Spring Data JPA

ormLearnApplication.java:

```
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);
        countryService = context.getBean(CountryService.class);
        LOGGER.info("Inside main");
        testGetAllCountries();
    }
    private static void testGetAllCountries() {
        LOGGER.info("Start");
        List<Country> countries = countryService.getAllCountries();
        LOGGER.debug("countries={}", countries);
        LOGGER.info("End");
    }
}
```

Country.java:

```
package com.cognizant.ormlearn.model;
import javax.persistence.*;
@Entity
@Table(name = "country")
public class Country {
    @Id
    @Column(name = "code")
    private String code;
    @Column(name = "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.library.repository;
import org.springframework.data.jpa.repository.JpaRepository;
import com.library.entity.Country;
public interface CountryRepository extends JpaRepository<Country, String> {

}
```

countryService.java:

```
package com.library.service;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.library.entity.Country;
import com.library.repository.CountryRepository;
@Service
public class CountryService {
    @Autowired
    private CountryRepository countryRepository;
    public Country findCountryByCode(String code) {
        return countryRepository.findById(code).orElse(null);
    }
    public Country addCountry(Country country) {
        return countryRepository.save(country);
    }
}
```

OrmLearnApplication.properties:

```
package com.library;
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.library.entity.Country;
import com.library.service.CountryService;
@SpringBootApplication
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) throws Exception {
        System.out.println("Started...");
        Country country = countryService.findCountryByCode("IN");
        System.out.println("Found: " + country);
        Country newCountry = new Country();
        newCountry.setCode("AU");
        newCountry.setName("Australia");
        Country saved = countryService.addCountry(newCountry);
        System.out.println("Saved: " + saved);
    }
}
```

application.properties

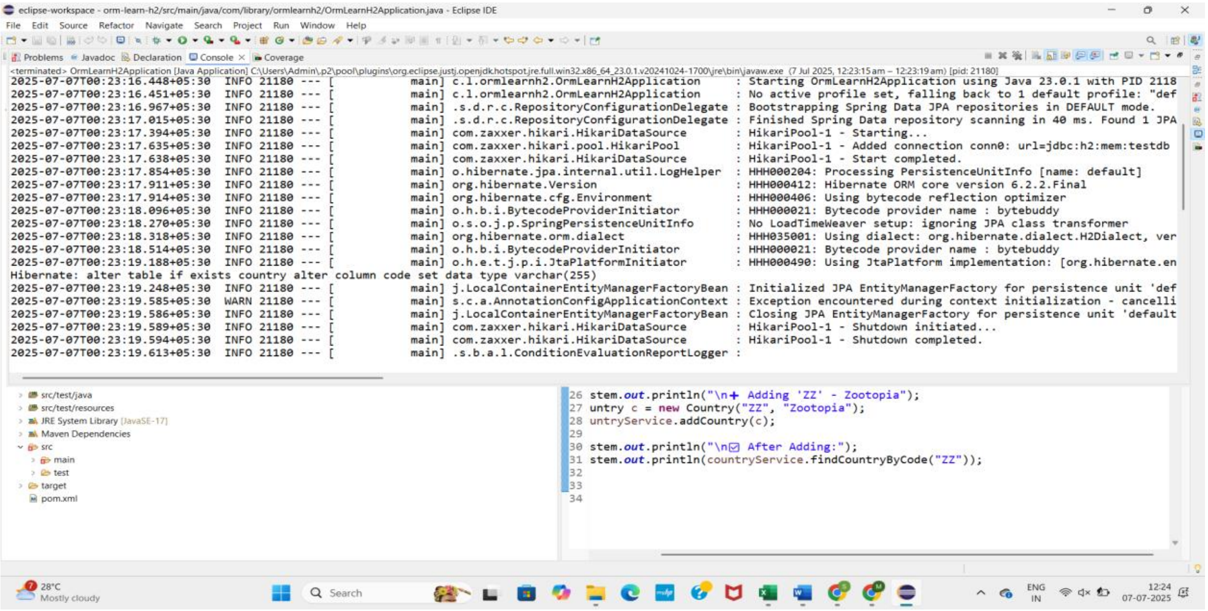
```
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driver-class-name=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.jpa.hibernate.ddl-auto=create
spring.jpa.show-sql=true
spring.h2.console.enabled=true
spring.h2.console.path=/h2-console
```

pox.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
         http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.library</groupId>
    <artifactId>orm-learn</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>ORM Learn</name>
    <description>Spring Boot JPA + MySQL Demo</description>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.18</version> <!-- You can use latest compatible version -->
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </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>
```

OUTPUT:


eclipse-workspace - orm-learn-h2/src/main/java/com/library/ormlearnh2/OrmLearnH2Application.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Problems Javadoc Declaration Console Coverage
src/test/java
src/test/resources
JRE System Library [JavaSE-17]
Maven Dependencies
src
main
test
target
pom.xml
28°C
Mostly cloudy
Search
ENG
IN
12:24
07-07-2025

```
<terminated> OrmLearnH2Application [Java Application] C:\Users\Admin\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v20241024-1700\jre\bin\javaw.exe  (7 Jul 2025, 12:23:15 am – 12:23:19 am) [pid: 21180]
2025-07-07T00:23:16.448+05:30  INFO 21180 --- [           main] c.l.ormlearnh2.OrmLearnH2Application     : Starting OrmLearnH2Application using Java 23.0.1 with PID 2118
2025-07-07T00:23:16.451+05:30  INFO 21180 --- [           main] c.l.ormlearnh2.OrmLearnH2Application     : No active profile set, falling back to 1 default profile: "def
2025-07-07T00:23:16.967+05:30  INFO 21180 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2025-07-07T00:23:17.015+05:30  INFO 21180 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 40 ms. Found 1 JPA
2025-07-07T00:23:17.394+05:30  INFO 21180 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2025-07-07T00:23:17.635+05:30  INFO 21180 --- [           main] com.zaxxer.hikari.pool.HikariPool        : HikariPool-1 - Added connection conn0: url=jdbc:h2:mem:testdb
2025-07-07T00:23:17.638+05:30  INFO 21180 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2025-07-07T00:23:17.854+05:30  INFO 21180 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo [name: default]
2025-07-07T00:23:17.911+05:30  INFO 21180 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version 6.2.2.Final
2025-07-07T00:23:17.914+05:30  INFO 21180 --- [           main] org.hibernate.cfg.Environment            : HHH000406: Using bytecode reflection optimizer
2025-07-07T00:23:18.096+05:30  INFO 21180 --- [           main] o.h.b.i.BytecodeProviderInitiator        : HHH000021: Bytecode provider name : bytebuddy
2025-07-07T00:23:18.270+05:30  INFO 21180 --- [           main] o.s.o.j.p.SpringPersistenceUnitInfo      : No LoadTimeWeaver setup: ignoring JPA class transformer
2025-07-07T00:23:18.318+05:30  INFO 21180 --- [           main] org.hibernate.orm.dialect                : HHH035001: Using dialect: org.hibernate.dialect.H2Dialect, ver
2025-07-07T00:23:18.514+05:30  INFO 21180 --- [           main] o.h.b.i.BytecodeProviderInitiator        : HHH000021: Bytecode provider name : bytebuddy
2025-07-07T00:23:19.188+05:30  INFO 21180 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000490: Using JtaPlatform implementation: [org.hibernate.en
Hibernate: alter table if exists country alter column code set data type varchar(255)
2025-07-07T00:23:19.248+05:30  INFO 21180 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'def
2025-07-07T00:23:19.585+05:30  WARN 21180 --- [           main] s.c.a.AnnotationConfigApplicationContext : Exception encountered during context initialization - cancelli
2025-07-07T00:23:19.586+05:30  INFO 21180 --- [           main] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default
2025-07-07T00:23:19.589+05:30  INFO 21180 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown initiated...
2025-07-07T00:23:19.594+05:30  INFO 21180 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown completed.
2025-07-07T00:23:19.613+05:30  INFO 21180 --- [           main] .s.b.a.l.ConditionEvaluationReportLogger :
```

```
stem.out.println("\n➕ Adding 'ZZ' - Zootopia");
untry c = new Country("ZZ", "Zootopia");
untryService.addCountry(c);

stem.out.println("\n☑ After Adding:");
stem.out.println(countryService.findCountryByCode("ZZ"));
```

Difference between JPA, Hibernate and Spring Data JPA

Feature	JPA (Java Persistence API)	Hibernate	Spring Data JPA
Type	Specification	Implementation of JPA	Abstraction layer on top of JPA
Provides	API/annotations only	ORM features + native SQL, HQL, caching	Repository interfaces and query methods
Requires Implementation	Yes – needs a provider like Hibernate	No – it's already a provider	No – uses JPA provider (like Hibernate) underneath

Feature	JPA (Java Persistence API)	Hibernate	Spring Data JPA
Boilerplate Code	Requires EntityManager, transactions manually	Slightly better but still requires code for common operations	Reduces boilerplate via JpaRepository, CrudRepository etc.
Integration in Spring	Manual configuration required	Can be configured easily	Auto-configured in Spring Boot
Query Support	JPQL	JPQL + HQL + Criteria API	JPQL + method names + @Query support
Ease of Use	Intermediate	Moderate	Very high

Hands on 5-Implement services for managing Country

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

    public static void main(String[] args) {
        ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);
        countryService = context.getBean(CountryService.class);

        LOGGER.info("Inside main");

        testGetAllCountries(); // Hands-on 5
    }

    private static void testGetAllCountries() {
        LOGGER.info("Start");
        List<Country> countries = countryService.getAllCountries();
        LOGGER.debug("countries={}", countries);
        LOGGER.info("End");
    }
}
```

Country.java:

```
package com.cognizant.ormlearn.model;

import javax.persistence.*;

@Entity
@Table(name = "country")
public class Country {

    @Id
    @Column(name = "code")
    private String code;

    @Column(name = "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> {}
```

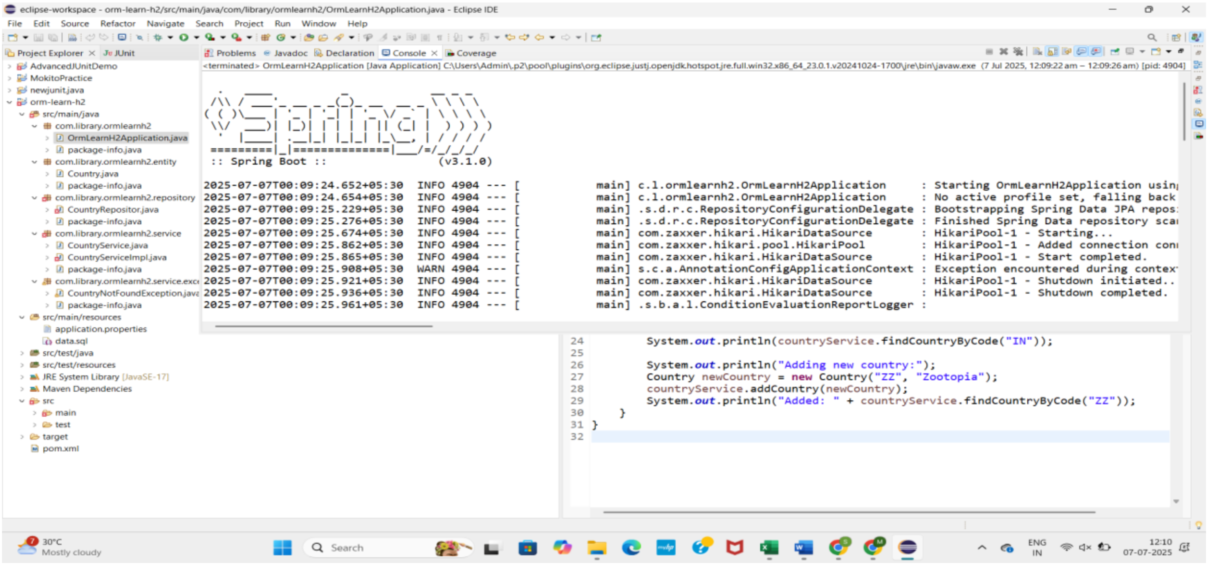
CountryService.java:

```
package com.cognizant.ormlearn.service;
import java.util.List;
import 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();
    }
}
```

