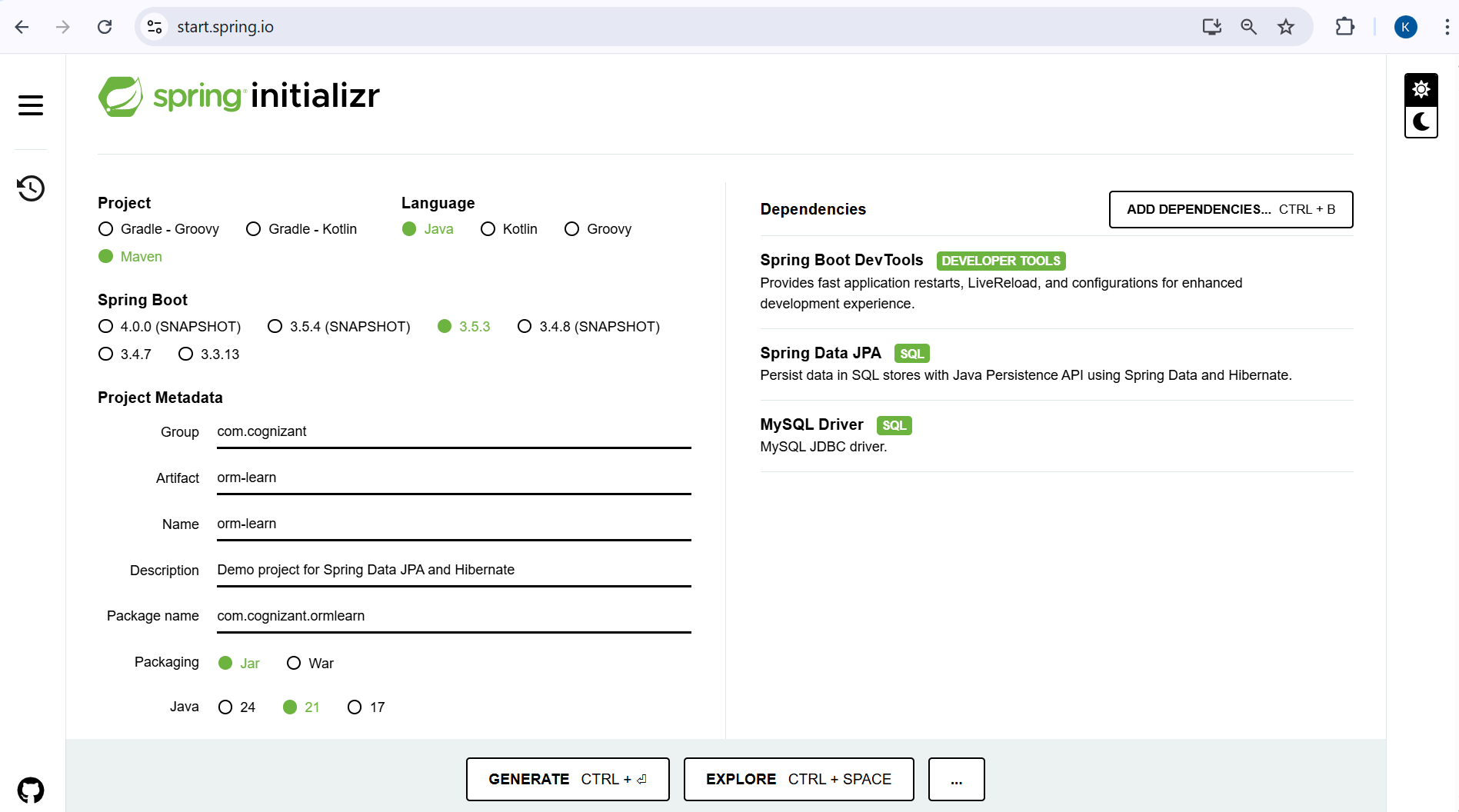
**Week 3 Spring Data JPA Mandatory HandsOn Solutions.**

