**Spring Core and Maven**

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

**LibraryManagement/pom.xml/src/ main/java/com/library/MainApp.java/service/BookService.java/repository.java/resources/applicationContext.xml**  
  
**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.library</groupId>

<artifactId>LibraryManagement</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>LibraryManagement</name>

<description>Spring Core Hands-On Project</description>

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.34</version>

</dependency>

</dependencies>

</project>  
  
**BookRepository.java**  
  
package com.library.repository;

public class BookRepository {

public void saveBook(String name) {

System.out.println("Saving book: " + name);

}

}  
  
**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 bookName) {

bookRepository.saveBook(bookName);

}

}  
  
**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 in Action");

}

}  
  
**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

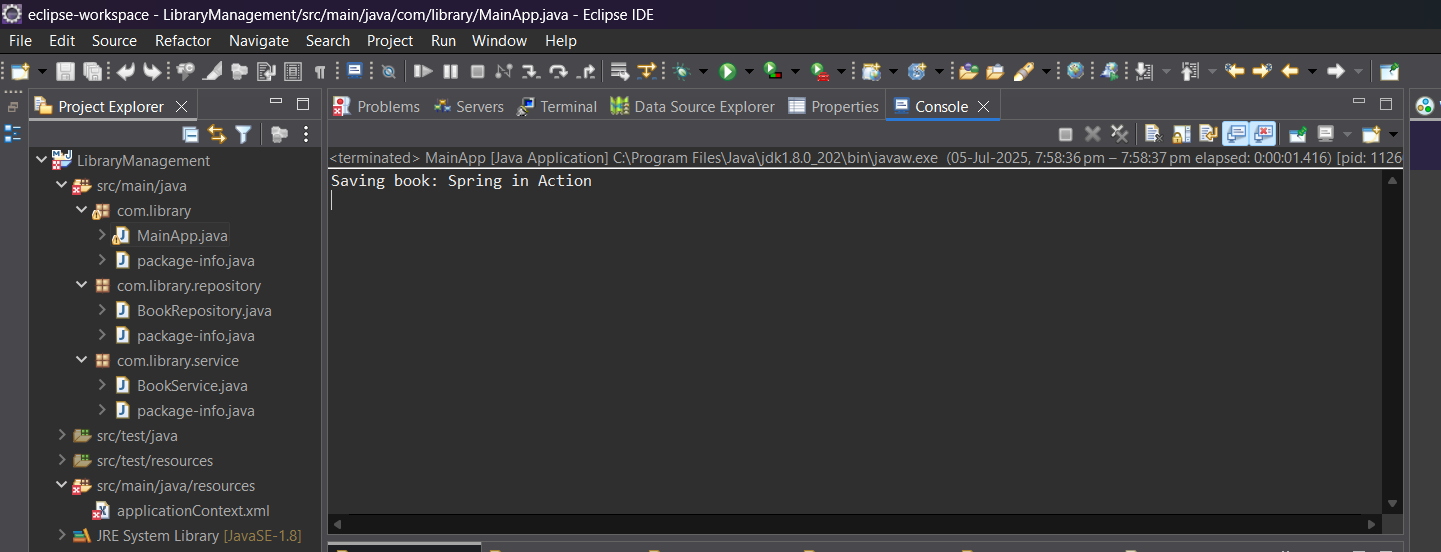
http://www.springframework.org/schema/beans/spring-beans.xsd">

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

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

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

</bean>

</beans>  
  
**Output Screenshot**  
  
Saving book: Spring in Action  
  


**Exercise 2: Implementing Dependency Injection**

**BookService.java**  
  
package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

// Setter method used by Spring for injection

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

bookRepository.saveBook(bookName);

}

}  
  
**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

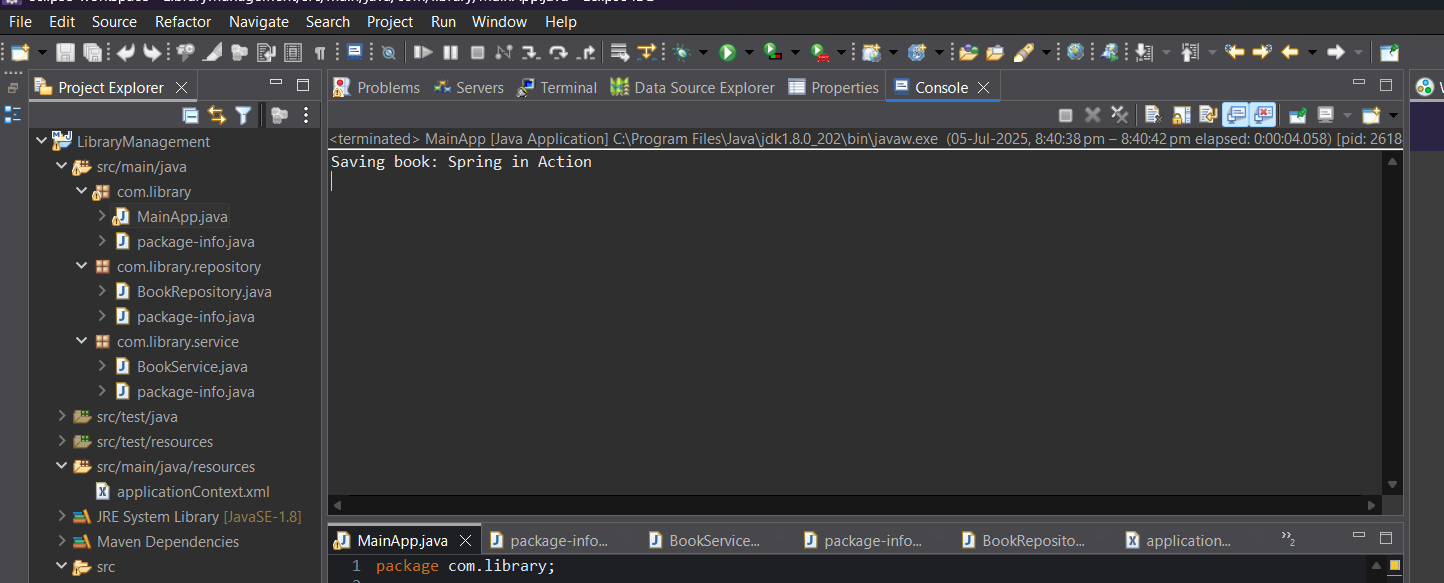
http://www.springframework.org/schema/beans/spring-beans.xsd">

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

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

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

</bean>

</beans>  
  
**Output**  
  
Saving book: Spring in Action  
  


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

**pom.xml with Spring and Maven Compiler Plugin**

<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.library</groupId>

<artifactId>LibraryManagement</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>LibraryManagement</name>

<description>Spring Core Maven Project</description>

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.34</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.10.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

