Step 1: Install Required Software

First, I installed all the necessary tools on my system:

~ MySQL Server 8.0 – for the relational database.

~ MySQL Workbench 8 – to manage MySQL easily.

~ Eclipse IDE for Enterprise Java Developers (2019-03 R or later) – to write and run the Java application.

~ Apache Maven 3.6.2 – to manage dependencies and build the project.

Step 2: Create a Spring Boot Project Using Spring Initializr

Next, I created a Spring Boot project by following these steps:

.I opened https://start.spring.io

.Set the Group to com.cognizant

.Set the Artifact to orm-learn

In the Description, I entered:

Demo project for Spring Data JPA and Hibernate

I selected the following dependencies:

Spring Boot DevTools

Spring Data JPA

MySQL Driver

I clicked Generate to download the ZIP file.

Then, I extracted the ZIP into my Eclipse workspace.

Step 3: Import the Project into Eclipse

To bring the project into Eclipse:

I went to File > Import > Maven > Existing Maven Projects

I browsed to the folder I extracted earlier and selected it

Then I clicked Finish to complete the import.

Step 4: Set Up the Database Schema

Now I set up the database in MySQL:

I opened MySQL Workbench or command-line

I ran the following SQL commands to create a schema and table:

create schema 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');

Step 5: Configure Database Properties

Then I opened the file:

src/main/resources/application.properties

And I added the following configuration:

# 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

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.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

spring.jpa.hibernate.ddl-auto=validate

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

This configuration connects the Spring Boot application to the MySQL database.

Step 6: Create the Country Model Class

I created a new package:

com.cognizant.ormlearn.model

Inside that, I created the class Country.java as follows:

package com.cognizant.ormlearn.model;

import javax.persistence.\*;

@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 + "]";

}

}

Step 7: Create the Repository Interface

Then I created a new package:

com.cognizant.ormlearn.repository

Inside it, I added the interface 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> {

}

This interface provides CRUD operations for the Country entity.

Step 8: Create the Service Class

I created another package:

com.cognizant.ormlearn.service

Then I added the class CountryService.java:

package com.cognizant.ormlearn.service;

import java.util.List;

import javax.transaction.Transactional;

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

import org.springframework.stereotype.Service;

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();

}

}

Step 9: Modify the Main Class for Testing

I opened the file:

OrmLearnApplication.java

And I updated it as shown:

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

private static CountryService countryService;

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");

}

}

This will automatically call the service method when the application runs.

Step 10: Run and Verify the Output

I right-clicked OrmLearnApplication.java and ran it as a Java application.

Console Output:

