**Hands on 1**

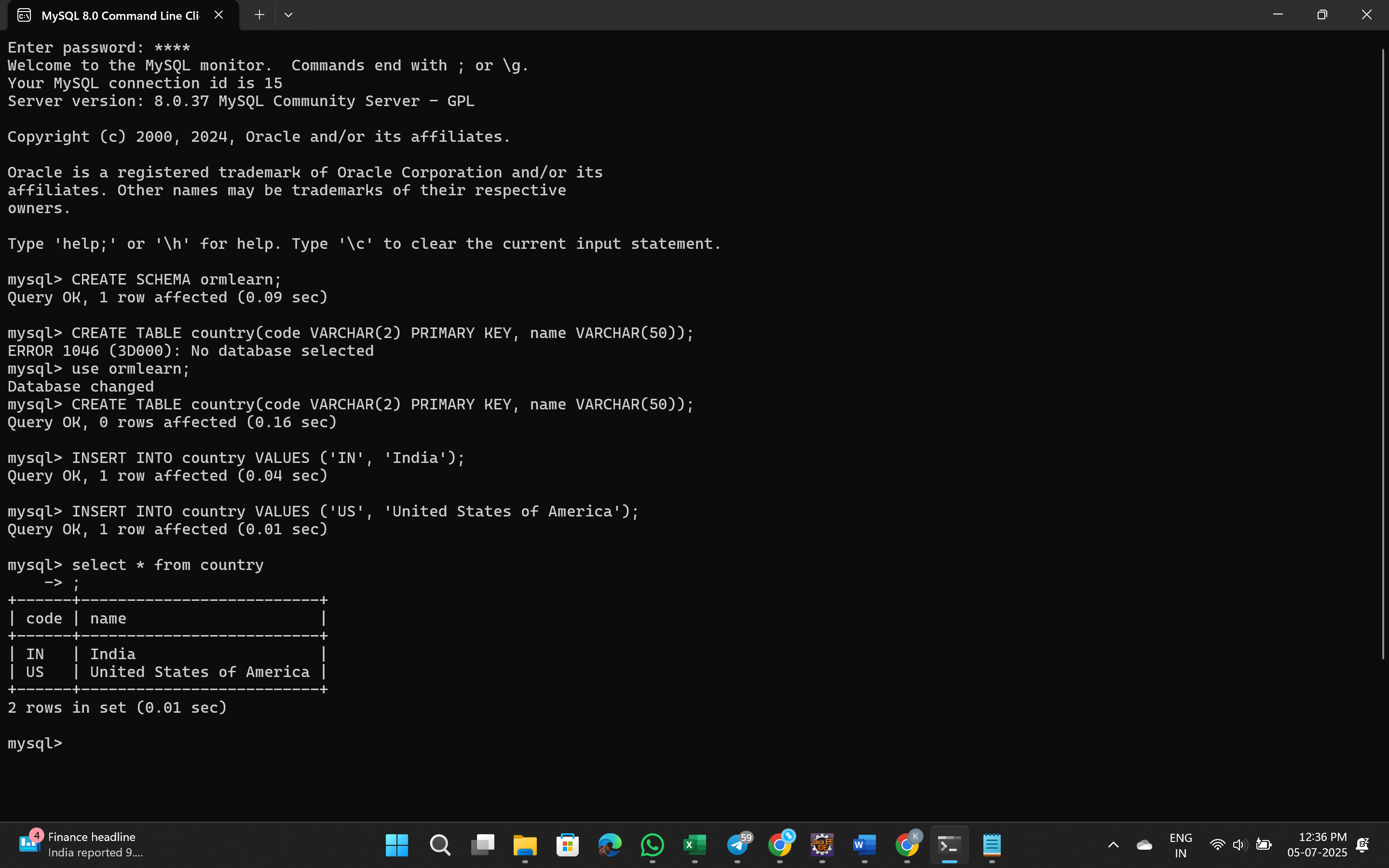
**Spring Data JPA - Quick Example**

A screenshot of a computer

AI-generated content may be incorrect.

A computer screen shot of a computer

AI-generated content may be incorrect.