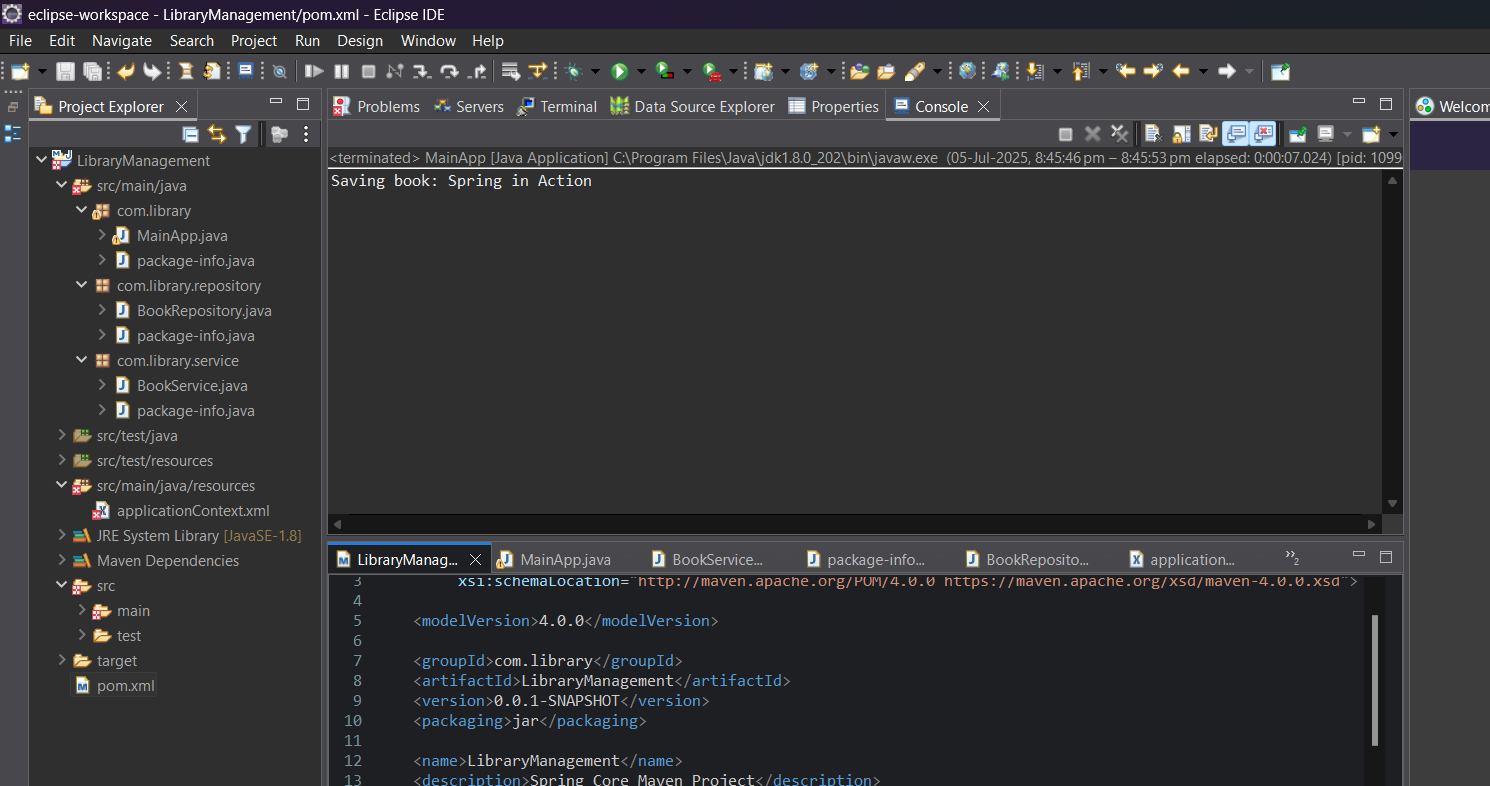
</configuration>

</plugin>

</plugins>

</build>

</project>

**Output**  
  
Saving book: Spring in Action  
  


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

**Hands on 1**

**Spring Data JPA - Quick Example**

**Project Title:** Spring Boot JPA MySQL Integration – Country Demo  
  
**Used:**

* Spring Boot 3.5.3
* Spring Data JPA
* MySQL 8.0.42
* Hibernate 6.6.18
* Java 21

**Packages Created:**

* Copy code
* com.cts.orm.model → Country.java
* com.cts.orm.repository → CountryRepository.java
* com.cts.orm.service → CountryService.java

**Database:**

pgsql

Copy code

Database name: cognizant\_truyum

Table: country (fields: code, name)

**Dependencies (pom.xml):**

* spring-boot-starter-data-jpa
* mysql-connector-j
* spring-boot-starter

**Path in eclipse:**

**src/main/java/com/cts/orm/OrmlearnApplication.java /Country.javarepository/CountryRepository.java/CountryService.javaresources/**

**application.properties**

**application.properties**

spring.datasource.url=jdbc:mysql://localhost:3306/cognizant\_truyum

spring.datasource.username=root

spring.datasource.password=Janakavi@2003

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

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

**Country.java**

package com.cts.orm.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

@Entity

public class Country {

@Id

private String code;

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; }

}

**CountryRepository.java**

package com.cts.orm.repository;

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

import com.cts.orm.model.Country;

public interface CountryRepository extends JpaRepository<Country, String> {

}

**CountryService.java**

package com.cts.orm.service;

import java.util.List;

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

import org.springframework.stereotype.Service;

import com.cts.orm.model.Country;

import com.cts.orm.repository.CountryRepository;

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

public void addCountry(Country country) {

countryRepository.save(country);

}

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

}

**OrmlearnApplication.java**

package com.cts.orm;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

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

import com.cts.orm.model.Country;

import com.cts.orm.service.CountryService;

@SpringBootApplication

@EntityScan("com.cts.orm.model")

public class OrmlearnApplication implements CommandLineRunner {

@Autowired

private CountryService countryService;

public static void main(String[] args) {

SpringApplication.run(OrmlearnApplication.class, args);

}

@Override

public void run(String... args) {

Country country = new Country();

country.setCode("IN");

country.setName("India");

countryService.addCountry(country);

countryService.getAllCountries().forEach(c ->

System.out.println(c.getCode() + " - " + c.getName()));

}

}

**Console Output**

. \_\_\_\_ \_ \_\_ \_ \_

/\\ / \_\_\_'\_ \_\_ \_ \_(\_)\_ \_\_ \_\_ \_ \ \ \ \

( ( )\\_\_\_ | '\_ | '\_| | '\_ \/ \_` | \ \ \ \

\\/ \_\_\_)| |\_)| | | | | || (\_| | ) ) ) )

' |\_\_\_\_| .\_\_|\_| |\_|\_| |\_\\_\_, | / / / /

=========|\_|==============|\_\_\_/=/\_/\_/\_/

:: Spring Boot :: (v3.5.3)

2025-07-05T22:03:00.654+05:30 INFO 4700 --- [ main] com.cts.orm.OrmlearnApplication : Starting OrmlearnApplication using Java 21.0.7 with PID 4700 (C:\Users\Admin\eclipse-workspace\ormlearn\target\classes started by Admin in C:\Users\Admin\eclipse-workspace\ormlearn)

2025-07-05T22:03:00.657+05:30 INFO 4700 --- [ main] com.cts.orm.OrmlearnApplication : No active profile set, falling back to 1 default profile: "default"

2025-07-05T22:03:01.435+05:30 INFO 4700 --- [ main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.

2025-07-05T22:03:01.562+05:30 INFO 4700 --- [ main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 112 ms. Found 1 JPA repository interface.

2025-07-05T22:03:02.207+05:30 INFO 4700 --- [ main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]

2025-07-05T22:03:02.284+05:30 INFO 4700 --- [ main] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.6.18.Final

2025-07-05T22:03:02.335+05:30 INFO 4700 --- [ main] o.h.c.internal.RegionFactoryInitiator : HHH000026: Second-level cache disabled

2025-07-05T22:03:02.852+05:30 INFO 4700 --- [ main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA class transformer

2025-07-05T22:03:02.897+05:30 INFO 4700 --- [ main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...

2025-07-05T22:03:03.547+05:30 INFO 4700 --- [ main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection com.mysql.cj.jdbc.ConnectionImpl@68ea253b

2025-07-05T22:03:03.550+05:30 INFO 4700 --- [ main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.