**Exercise 1**:

**Hands on 1- Spring Data JPA - Quick Example**

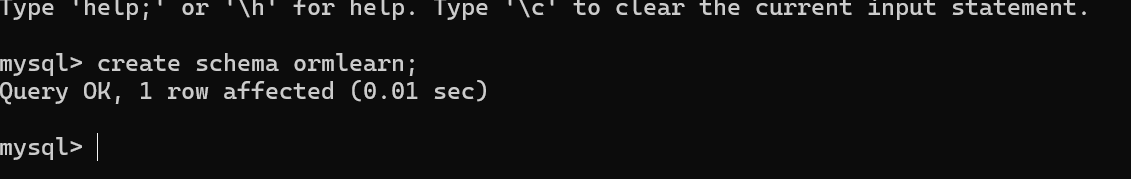
Step-1 : Create Spring Boot Project using Spring Initializr



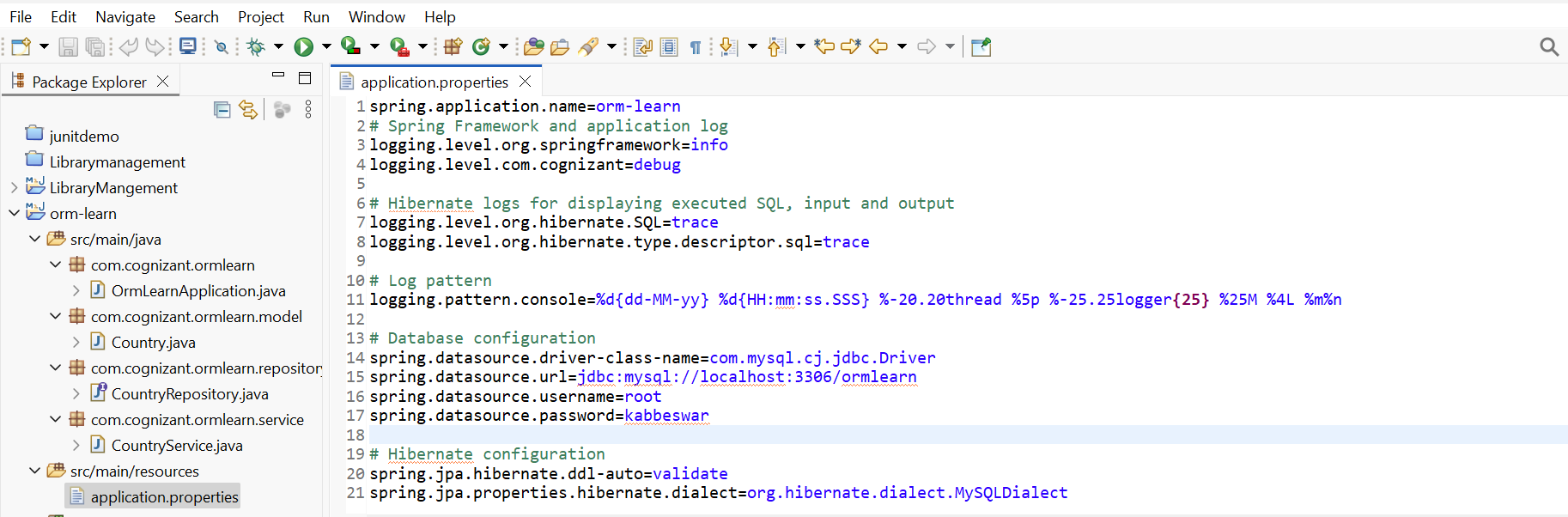
Step 2: Import Project in Eclipse

File → Import → Maven → Existing Maven Projects → Browse → Select project → Finish

Step 3: Create Database Schema

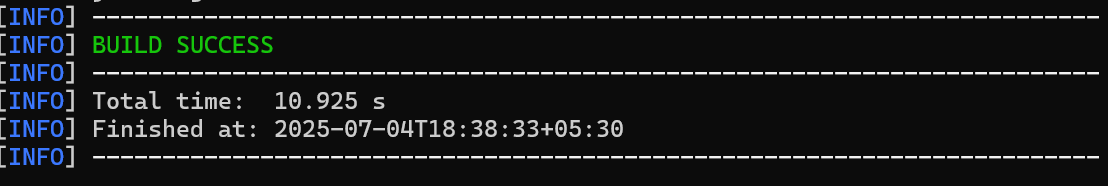


Step 4: Configure application.properties

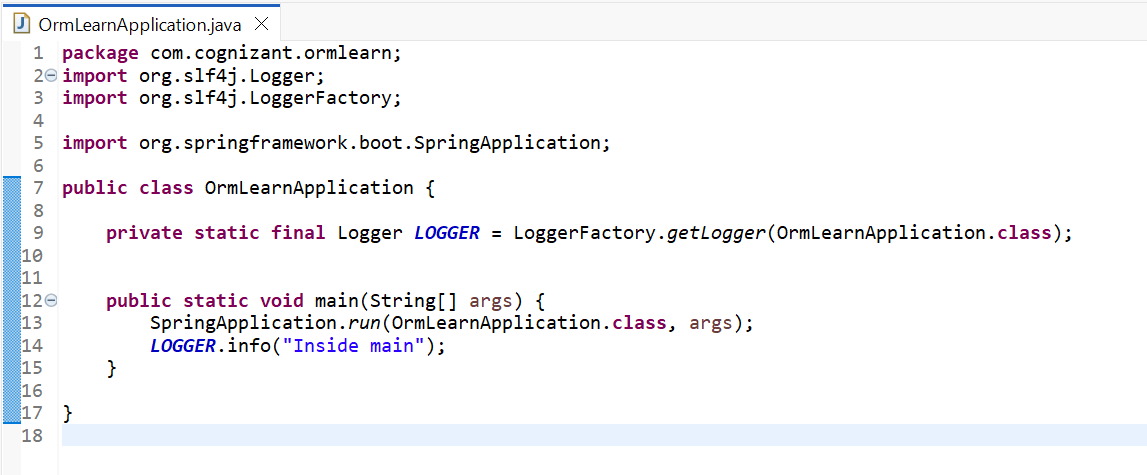


Step 5: Build the Project

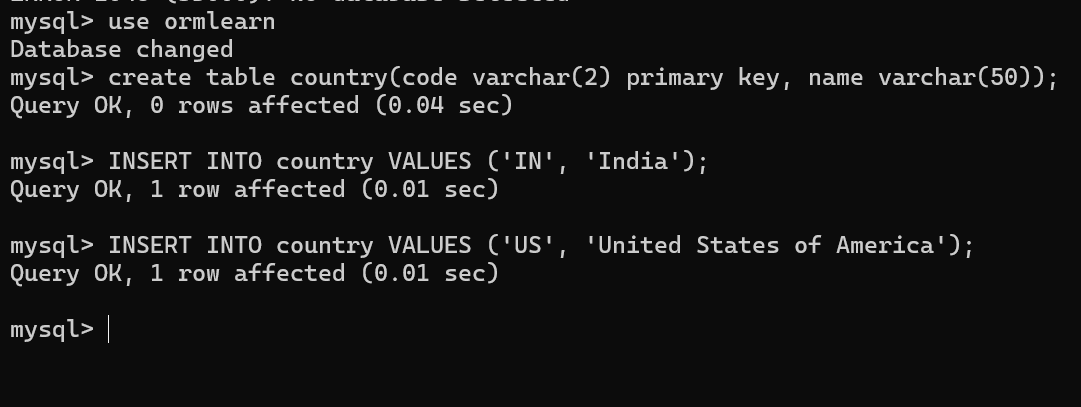
Build the project using ‘mvn clean package -Dhttp.proxyHost=proxy.cognizant.com -Dhttp.proxyPort=6050 -Dhttps.proxyHost=proxy.cognizant.com -Dhttps.proxyPort=6050 -Dhttp.proxyUser=123456’



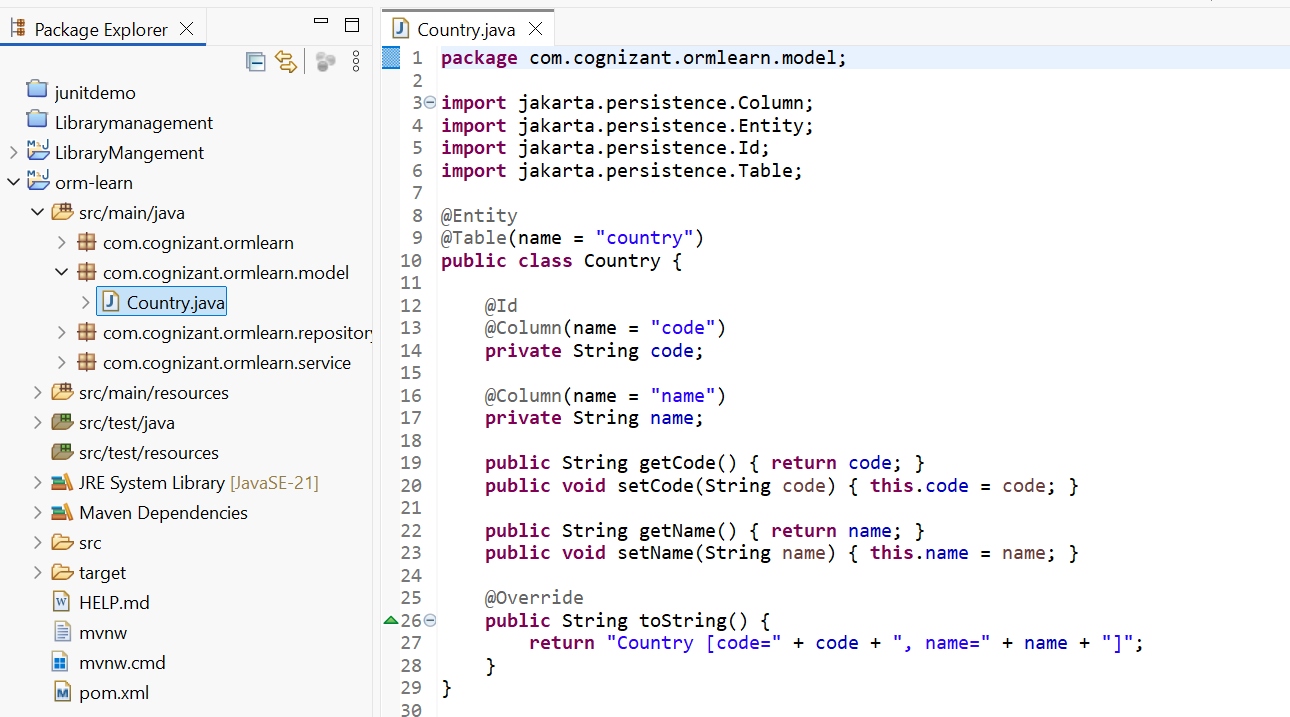
Step 6: Add Logging in OrmLearnApplication.java



Step 7: Create Country Table and Insert Data.



