## WEEK 3 HANDS\_ON EXERCISES

### **Spring core and Maven**

**Exercise 1: Configuring a Basic Spring Application**

**SOURCE CODE:**

**BookRepository.java:**

package com.library.repository;

public class BookRepository {

public void saveBook(String book) {

System.out.println("Book \"" + book + "\" saved to the repository!");

}

}

**BookService.java:**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String book) {

System.out.println("Adding book: " + book);

bookRepository.saveBook(book);

}

}

**MainApp.java:**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args)

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = (BookService) context.getBean("bookService");

bookService.addBook("Spring Framework Guide");

}

}

**applicationContext.xml:**

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

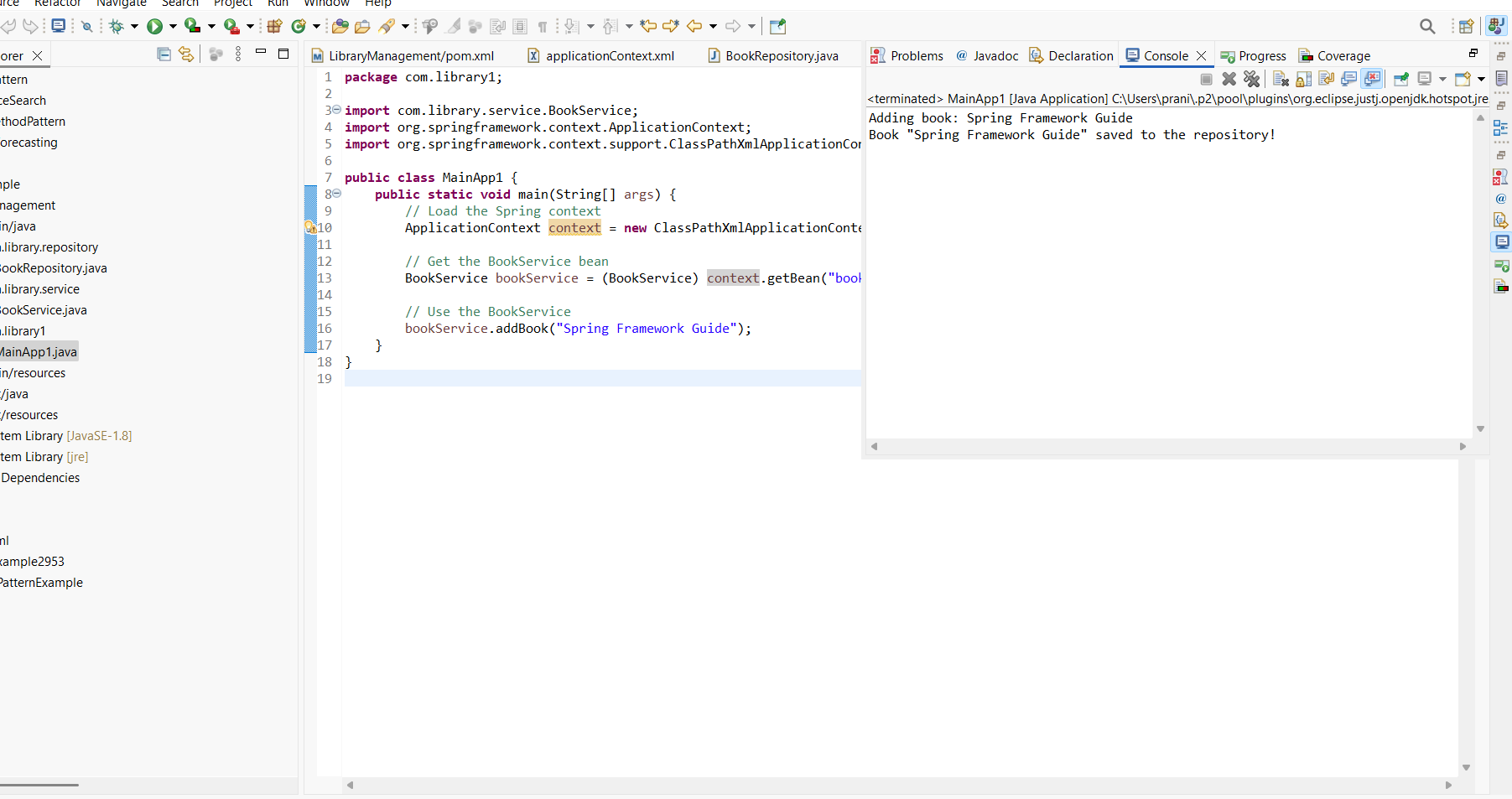
<bean id="bookRepository" class="com.library.repository.BookRepository" />