2025-07-05T22:03:03.718+05:30 WARN 4700 --- [ main] org.hibernate.orm.deprecation : HHH90000025: MySQLDialect does not need to be specified explicitly using 'hibernate.dialect' (remove the property setting and it will be selected by default)

2025-07-05T22:03:03.754+05:30 INFO 4700 --- [ main] org.hibernate.orm.connections.pooling : HHH10001005: Database info:

Database JDBC URL [Connecting through datasource 'HikariDataSource (HikariPool-1)']

Database driver: undefined/unknown

Database version: 8.0.42

Autocommit mode: undefined/unknown

Isolation level: undefined/unknown

Minimum pool size: undefined/unknown

Maximum pool size: undefined/unknown

2025-07-05T22:03:05.262+05:30 INFO 4700 --- [ main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)

Hibernate: create table country (code varchar(255) not null, name varchar(255), primary key (code)) engine=InnoDB

2025-07-05T22:03:05.506+05:30 INFO 4700 --- [ main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'

2025-07-05T22:03:06.014+05:30 INFO 4700 --- [ main] com.cts.orm.OrmlearnApplication : Started OrmlearnApplication in 6.131 seconds (process running for 6.728)

Hibernate: select c1\_0.code,c1\_0.name from country c1\_0 where c1\_0.code=?

Hibernate: insert into country (name,code) values (?,?)

Hibernate: select c1\_0.code,c1\_0.name from country c1\_0

IN - India

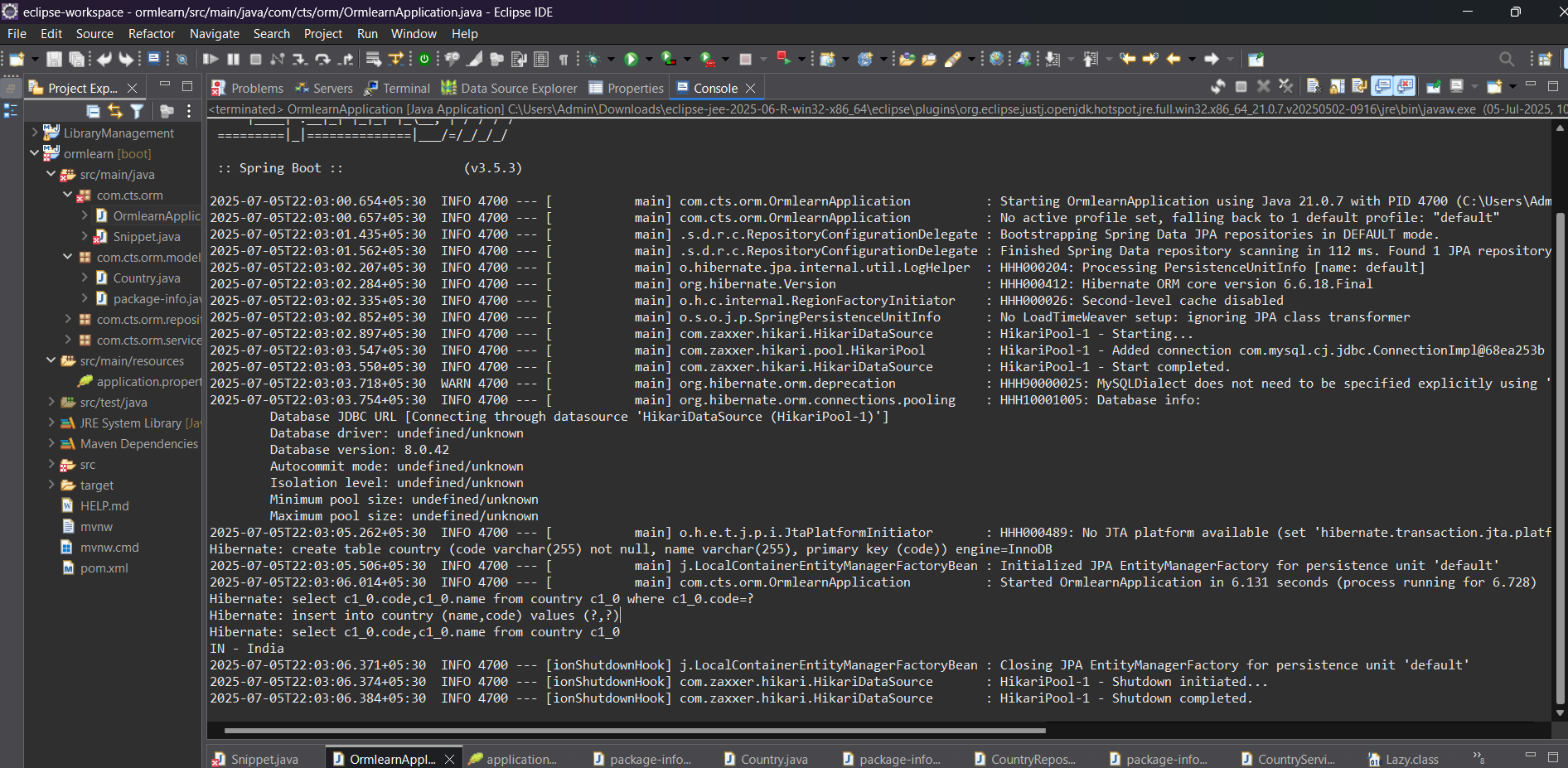
2025-07-05T22:03:06.371+05:30 INFO 4700 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'

2025-07-05T22:03:06.374+05:30 INFO 4700 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...

2025-07-05T22:03:06.384+05:30 INFO 4700 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.

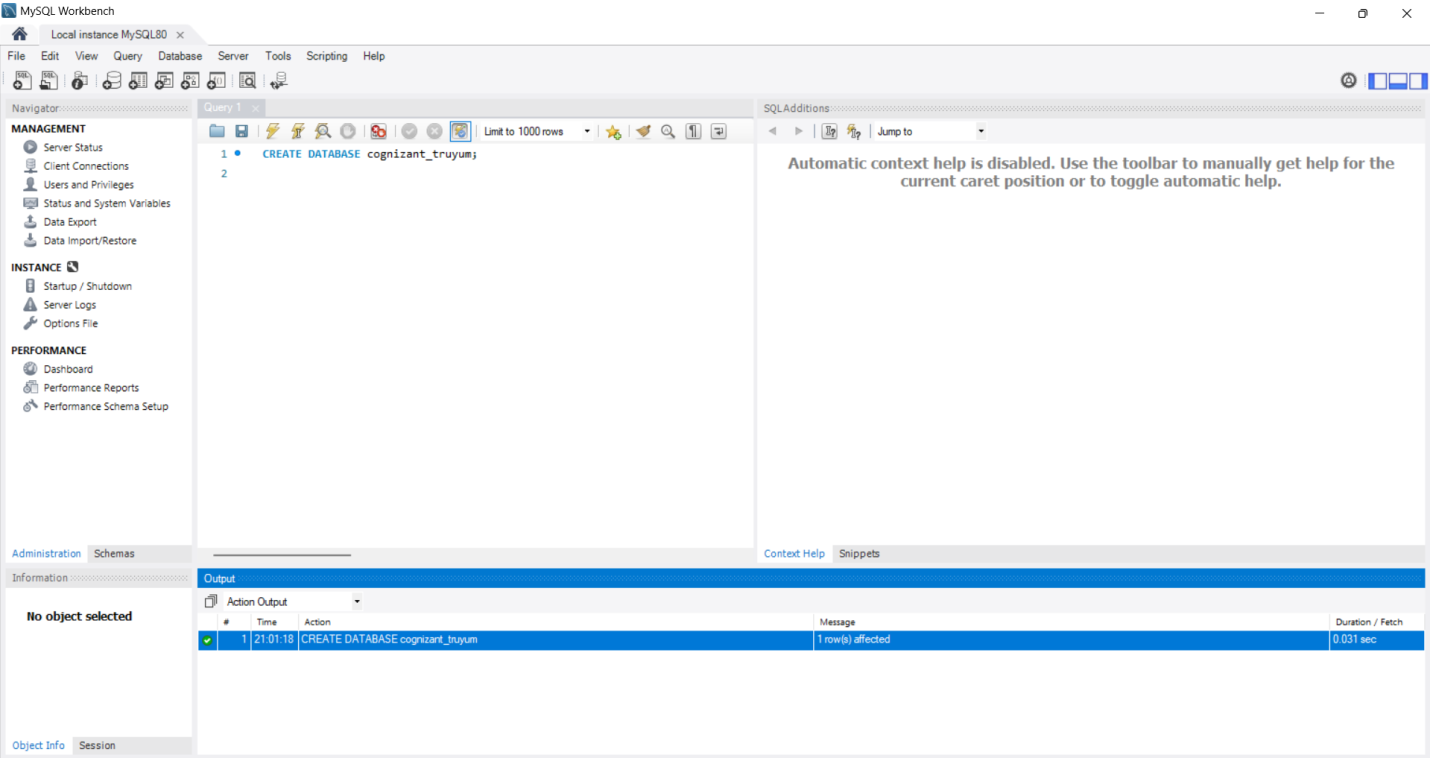
Hibernate: insert into country (name, code) values (?, ?)