Application.properties:

```
spring.datasource.url=jdbc:mysql://localhost:3306/orm_learn
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE
```

OUTPUT:



```
<terminated> OrmLearnH2Application [Java Application] C:\Users\Admin\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v20241024-1700\jre\bin\javaw.exe  (7 Jul 2025, 12:09:22 am – 12:09:26 am) [pid: 4904]

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.1.0)

2025-07-07T00:09:24.652+05:30  INFO 4904 --- [           main] c.l.ormlearnh2.OrmLearnH2Application     : Starting OrmLearnH2Application usin
2025-07-07T00:09:24.654+05:30  INFO 4904 --- [           main] c.l.ormlearnh2.OrmLearnH2Application     : No active profile set, falling back
2025-07-07T00:09:25.229+05:30  INFO 4904 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repos
2025-07-07T00:09:25.276+05:30  INFO 4904 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository sca
2025-07-07T00:09:25.674+05:30  INFO 4904 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2025-07-07T00:09:25.862+05:30  INFO 4904 --- [           main] com.zaxxer.hikari.pool.HikariPool        : HikariPool-1 - Added connection con
2025-07-07T00:09:25.865+05:30  INFO 4904 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2025-07-07T00:09:25.908+05:30  WARN 4904 --- [           main] s.c.a.AnnotationConfigApplicationContext : Exception encountered during contex
2025-07-07T00:09:25.921+05:30  INFO 4904 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown initiated..
2025-07-07T00:09:25.936+05:30  INFO 4904 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Shutdown completed.
2025-07-07T00:09:25.961+05:30  INFO 4904 --- [           main] .s.b.a.l.ConditionEvaluationReportLogger :
```

```
        System.out.println(countryService.findCountryByCode("IN"));

        System.out.println("Adding new country:");
        Country newCountry = new Country("ZZ", "Zootopia");
        countryService.addCountry(newCountry);
        System.out.println("Added: " + countryService.findCountryByCode("ZZ"));
    }
}
```

Find a country based on country code

Country.java

```
package com.library.entity;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
@Entity
@Table(name = "country")
public class Country {
    @Id
    private String code;
    private String name;

    // Getters and 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;
    }
}
```

CountryRepository.java

```
import org.springframework.data.jpa.repository.JpaRepository;
import com.library.entity.Country;

public interface CountryRepository extends JpaRepository<Country, String> {
}
```

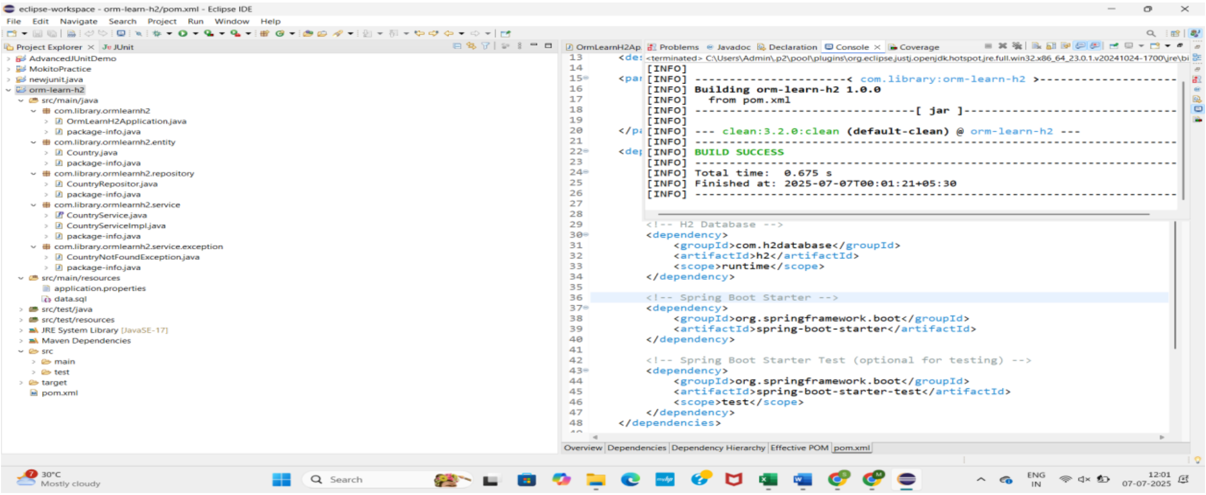
3. OrmLearnApplication.java – Main Class

```
package com.library;

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.library.entity.Country;
import com.library.repository.CountryRepository;

@SpringBootApplication
public class OrmLearnApplication implements CommandLineRunner {

    @Autowired
    private CountryRepository countryRepository;

    public static void main(String[] args) {
        SpringApplication.run(OrmLearnApplication.class, args);
    }

    @Override
    public void run(String... args) throws Exception {
        String code = "IN"; // Replace with user input if needed
        Country country = countryRepository.findById(code).orElse(null);

        if (country != null) {
            System.out.println("Country Code: " + country.getCode());
            System.out.println("Country Name: " + country.getName());
        } else {
            System.out.println("Country not found!");
        }
    }
}
```

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
         http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.library</groupId>
    <artifactId>orm-learn</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>ORM Learn</name>
    <description>Spring Boot JPA + MySQL Demo</description>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.7.18</version> <!-- You can use latest compatible version -->
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <properties>
        <java.version>1.8</java.version>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <!-- Spring Boot Maven Plugin -->
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

OUTPUT:



```
<terminated> C:\Users\Admin\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v20241024-1700\jre\bi
[INFO]
[INFO] ----------------------< com.library:orm-learn-h2 >----------------------
[INFO] Building orm-learn-h2 1.0.0
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ orm-learn-h2 ---
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.675 s
[INFO] Finished at: 2025-07-07T00:01:21+05:30
[INFO] ------------------------------------------------------------------------
```

```
        <!-- H2 Database -->
        <dependency>
            <groupId>com.h2database</groupId>
            <artifactId>h2</artifactId>
            <scope>runtime</scope>
        </dependency>

        <!-- Spring Boot Starter -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>

        <!-- Spring Boot Starter Test (optional for testing) -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
```

Add a new country

Country.java

```
package com.library.entity;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;
@Entity
@Table(name = "country")
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.library.repository;

import org.springframework.data.jpa.repository.JpaRepository;
import com.library.entity.Country;

public interface CountryRepository extends JpaRepository<Country, String> {
}
```

Update OrmLearnApplication.java

```
package com.library;
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.library.entity.Country;
import com.library.repository.CountryRepository;
@SpringBootApplication
public class OrmLearnApplication implements CommandLineRunner {
    @Autowired
    private CountryRepository countryRepository;
    public static void main(String[] args) {
        SpringApplication.run(OrmLearnApplication.class, args);
    }
    @Override
    public void run(String... args) throws Exception {
        // Add new country
        Country newCountry = new Country();
        newCountry.setCode("JP"); // Japan
        newCountry.setName("Japan");

        countryRepository.save(newCountry);
        System.out.println("✅ New country added: " + newCountry.getCode() + " - " + newCountry.getName());
    }
}
```

CountryService.java:

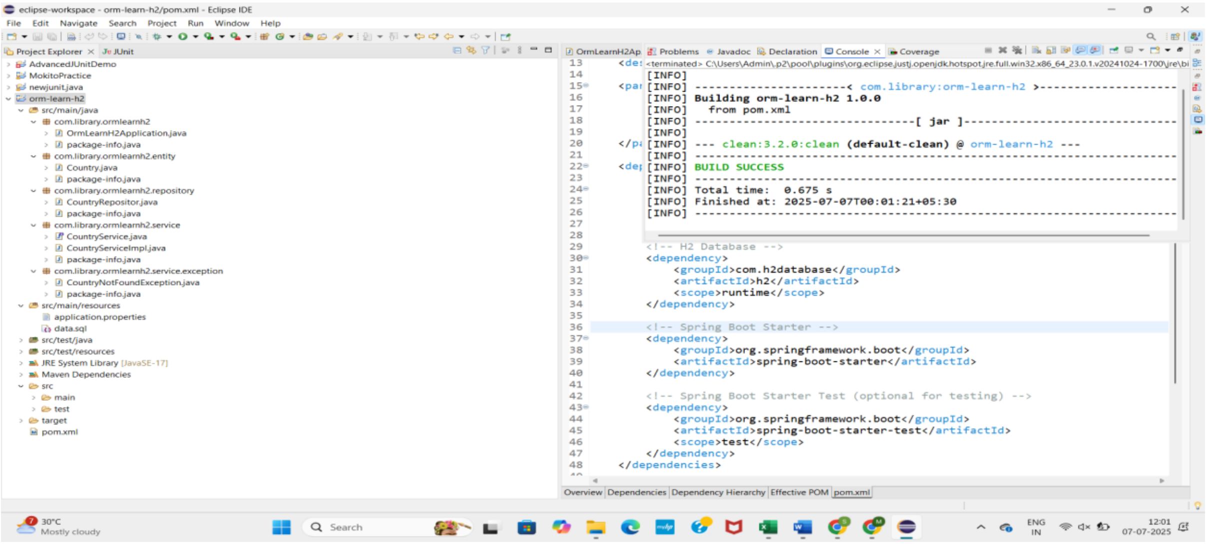
```
package com.library.service;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.library.entity.Country;
import com.library.repository.CountryRepository;

import java.util.Optional;
@Service
public class CountryService {
   @Autowired
   private CountryRepository countryRepository;
   public Country findCountryByCode(String code) {
      Optional<Country> result = countryRepository.findById(code);
      return result.orElse(null);
   }
   public Country addCountry(Country country) {
      return countryRepository.save(country);
   }
}
```

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
       http://maven.apache.org/xsd/maven-4.0.0.xsd">
   <modelVersion>4.0.0</modelVersion>
   <groupId>com.library</groupId>
   <artifactId>orm-learn</artifactId>
   <version>0.0.1-SNAPSHOT</version>
   <name>ORM Learn</name>
   <description>Spring Boot JPA + MySQL Demo</description>
   <parent>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-parent</artifactId>
      <version>2.7.18</version> <!-- You can use latest compatible version -->
      <relativePath/> <!-- lookup parent from repository -->
   </parent>
   <properties>
      <java.version>1.8</java.version>
   </properties>
   <dependencies>
      <dependency>
         <groupId>org.springframework.boot</groupId>
         <artifactId>spring-boot-starter-web</artifactId>
      </dependency>
      <dependency>
         <groupId>org.springframework.boot</groupId>
         <artifactId>spring-boot-starter-data-jpa</artifactId>
      </dependency>
      <dependency>
         <groupId>mysql</groupId>
         <artifactId>mysql-connector-java</artifactId>
         <scope>runtime</scope>
```

```
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-devtools</artifactId>
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <!-- Spring Boot Maven Plugin -->
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

OUTPUT:



```
<terminated> C:\Users\Admin\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v20241024-1700\jre\bi
[INFO]
[INFO] ---------------------< com.library:orm-learn-h2 >---------------------
[INFO] Building orm-learn-h2 1.0.0
[INFO]   from pom.xml
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ orm-learn-h2 ---
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  0.675 s
[INFO] Finished at: 2025-07-07T00:01:21+05:30
[INFO] ------------------------------------------------------------------------
```

```
        <!-- H2 Database -->
        <dependency>
            <groupId>com.h2database</groupId>
            <artifactId>h2</artifactId>
            <scope>runtime</scope>
        </dependency>

        <!-- Spring Boot Starter -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>

        <!-- Spring Boot Starter Test (optional for testing) -->
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
    </dependencies>
```