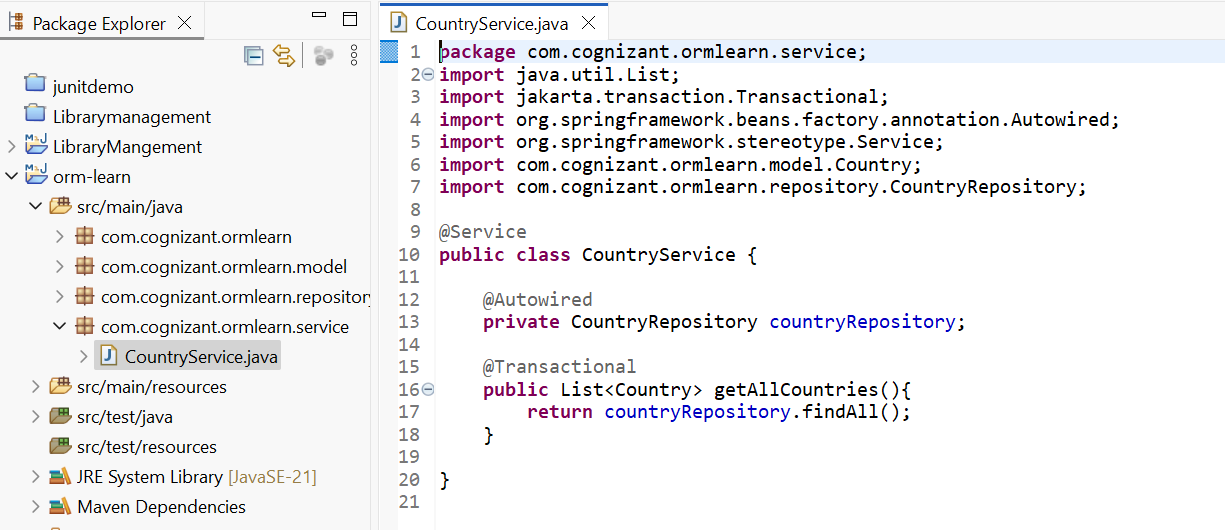
Step 8: Create Country Entity Class



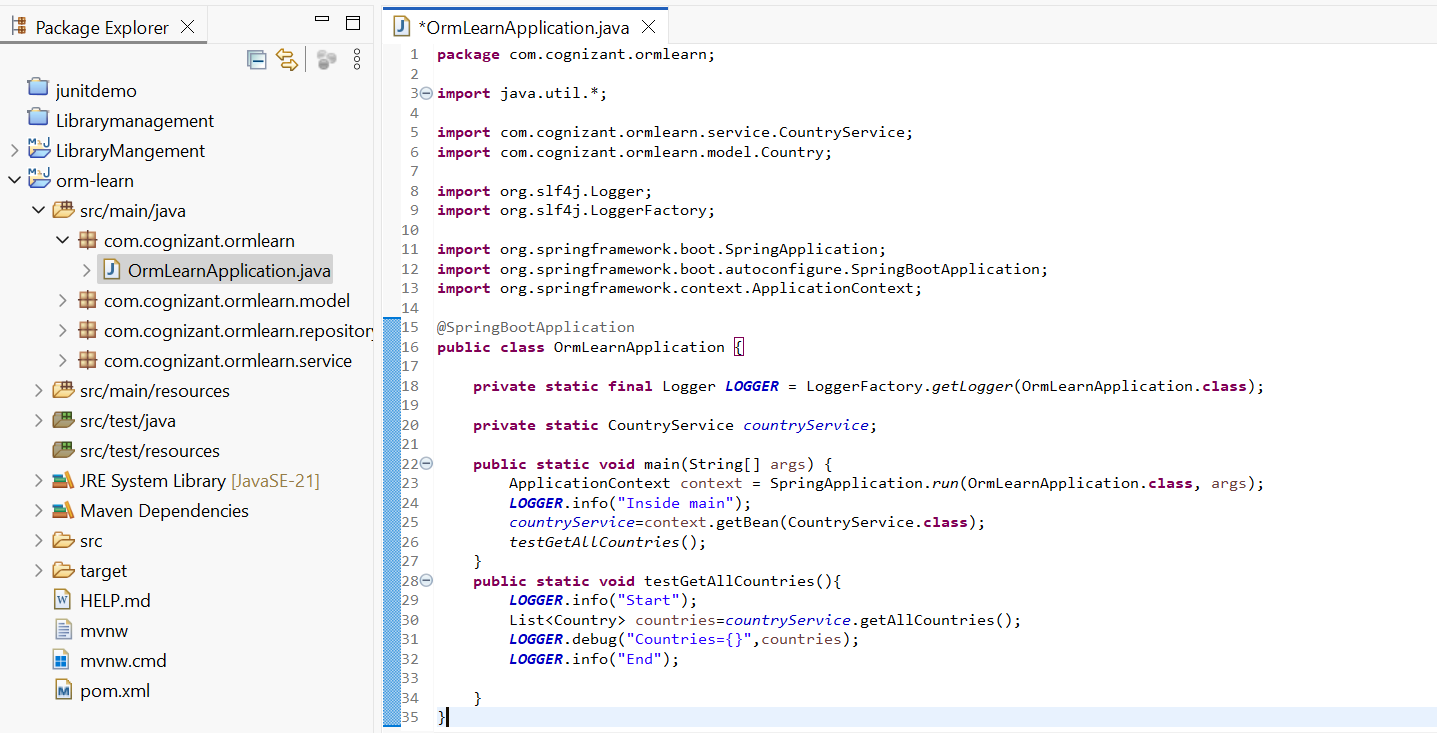
Step 9: Create CountryRepository Interface



