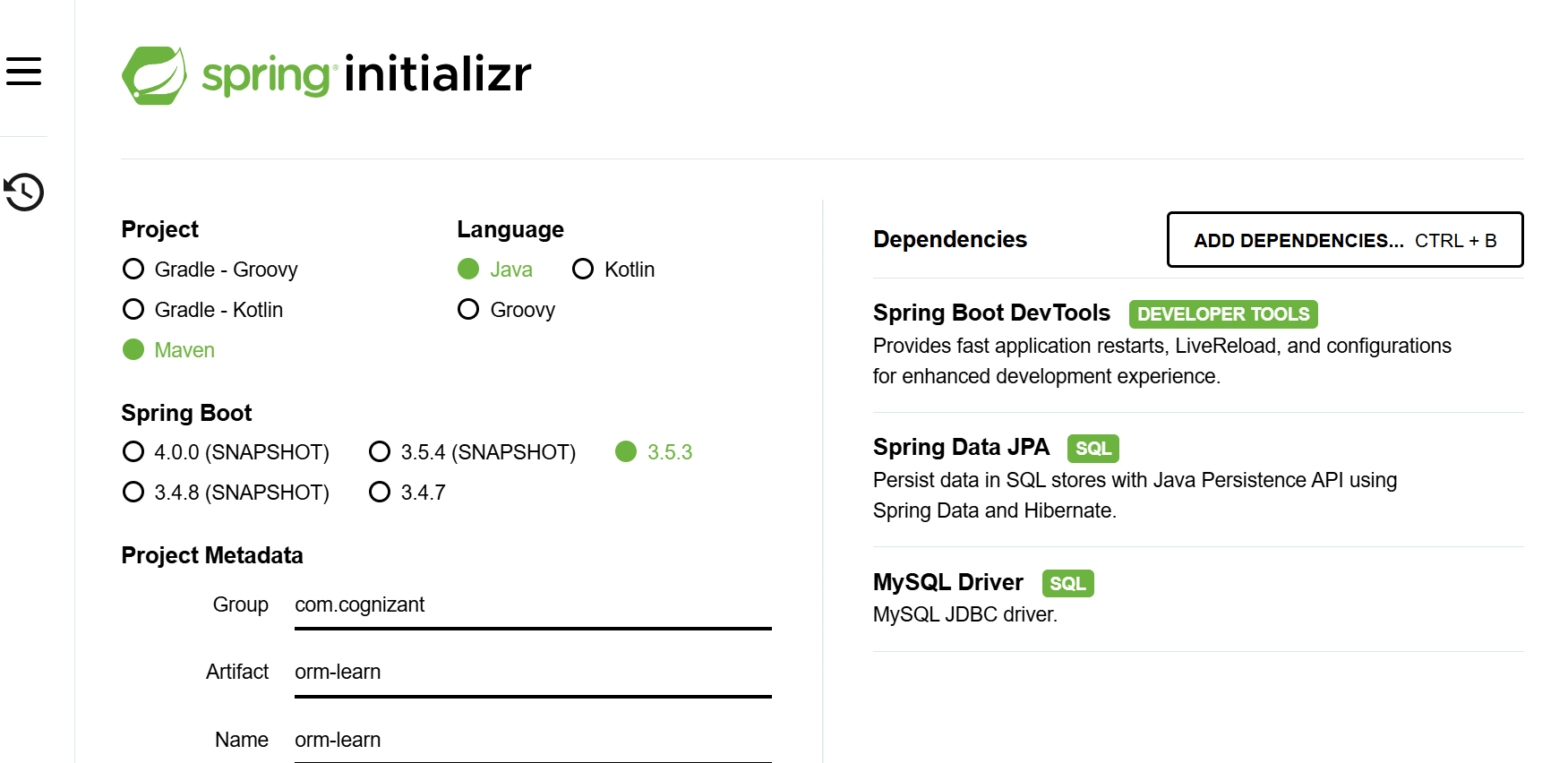
**Name:- Dharan Gowda Superset id:-6387532**