<bean id="bookService" class="com.library.service.BookService">

<property name="bookRepository" ref="bookRepository" />

</bean>

</beans>

**OUTPUT:**

**Exercise 2: Implementing Dependency Injection**

**SOURCE CODE:**

**BookRepository.java:**

package com.example.repository;

public class BookRepository {

public void getAllBooks() {

System.out.println("Fetching all books from repository...");

}

}

**BookService.java:**

package com.example.service;

import com.example.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void displayBooks() {

bookRepository.getAllBooks();

}

}

**LibraryMangaementApplication.java:**

package com.example.main;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService service = (BookService) context.getBean("bookService");

service.displayBooks();

}

}

**applicationContext.xml:**

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

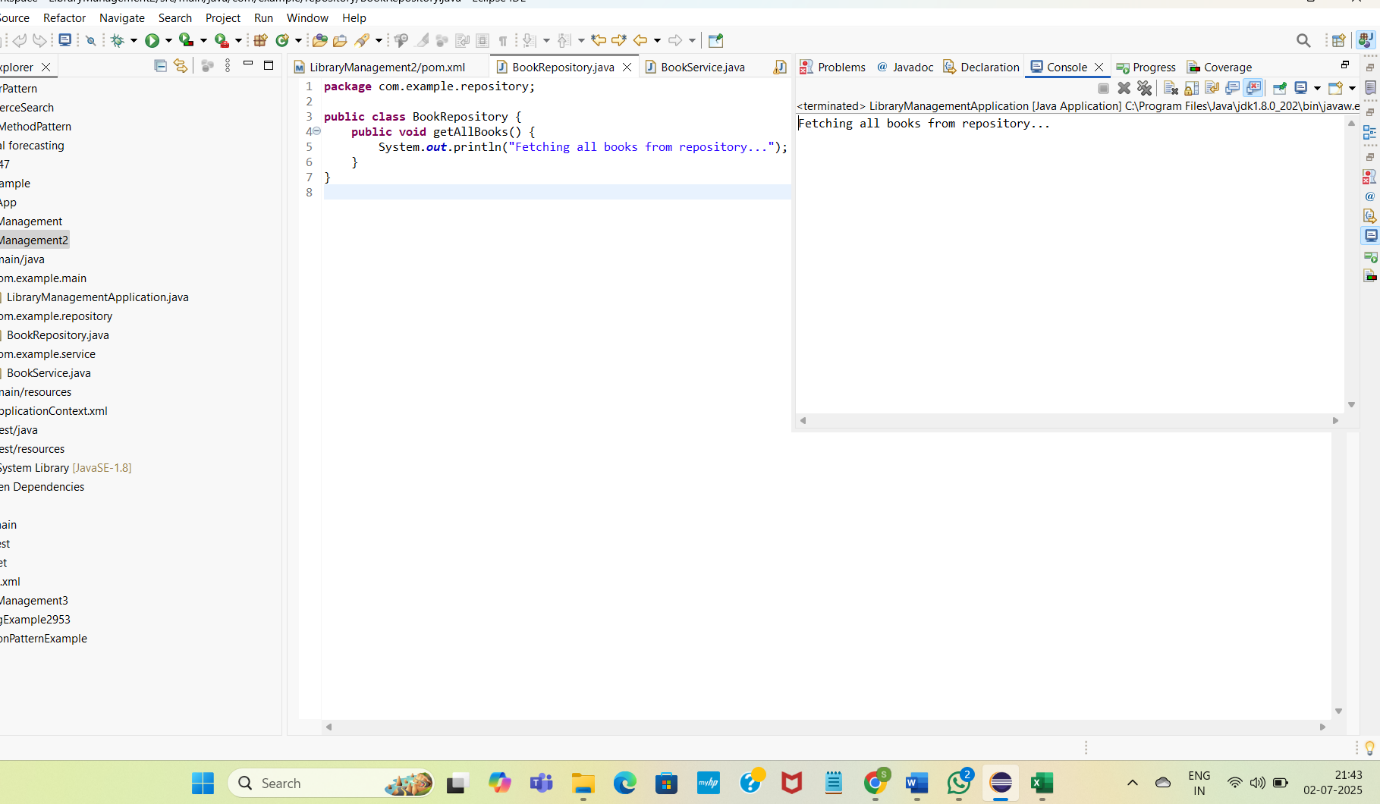
<bean id="bookRepository" class="com.example.repository.BookRepository" />

<bean id="bookService" class="com.example.service.BookService">

<property name="bookRepository" ref="bookRepository" />

</bean>

</beans>

**OUTPUT:**

**Exercise 4: Creating and Configuring a Maven Project**

**SOURCE CODE:**

**LibraryApp.java:**

package com.example.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryApp {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

LibraryService service = (LibraryService) context.getBean("libraryService");

service.getLibraryDetails();

}

}

**LibraryService.java:**

package com.example.library;

public class LibraryService {

public void getLibraryDetails() {

System.out.println("Welcome to GPREC Library!");

}

}

**applicationContext.xml:**

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

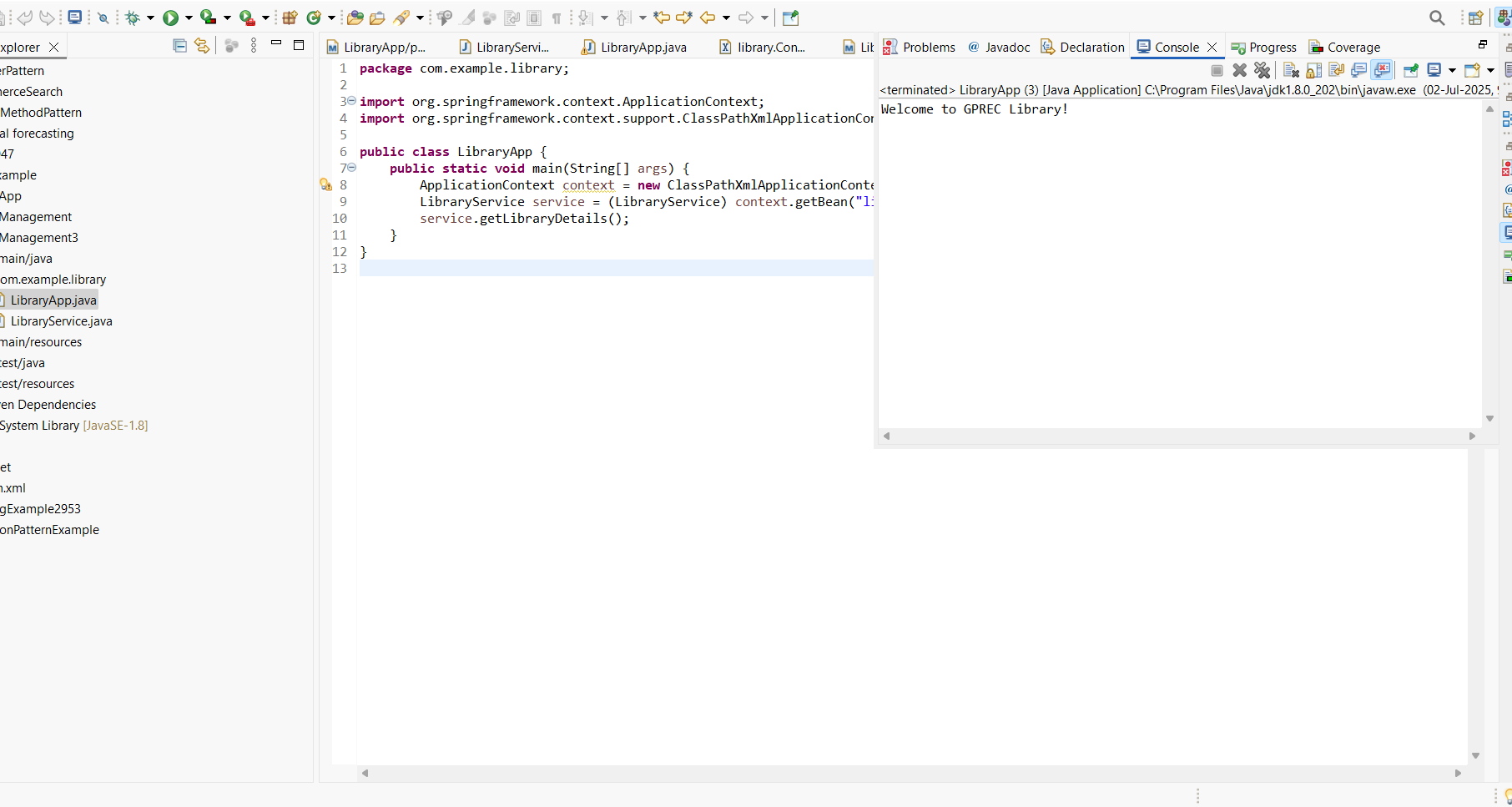
xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="libraryService" class="com.example.library.LibraryService"/>

</beans>

**Output:**

### **Spring Data JPA with Spring Boot, Hibernate Exercises**

**Spring Data JPA - Quick Example**

**SOURCE CODE:**

**OrmlearnApplication.java:**

**package** com.cognizant.orm\_learn;

**import** java.util.List;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

**import** org.springframework.context.ApplicationContext;

**import** com.cognizant.orm\_learn.model.Country;

**import** com.cognizant.orm\_learn.service.CountryService;

@SpringBootApplication

**public** **class** OrmLearnApplication {

**public** **static** **void** main(String[] args) {

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

CountryService countryService = context.getBean(CountryService.**class**);

System.***out***.println("Start...");

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

**for** (Country country : countries) {

System.***out***.println(country);

}

System.***out***.println("End...");

}

}

**Country.java:**

**package** com.cognizant.orm\_learn.model;

**import** jakarta.persistence.Column;

**import** jakarta.persistence.Entity;

**import** jakarta.persistence.Id;

**import** jakarta.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.orm\_learn.repository;

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

**import** org.springframework.stereotype.Repository;

**import** com.cognizant.orm\_learn.model.Country;

@Repository

**public** **interface** CountryRepository **extends** JpaRepository<Country, String> {

}

**CountryService.java:**

**package** com.cognizant.orm\_learn.service;

**import** java.util.List;

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

**import** org.springframework.stereotype.Service;

**import** com.cognizant.orm\_learn.model.Country;

**import** com.cognizant.orm\_learn.repository.CountryRepository;

@Service

**public** **class** CountryService {

@Autowired

**private** CountryRepository countryRepository;

**public** List<Country> getAllCountries() {

**return** countryRepository.findAll();

}

}

**application.properties:**

# Logging

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 connection

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

spring.datasource.username=root

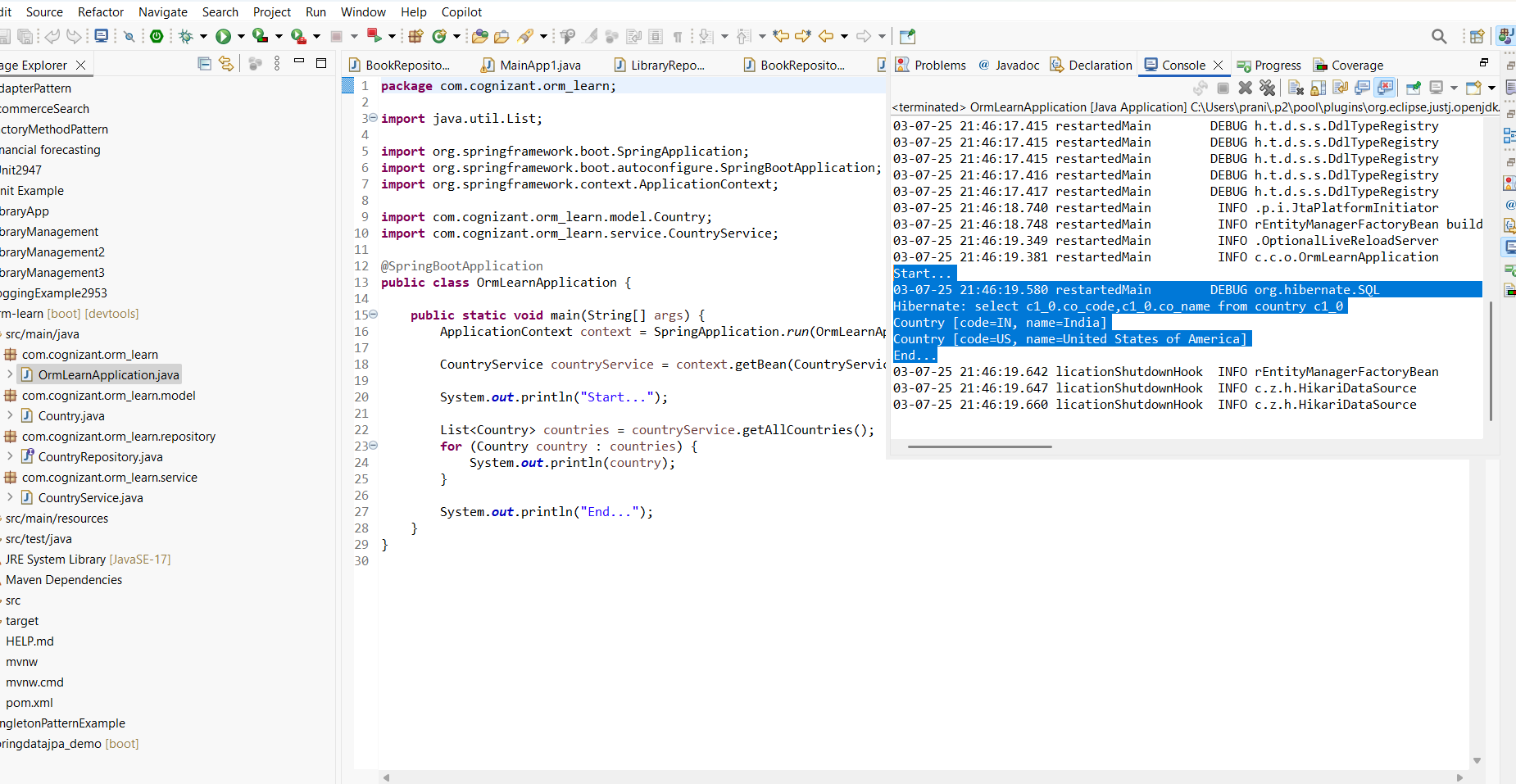
spring.datasource.password=Bunny###1

# Hibernate properties

spring.jpa.show-sql=true

spring.jpa.hibernate.ddl-auto=none

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

**OUTPUT:**

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

**SOURCECODE:**

**Employee.java:**

**package** com.example.springdatajpa\_demo;

**import** jakarta.persistence.\*;

@Entity

**public** **class** Employee {

@Id

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

**private** Integer id;

**private** String name;

**private** String department;

**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.springdatajpa\_demo;

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

**public** **interface** EmployeeRepository **extends** JpaRepository<Employee, Integer> {

}

**EmployeeService.java:**

**package** com.example.springdatajpa\_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);

}

}

**SpringdatajpaDemoApllication.java:**

**package** com.example.springdatajpa\_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** SpringdatajpaDemoApplication **implements** CommandLineRunner {

@Autowired

**private** EmployeeService employeeService;

**public** **static** **void** main(String[] args) {

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

}

@Override

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

Employee emp = **new** Employee();

emp.setName("Pranith");

emp.setDepartment("CSB");

employeeService.addEmployee(emp);

System.***out***.println("Employee added successfully.");

}

}

**application.properties:**

#Database Connection

spring.datasource.url=jdbc:mysql://localhost:3306/employee\_db

spring.datasource.username=root

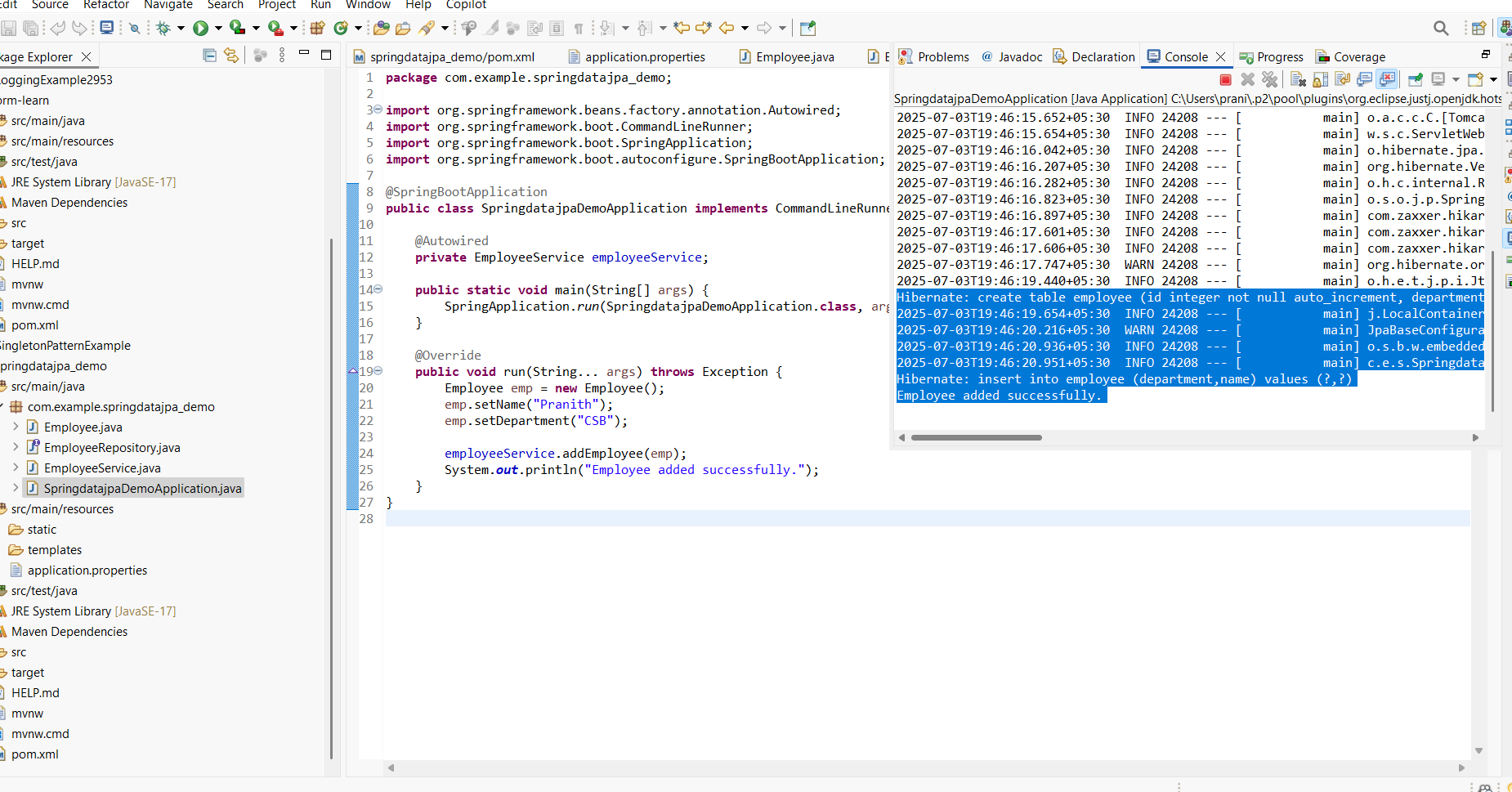
spring.datasource.password=Bunny###1

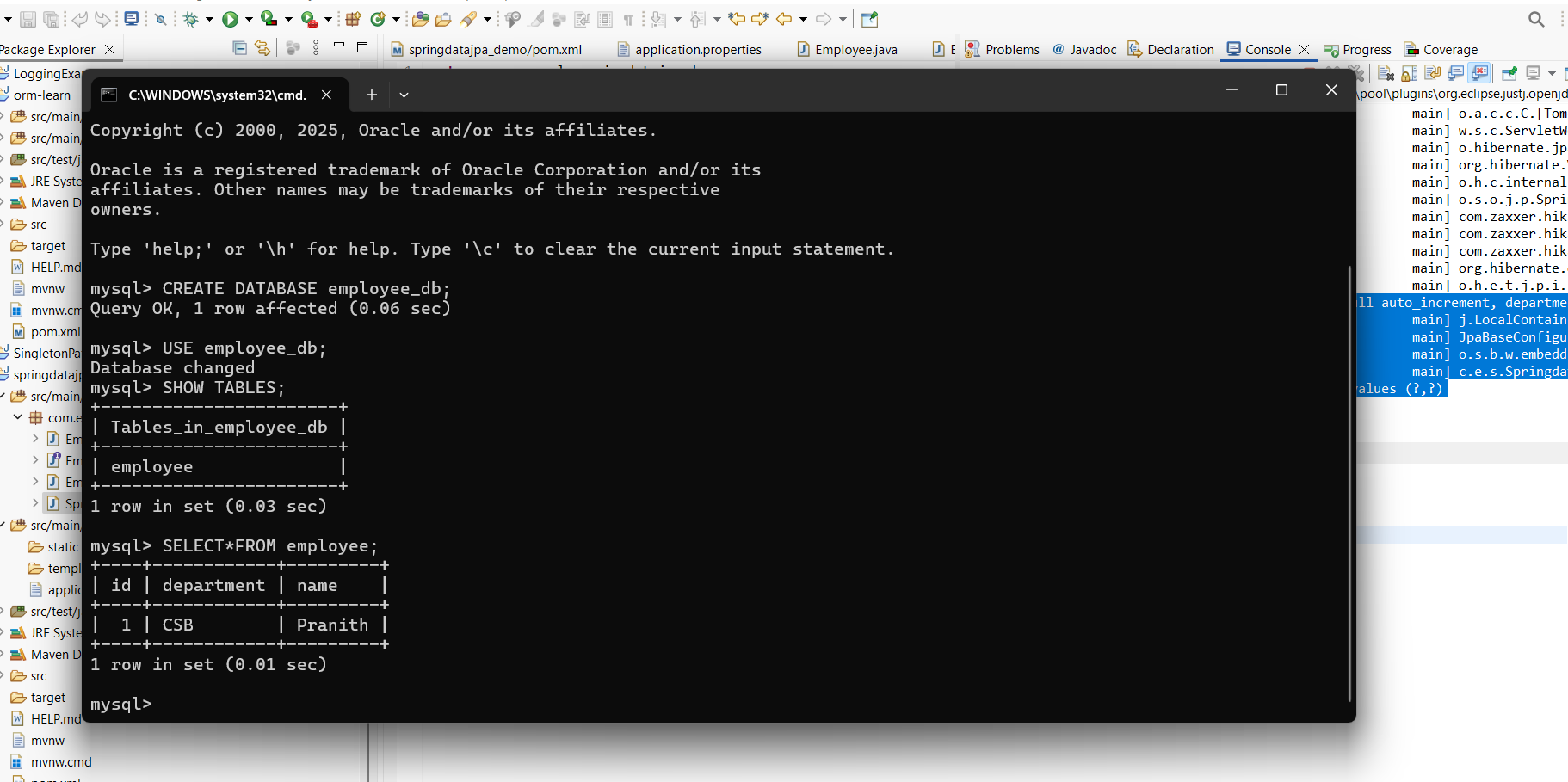
#Hibernate Properties

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

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

**OUTPUT:**

****