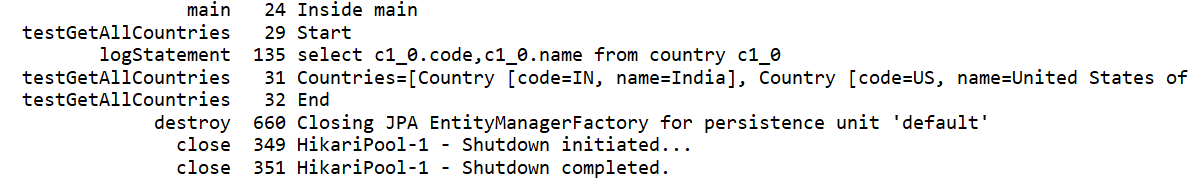
Step 10: Create CountryService Class



Step 11: Modify Main Class for Testing



Step 12: Run and Verify Output



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

Java Persistence API (JPA)

* JSR 338 Specification for persisting, reading and managing data from Java objects
* Does not contain concrete implementation of the specification
* Hibernate is one of the implementations of JPA

Hibernate

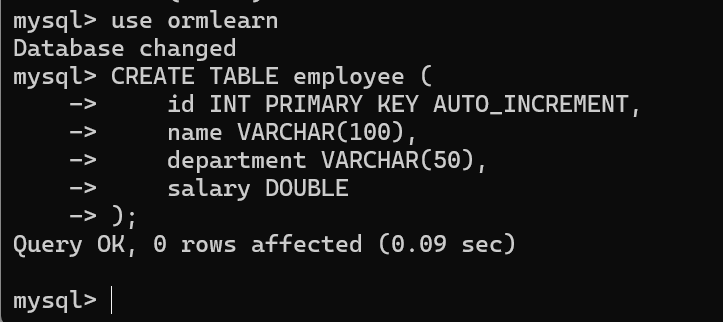
* ORM Tool that implements JPA

Spring Data JPA

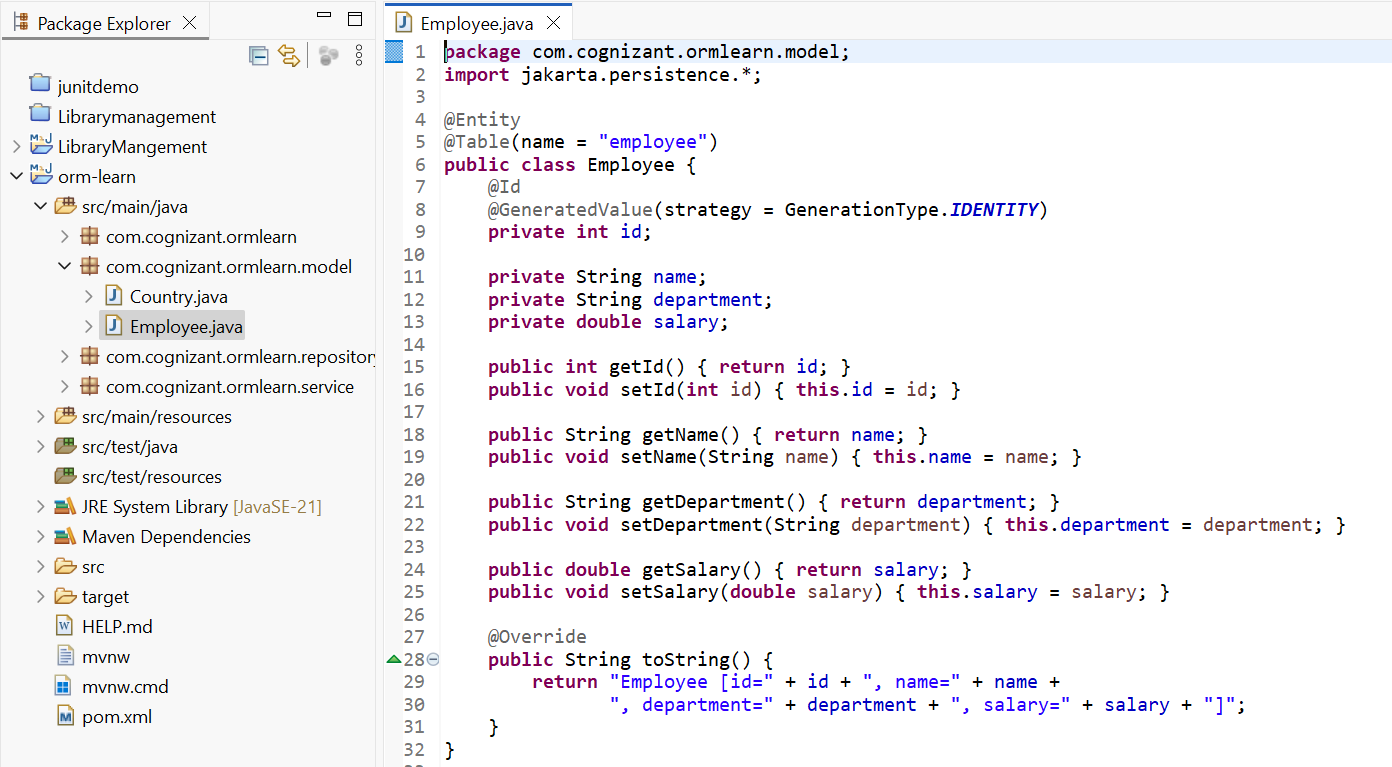
* Does not have JPA implementation, but reduces boiler plate code
* This is another level of abstraction over JPA implementation provider like Hibernate
* Manages transactions

**Code:**

Step 1: Create the employee Table in MySQL



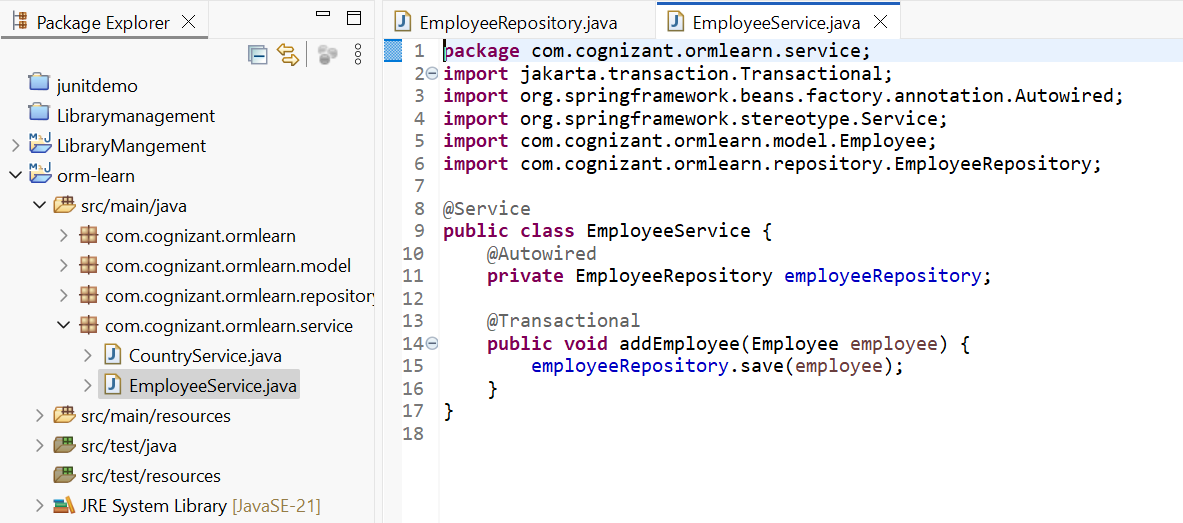
Step 2: Create the Employee Entity Class