**1) Spring Data JPA - Quick Example** 

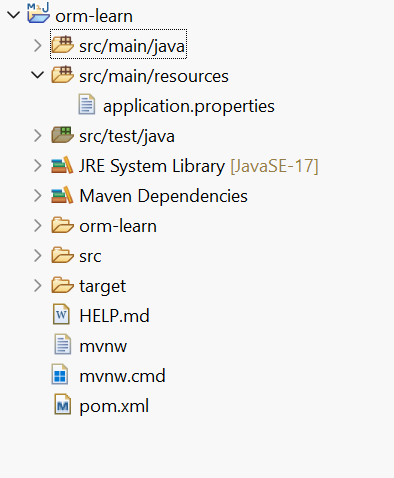
**1. Install Prerequisites**

1. **MySQL Server 8.0**
2. **MySQL Workbench 8.0**
3. **Eclipse IDE for Enterprise Java Developers 2019-03 R**
4. **Maven 3.6.2**
5. **Java 8 or higher installed**

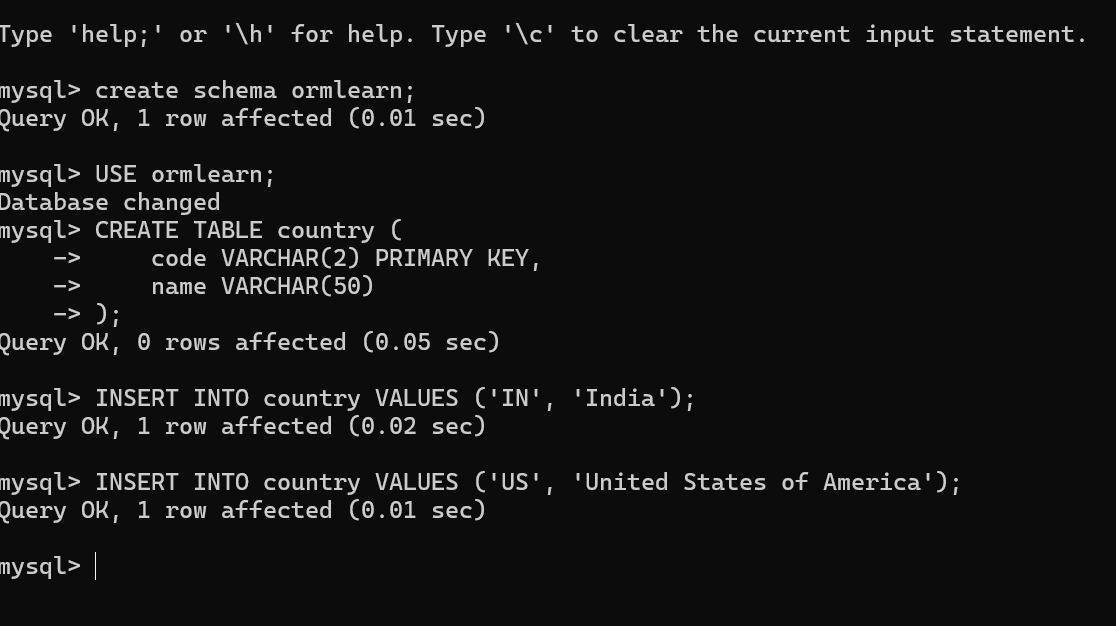
**2. Create Spring Boot Project via Spring Initializer**



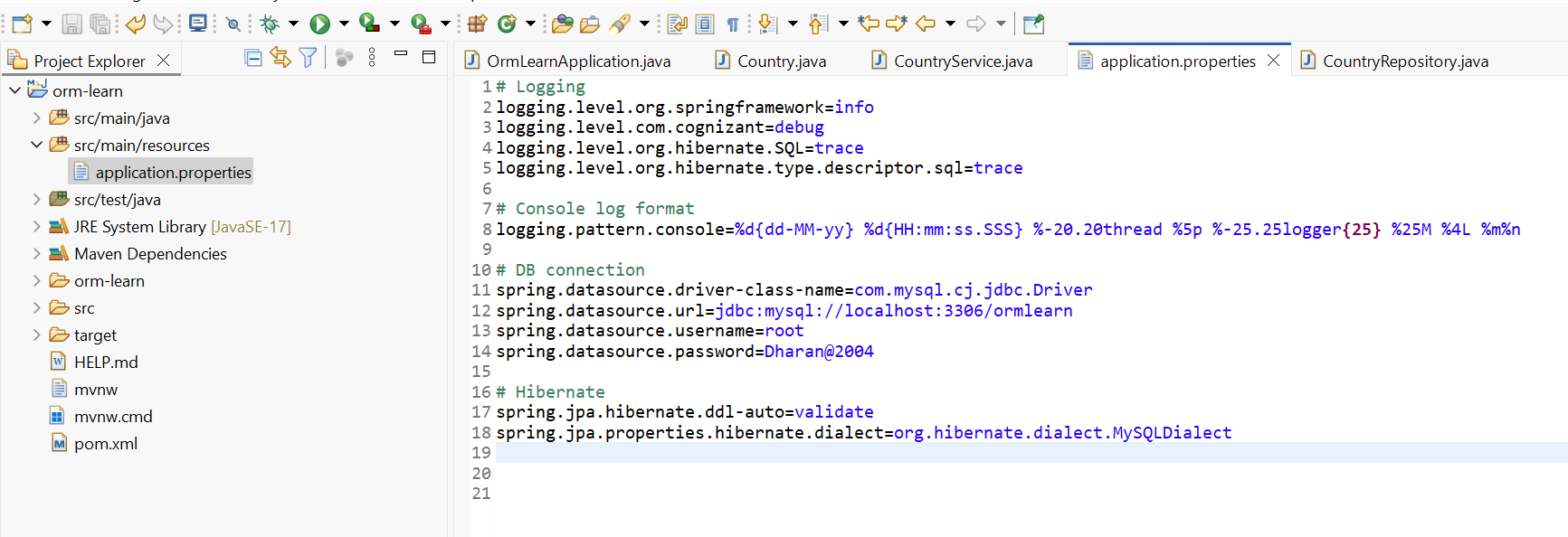
**3. Import into Eclipse**

****

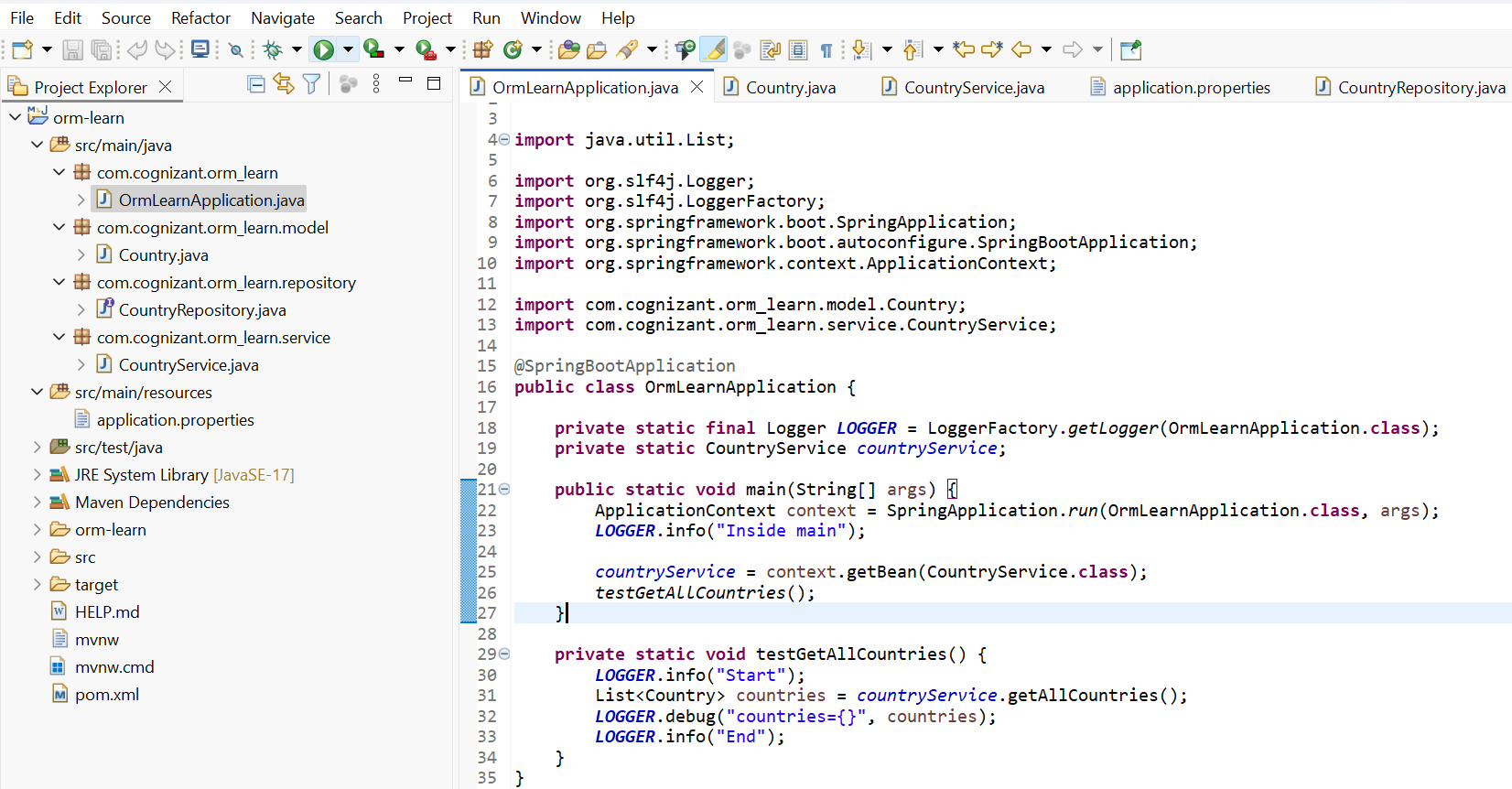
**4. Create MySQL Schema**

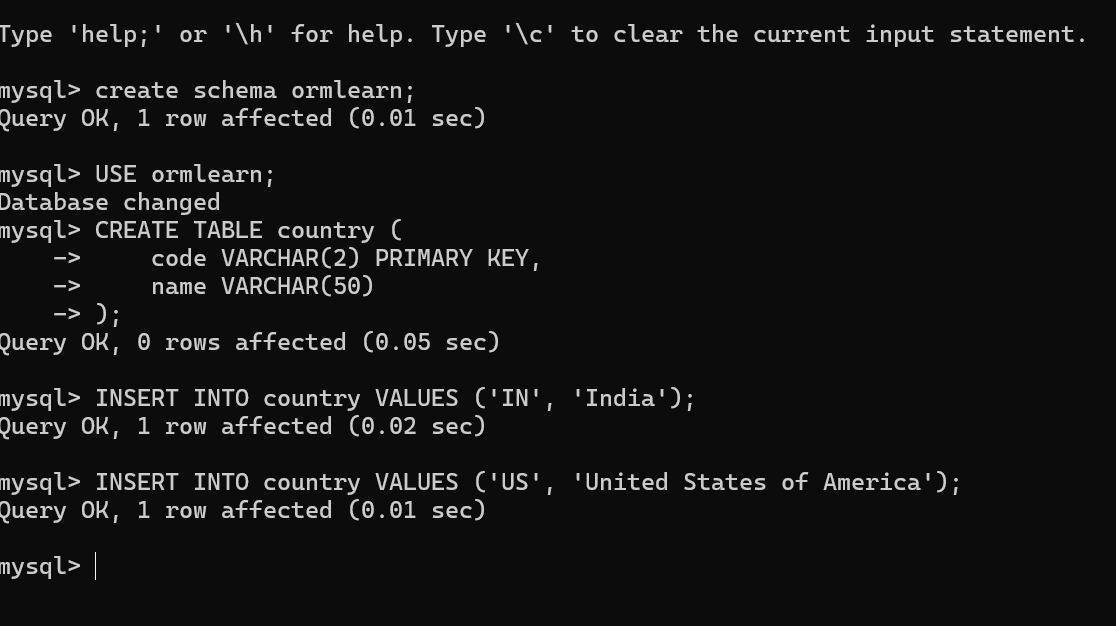