Hibernate: select ...

IN – India  
  


**Spring Boot application:**

* Connected to MySQL
* Created the table country
* Inserted and displayed data successfully.



**Hands on 4**

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

### Difference between JPA, Hibernate and Spring Data JPA Java Persistence API (JPA)

* Specification for persisting, reading, and managing Java objects.
* Doesn't contain implementation.
* Hibernate is one such implementation.

### Hibernate

* ORM tool implementing JPA.
* Requires manual setup for sessions, transactions, etc.

### Spring Data JPA

* Abstraction over JPA (not a provider).
* Reduces boilerplate using Spring.
* Manages transactions automatically.

**Code Comparison  
  
Hibernate**

public Integer addEmployee(Employee employee){

Session session = factory.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;

}

**Spring Data JPA**

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

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

}

**Main class**

EntityManagerFactory emf = Persistence.createEntityManagerFactory("country-pu");

EntityManager em = emf.createEntityManager();

Country country = new Country("IN", "India");

em.getTransaction().begin();

em.persist(country);

em.getTransaction().commit();

**Output:**

IN – India  
  
  
Jul 05, 2025 11:27:06 PM org.hibernate.jpa.internal.util.LogHelper logPersistenceUnitInformation

INFO: HHH000204: Processing PersistenceUnitInfo [name: country-pu]

Jul 05, 2025 11:27:06 PM org.hibernate.Version logVersion

INFO: HHH000412: Hibernate ORM core version 6.4.4.Final

Jul 05, 2025 11:27:06 PM org.hibernate.cache.internal.RegionFactoryInitiator initiateService

INFO: HHH000026: Second-level cache disabled

Jul 05, 2025 11:27:07 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure

WARN: HHH10001002: Using built-in connection pool (not intended for production use)

Jul 05, 2025 11:27:07 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001005: Loaded JDBC driver class: com.mysql.cj.jdbc.Driver

Jul 05, 2025 11:27:07 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001012: Connecting with JDBC URL [jdbc:mysql://localhost:3306/cognizant\_truyum]

Jul 05, 2025 11:27:07 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001001: Connection properties: {password=\*\*\*\*, user=root}

Jul 05, 2025 11:27:07 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator

INFO: HHH10001003: Autocommit mode: false

Jul 05, 2025 11:27:07 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>

INFO: HHH10001115: Connection pool size: 20 (min=1)

Jul 05, 2025 11:27:07 PM org.hibernate.engine.jdbc.dialect.internal.DialectFactoryImpl constructDialect

WARN: HHH90000025: MySQLDialect does not need to be specified explicitly using 'hibernate.dialect' (remove the property setting and it will be selected by default)

Jul 05, 2025 11:27:09 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService

INFO: HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)

Jul 05, 2025 11:27:09 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection

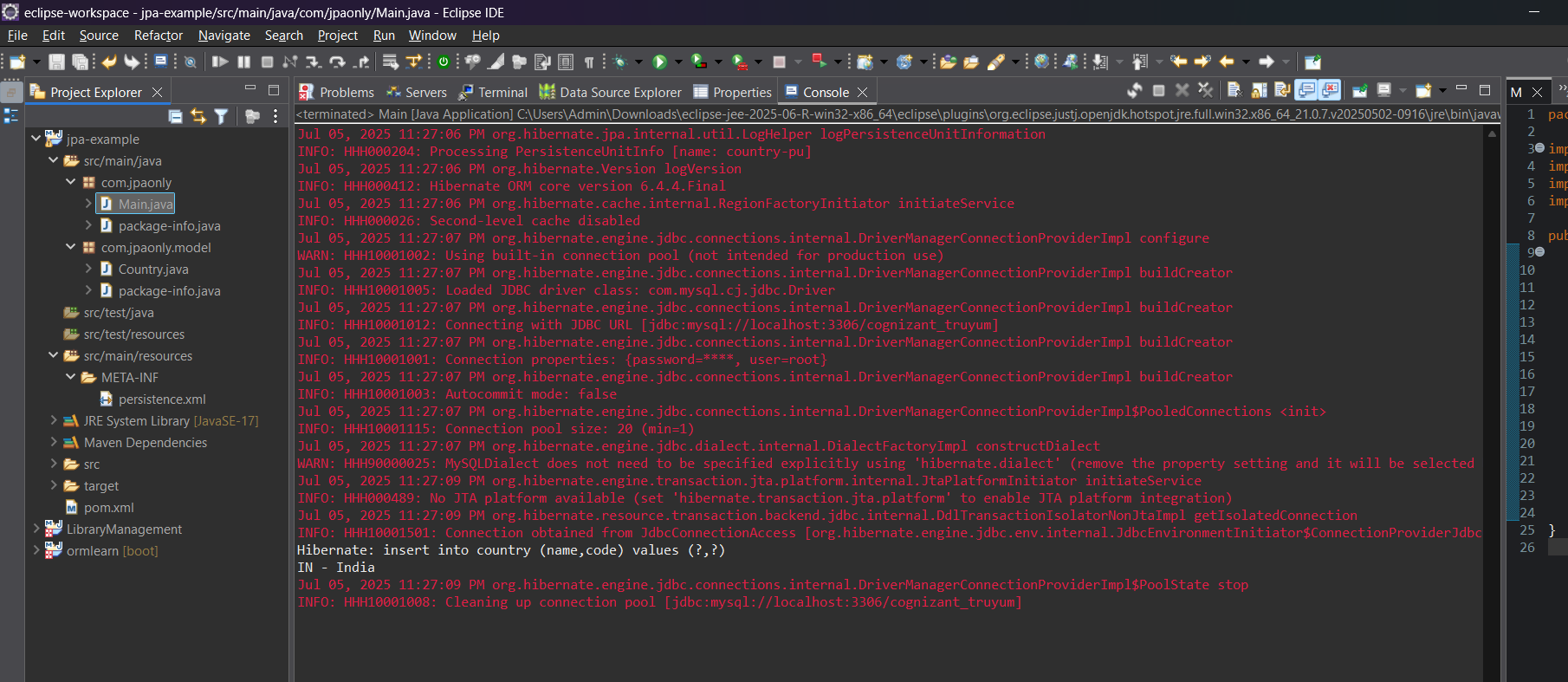
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@2785db06] for (non-JTA) DDL execution was not in auto-commit mode; the Connection 'local transaction' will be committed and the Connection will be set into auto-commit mode.

Hibernate: insert into country (name,code) values (?,?)

IN - India

Jul 05, 2025 11:27:09 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop

INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/cognizant\_truyum]



**Output :**

1. Created a Spring Boot application
2. Configured application.properties with MySQL
3. Used @Entity (Country)
4. Created @Repository (CountryRepository)
5. Injected it into a @Service class (CountryService)
6. Ran it with OrmlearnApplication.java
7. Verified console output including:

* Hibernate startup logs
* Table creation
* Insert queries
* IN - India output (final success)

**Thankyou**