A computer screen shot of a program

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

A computer screen shot of a black screen

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

OrmLearnApplication.java

package com.cognizant.orm\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.context.ApplicationContext;

import com.cognizant.orm\_learn.model.Country;

import com.cognizant.orm\_learn.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();

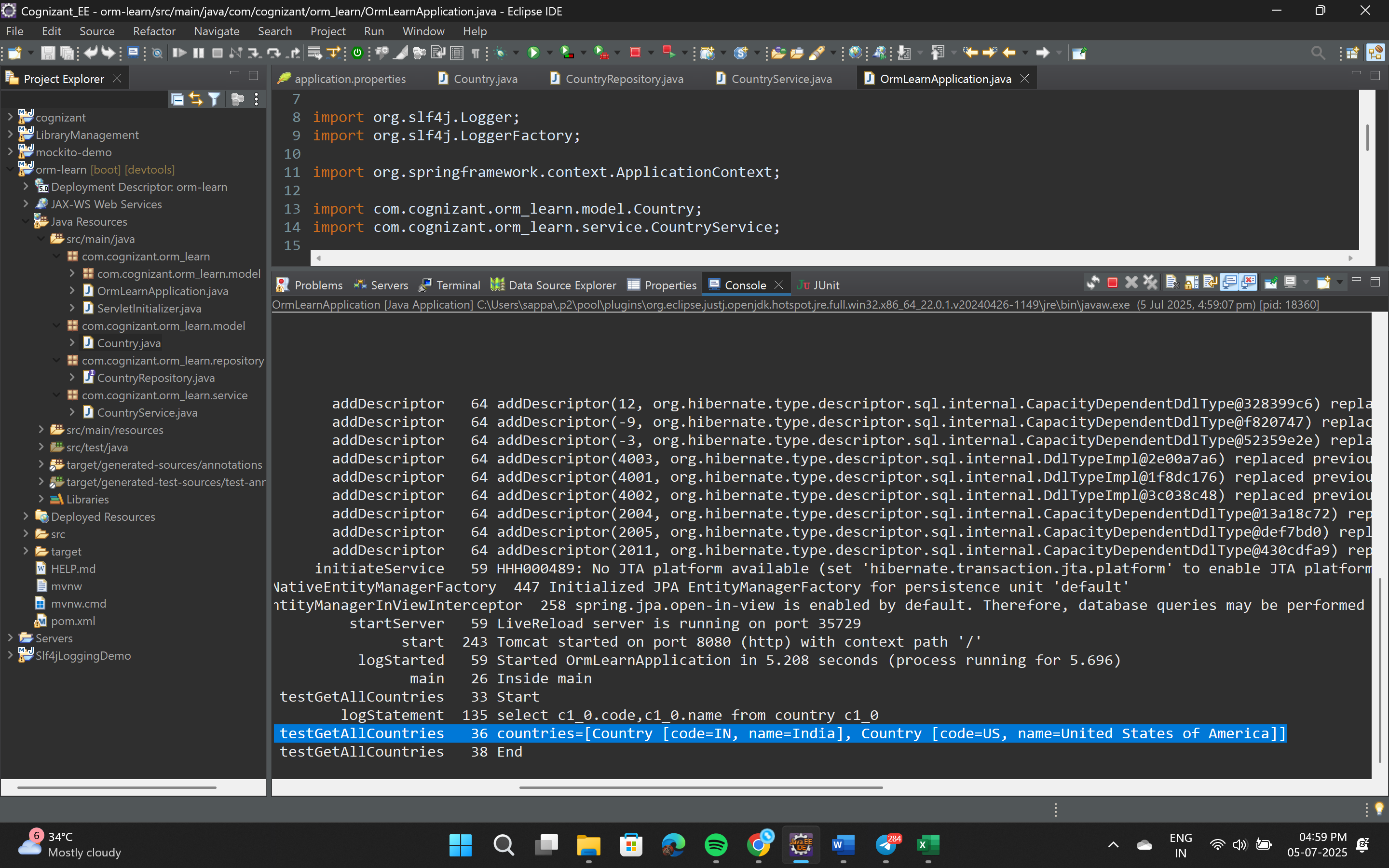
***LOGGER***.debug("countries={}", countries);

***LOGGER***.info("End");

}

}

**Output:**



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

| **Feature** | **JPA** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| **Type** | Specification | JPA Implementation | Abstraction over JPA |
| **Defined by** | Java EE (JSR 338) | Red Hat | Spring Framework |
| **Purpose** | Define standard ORM behavior | ORM engine with advanced features | Simplify JPA usage in Spring apps |
| **Implements** | Interface only, no implementation | Yes, implements JPA | Uses JPA provider (like Hibernate) |
| **Boilerplate Code** | Requires entity manager & queries | Still has manual session & transaction handling | Reduces all to interfaces and annotations |
| **Transaction Management** | Requires manual configuration | Partially automatic | Fully integrated with Spring's @Transactional |
| **Query Writing** | JPQL, Criteria API | JPQL, HQL | JPQL + Derived Queries + @Query support |
| **Learning Curve** | Medium | Higher (lots of APIs) | Easier, quick development |