****

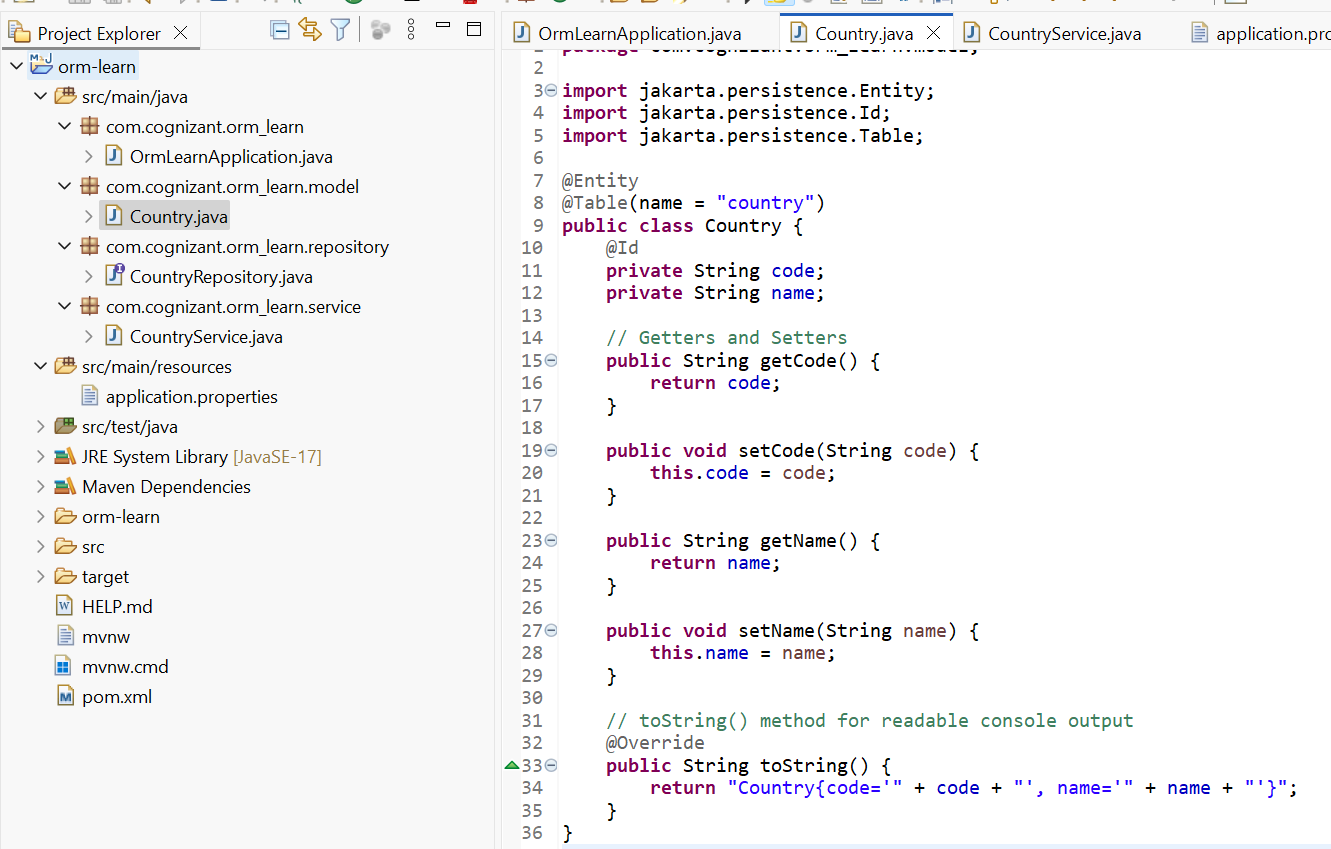
1. **Configure application.properties**

****

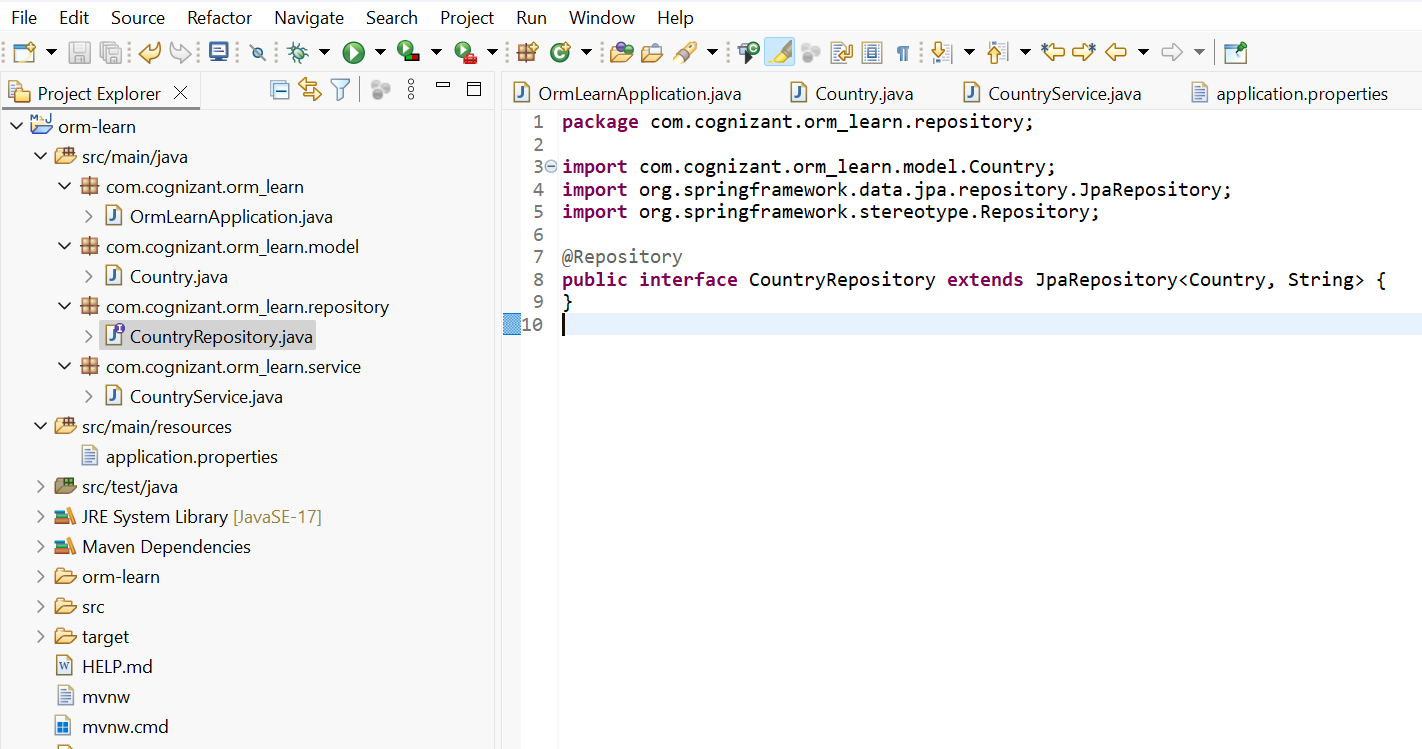
1. **Add Logging to OrmLearnApplication.java**

****

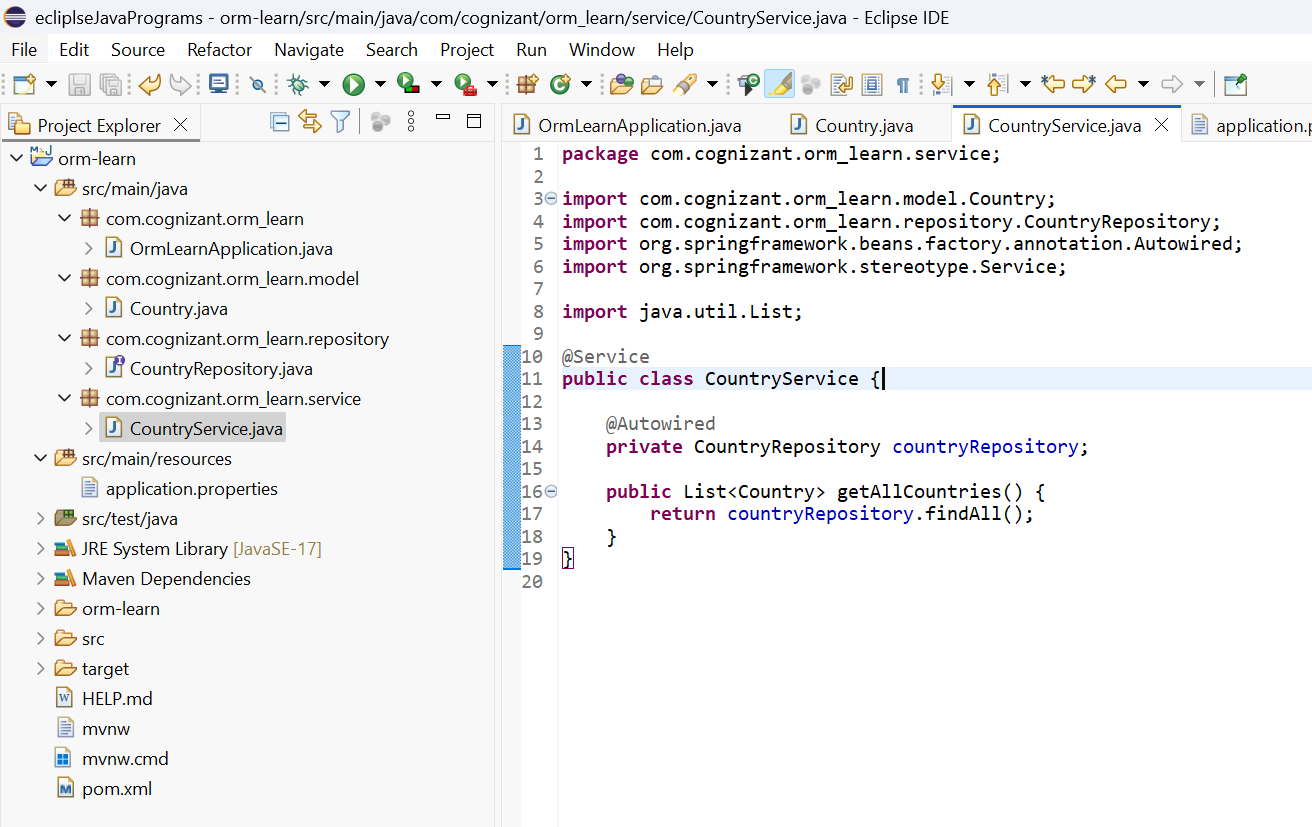
1. **Create MySQL**
2. **Create Entity Class**