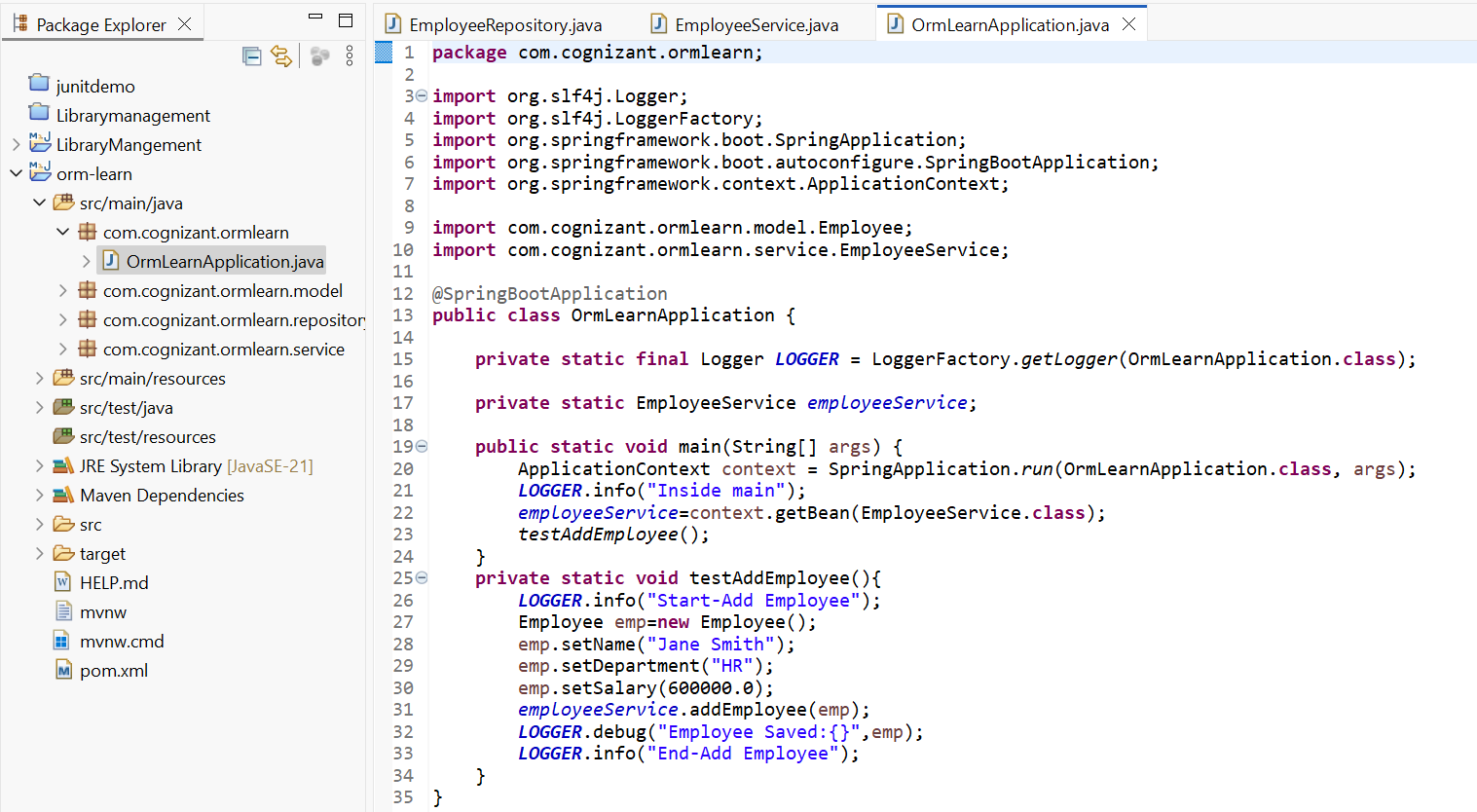
Step 3: Create the EmployeeRepository Interface



Step 4: Create the EmployeeService Class



Step 5: Add Testing Code in OrmLearnApplication.java



**Output**:

