

import jakarta.persistence.\*;

*@Entity*

*@Table*(name = "country\_info")

public class CountryInfo {

*@Id*

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

private String code;

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

private String name;

// Getters & Setters

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 "CountryInfo [code=" + code + ", name=" + name + "]";

}

}

**CountryDemoApplication:**

package com.example.country\_demo;

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.example.country\_demo.entity.CountryInfo;

import com.example.country\_demo.service.CountryInfoService;

@SpringBootApplication

public class CountryDemoApplication {

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

private static CountryInfoService countryService;

public static void main(String[] args) {

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

LOGGER.info("Application Started");

countryService = context.getBean(CountryInfoService.class);

testGetAllCountries();

}

private static void testGetAllCountries() {

LOGGER.info("Start testGetAllCountries");

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

LOGGER.debug("Countries: {}", countries);

LOGGER.info("End testGetAllCountries");

}

}

**CountryInfoRepository:**

package com.example.country\_demo.repository;

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

import org.springframework.stereotype.Repository;

import com.example.country\_demo.entity.CountryInfo;

@Repository

public interface CountryInfoRepository extends JpaRepository<CountryInfo, String> {

}

**CountryInfoService:**

package com.example.country\_demo.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.example.country\_demo.entity.CountryInfo;

import com.example.country\_demo.repository.CountryInfoRepository;

*@Service*

public class CountryInfoService {

*@Autowired*

private CountryInfoRepository countryRepo;

*@Transactional*

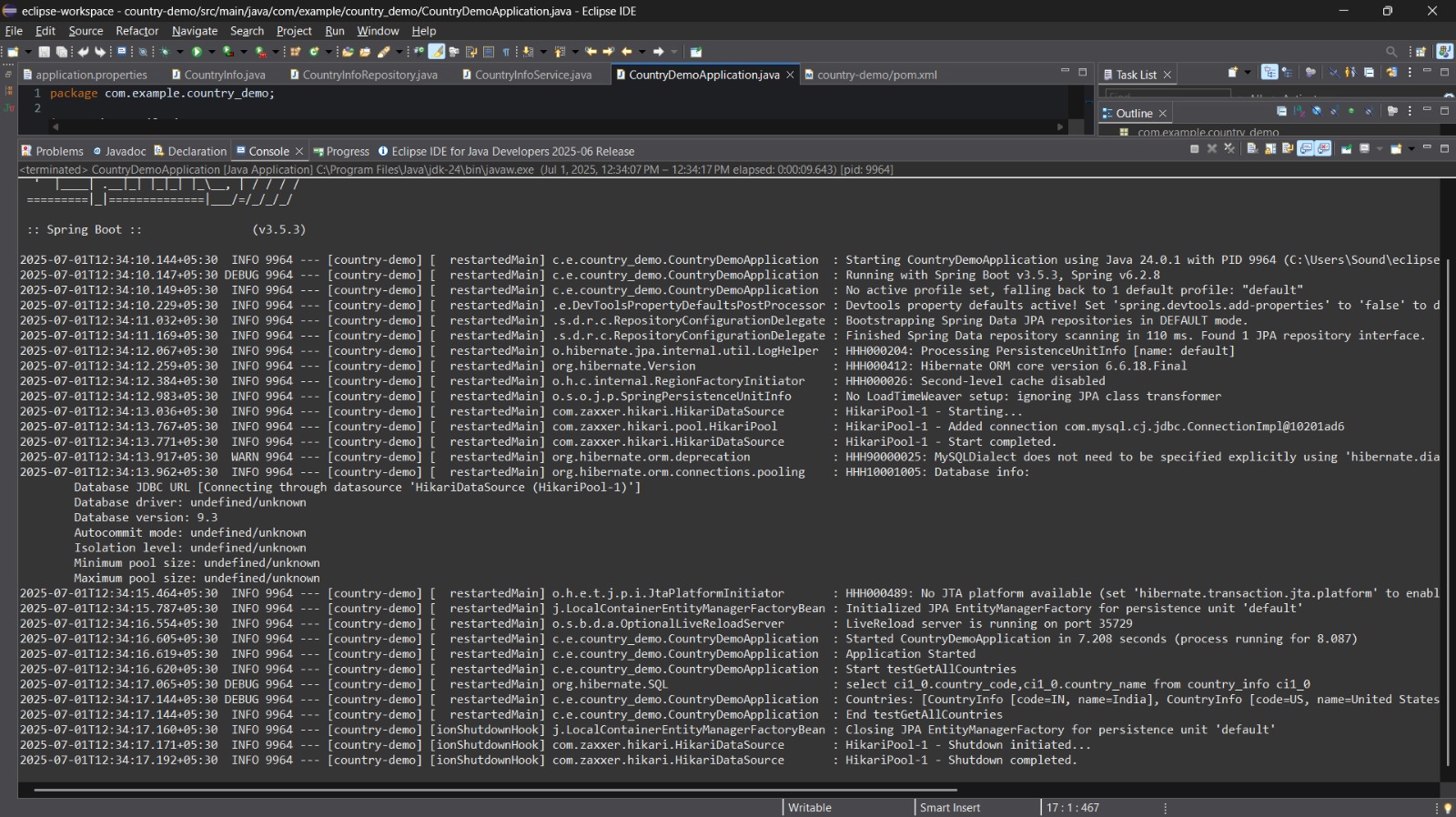
public List<CountryInfo> getAllCountries() {

return countryRepo.findAll();

}

}

**Output:**





**Exercise 2: Difference between JPA, Hibernate and Spring Data JPA**

**Hibernate**

**App.java:**

package com.example.hibernate;

public class App {

public static void main(String[] args) {

Employee emp = new Employee();

emp.setName("John");

emp.setDepartment("IT");

EmployeeService service = new EmployeeService();

Integer id = service.addEmployee(emp);

System.out.println("Employee ID: " + id);

}

}

**Employee.java:**

package com.example.hibernate;

import javax.persistence.\*;

@Entity

@Table(name = "employee")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String department;

// Getters and 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 String getDepartment() { return department; }

public void setDepartment(String department) { this.department = department; }

}

**EmployeeService.java:**

package com.example.hibernate;

import org.hibernate.\*;

public class EmployeeService {

public Integer addEmployee(Employee employee) {

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

Transaction tx = null;

Integer employeeID = null;

try {

tx = session.beginTransaction();

employeeID = (Integer) session.save(employee);

tx.commit();

} catch (HibernateException e) {

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

e.printStackTrace();

} finally {

session.close();

}

return employeeID;

}

}

**HibernateUtil.java:**

package com.example.hibernate;

import org.hibernate.SessionFactory;

import org.hibernate.cfg.Configuration;

public class HibernateUtil {

private static final SessionFactory sessionFactory;

static {

sessionFactory = new Configuration().configure().buildSessionFactory();

}

public static SessionFactory getSessionFactory() {

return sessionFactory;

}

}

**hibernate.clg.xml:**

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

<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/testdb</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="show\_sql">true</property>

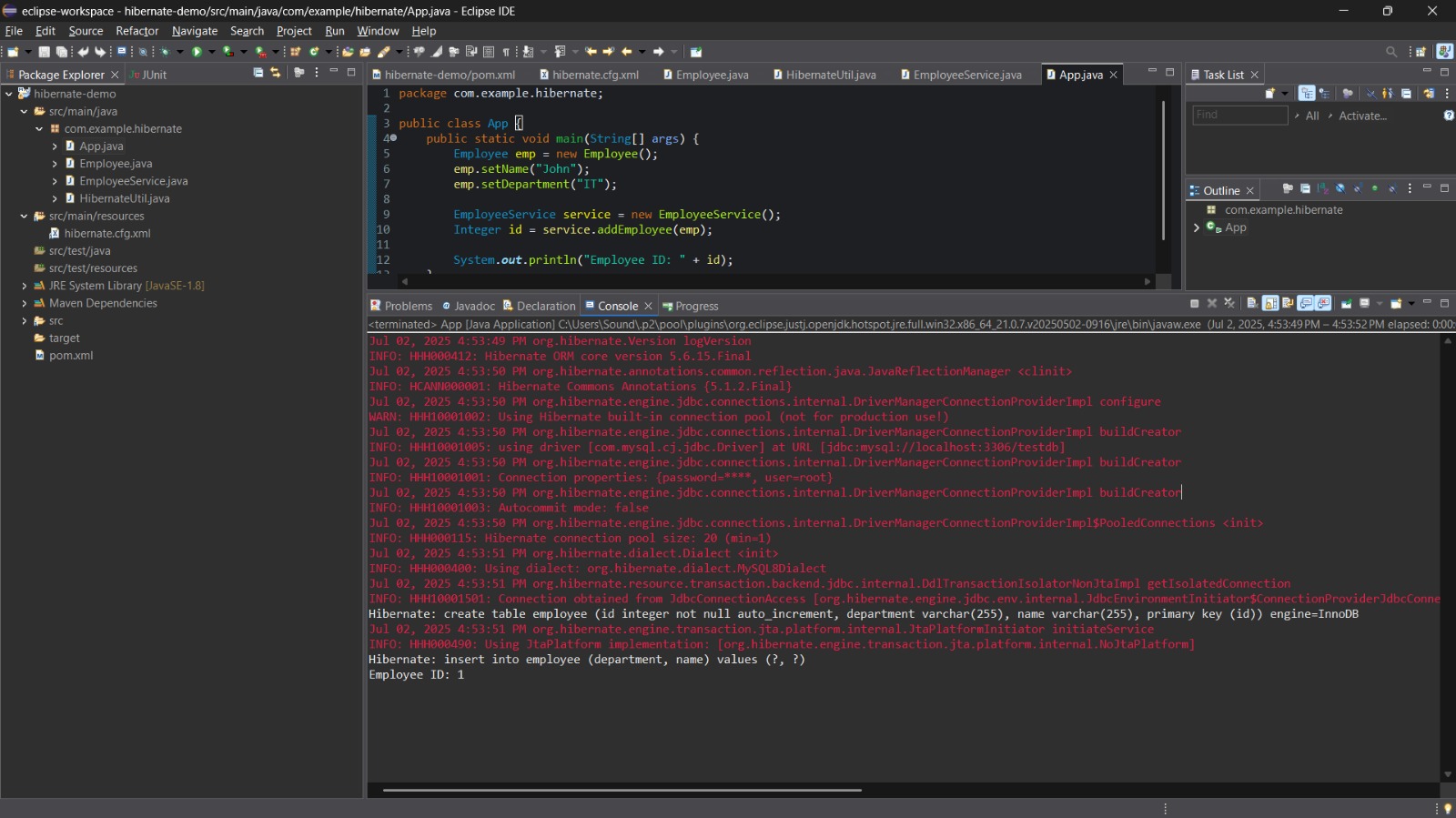
<property name="hbm2ddl.auto">update</property>

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

</session-factory>

</hibernate-configuration>

**Output:**





**Springdata JPA**

**application.properties:**

spring.application.name=springdata-demo

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

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

**Employee.java:**

package com.example.springdata\_demo;

import jakarta.persistence.\*;

@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Integer id;

private String name;

private String department;

// Getters and Setters

public Integer getId() { return id; }

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

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getDepartment() { return department; }

public void setDepartment(String department) { this.department = department; }

}

**EmployeeRepository.java:**

package com.example.springdata\_demo;

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

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**EmployeeService.java:**

package com.example.springdata\_demo;

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

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Transactional;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

}

**SpringdataDemoApplication.java:**

package com.example.springdata\_demo;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SpringdataDemoApplication implements CommandLineRunner {

@Autowired

private EmployeeService employeeService;

public static void main(String[] args) {

SpringApplication.run(SpringdataDemoApplication.class, args);

}

@Override

public void run(String... args) {

Employee emp = new Employee();

emp.setName("Alice");

emp.setDepartment("HR");

employeeService.addEmployee(emp);

System.out.println("Employee inserted successfully!");

}

}

**Output:**