****

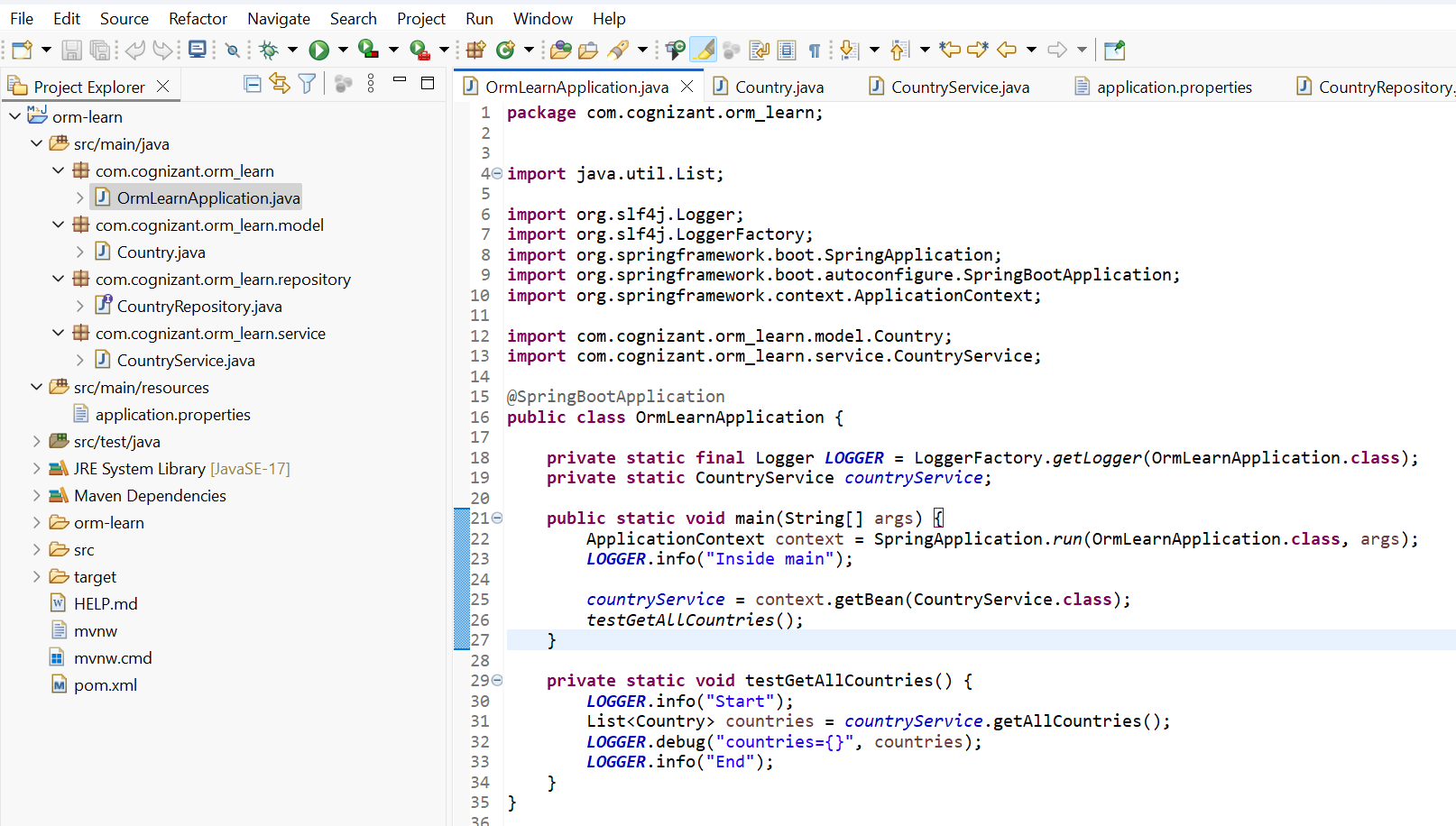
1. **Create Repository**

****

1. **Create Service**

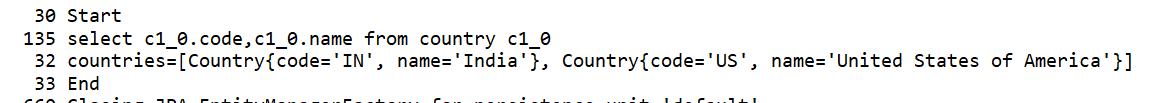
****

**11 Modify OrmLearnApplication.java for Testing**

****

1. **Run the Project**

**Output**

****

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

**Hibernate**

   /\* Method to CREATE an employee in the database \*/

   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 {

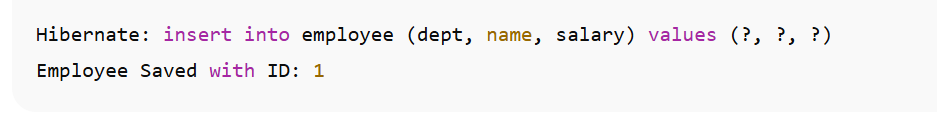
         session.close();

      }

      return employeeID;

   }

**OUTPUT :-**

****

**Spring Data JPA**  
EmployeeRespository.java

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

EmployeeService.java

@Autowire

  private EmployeeRepository employeeRepository;

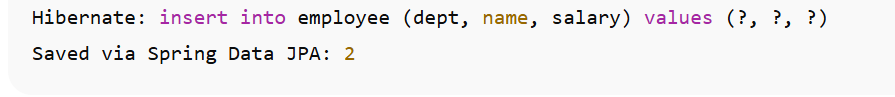
@Transactional

public void addEmployee(Employee employee) {

  employeeRepository.save(employee);

  }

**OUTPUT:-**

****