Hands on 4

Difference between JPA, Hibernate and Spring Data JPA ….

JPA vs Hibernate vs Spring Data JPA:

1️⃣ JPA (Java Persistence API)

* It's like a set of rules/standard for how Java objects are saved to the database.
* JPA itself doesn't do the work — it just defines how it should be done.
* Think of it like the blueprint, not the builder.

Example:

@Entity

public class Employee {

@Id

private int id;

private String name;

}

2️⃣ Hibernate

* Hibernate is the most popular implementation of JPA.
* It actually does the work of converting Java objects to database rows (ORM - Object Relational Mapping).
* Without Spring, you have to manually handle sessions, transactions, etc.

Example using Hibernate:

Session session = factory.openSession();

Transaction tx = session.beginTransaction();

session.save(employee);

tx.commit();

session.close();

3️⃣ Spring Data JPA

* Spring makes life easy by reducing the boilerplate code.
* It provides ready-made repositories, so you don't write queries or session code.
* Built on top of JPA and works with Hibernate internally.
* Just define an interface, Spring does the rest!

Example using Spring Data JPA:

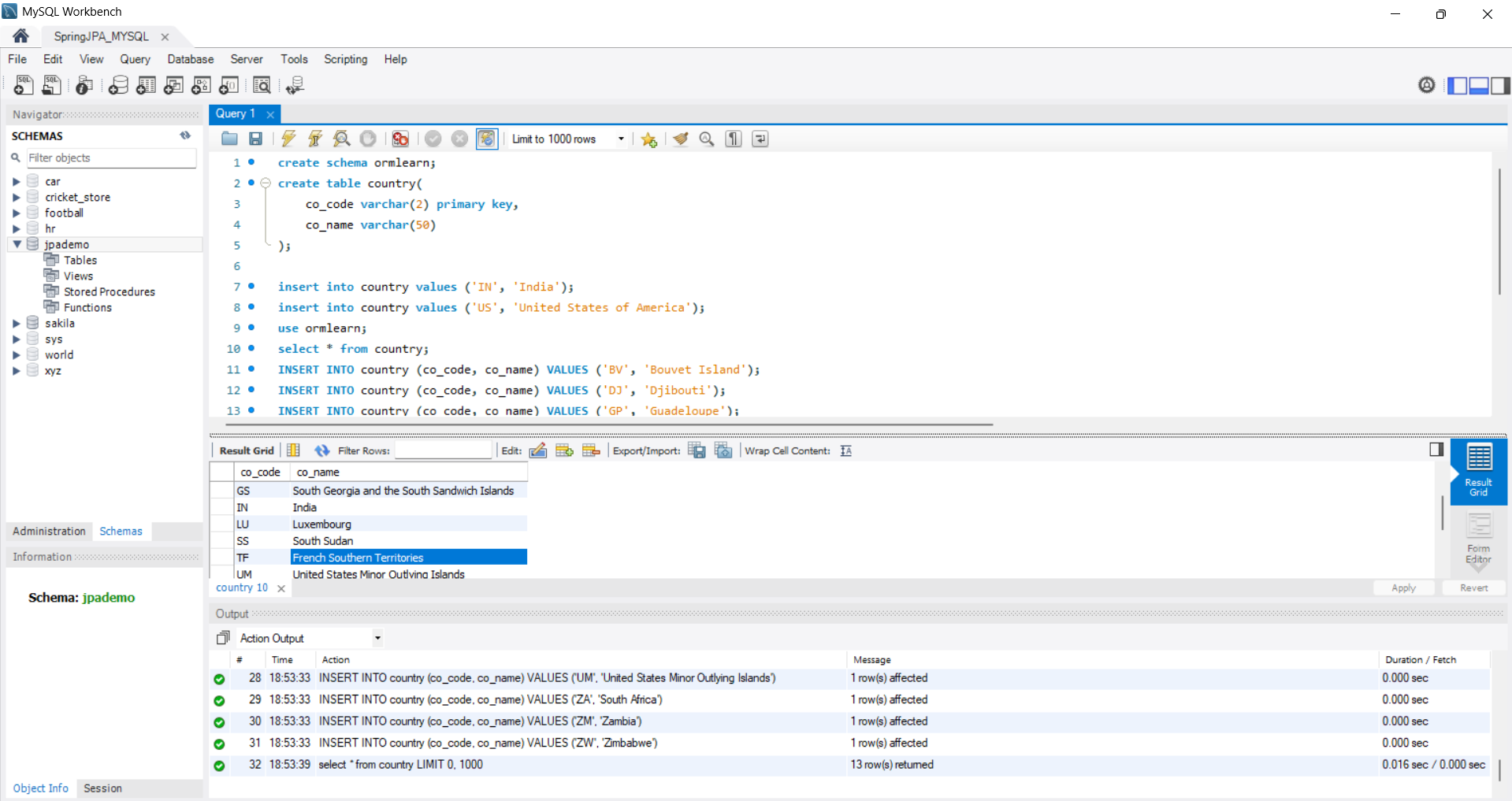
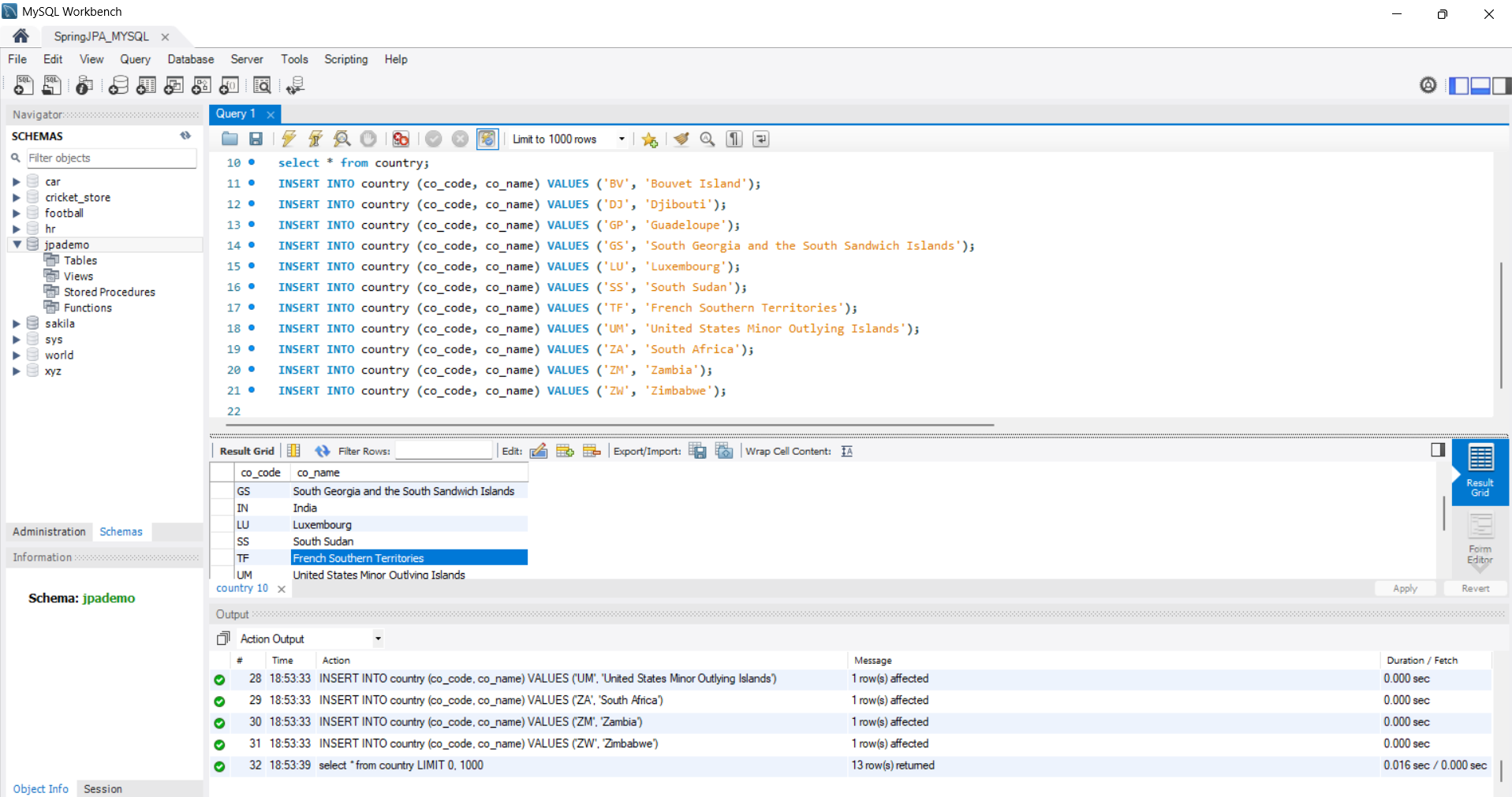
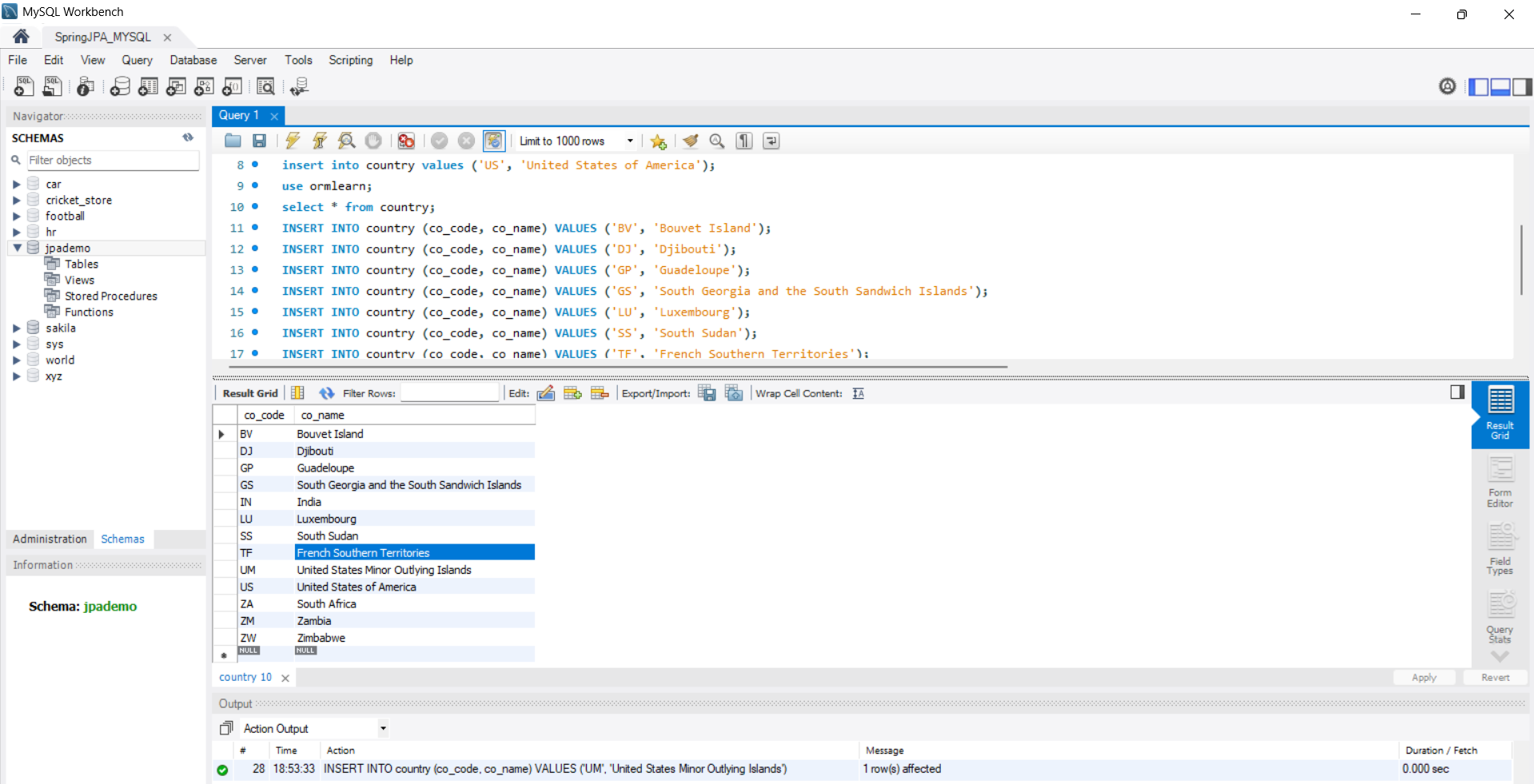
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {}

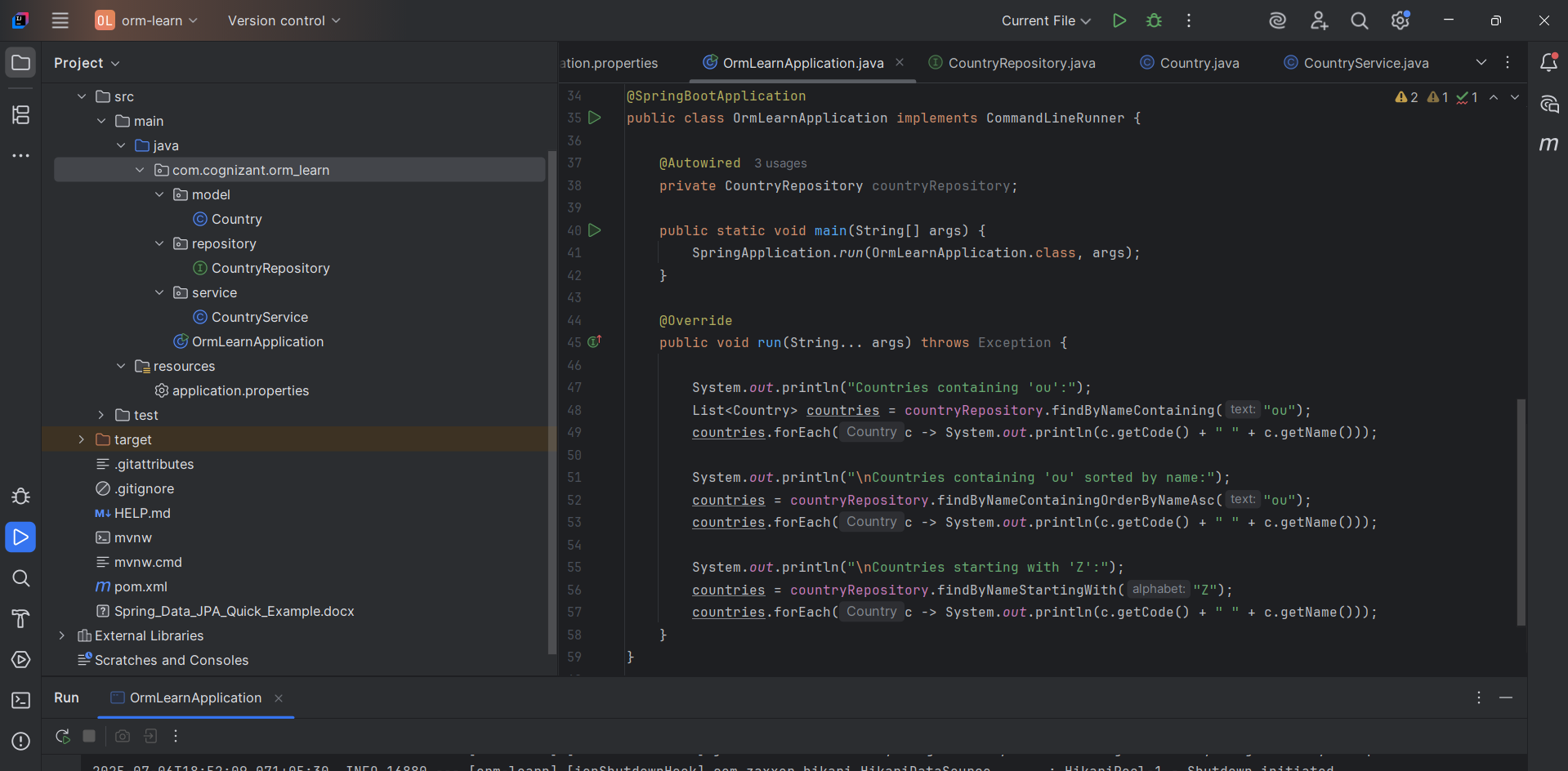
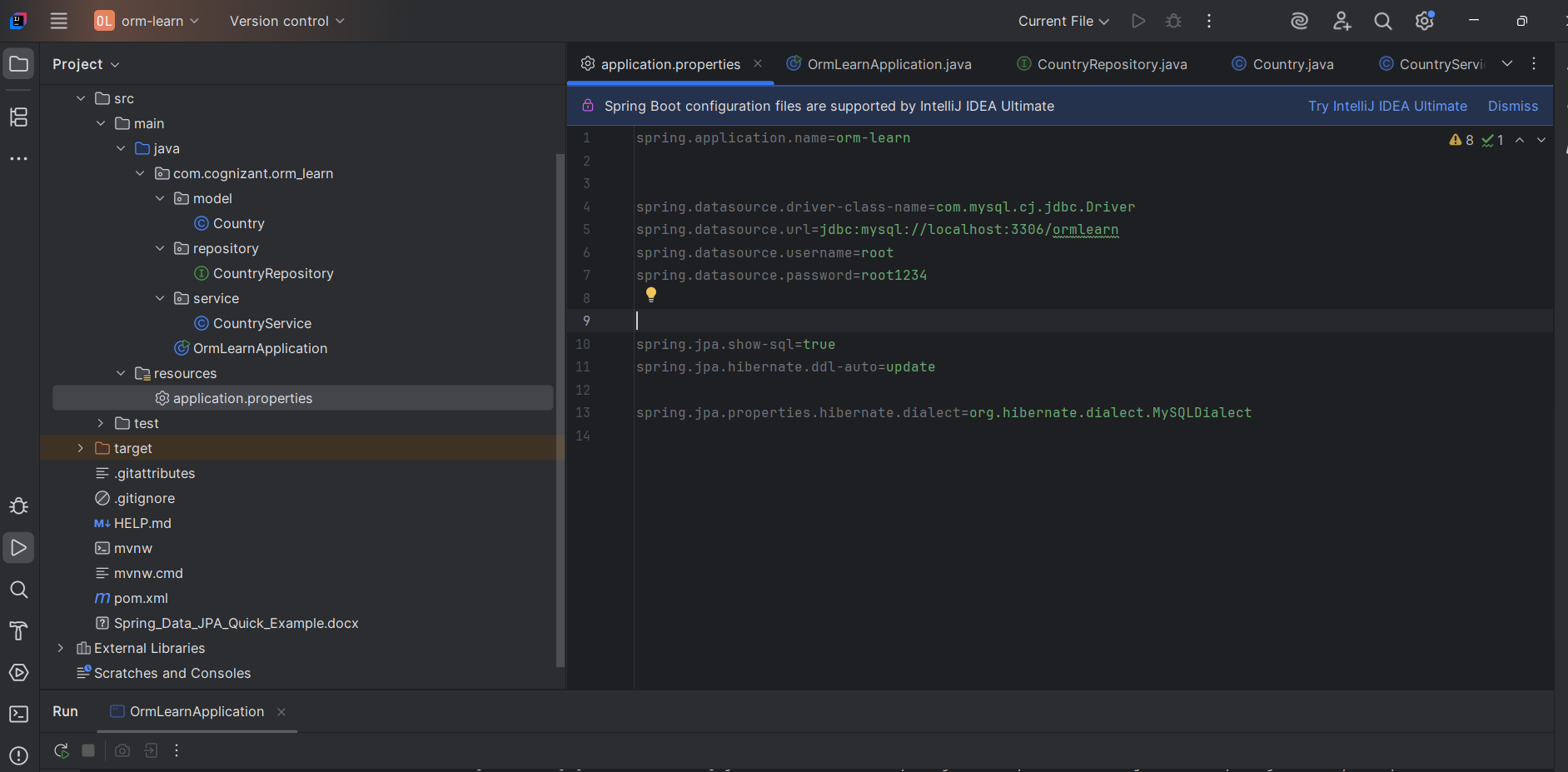
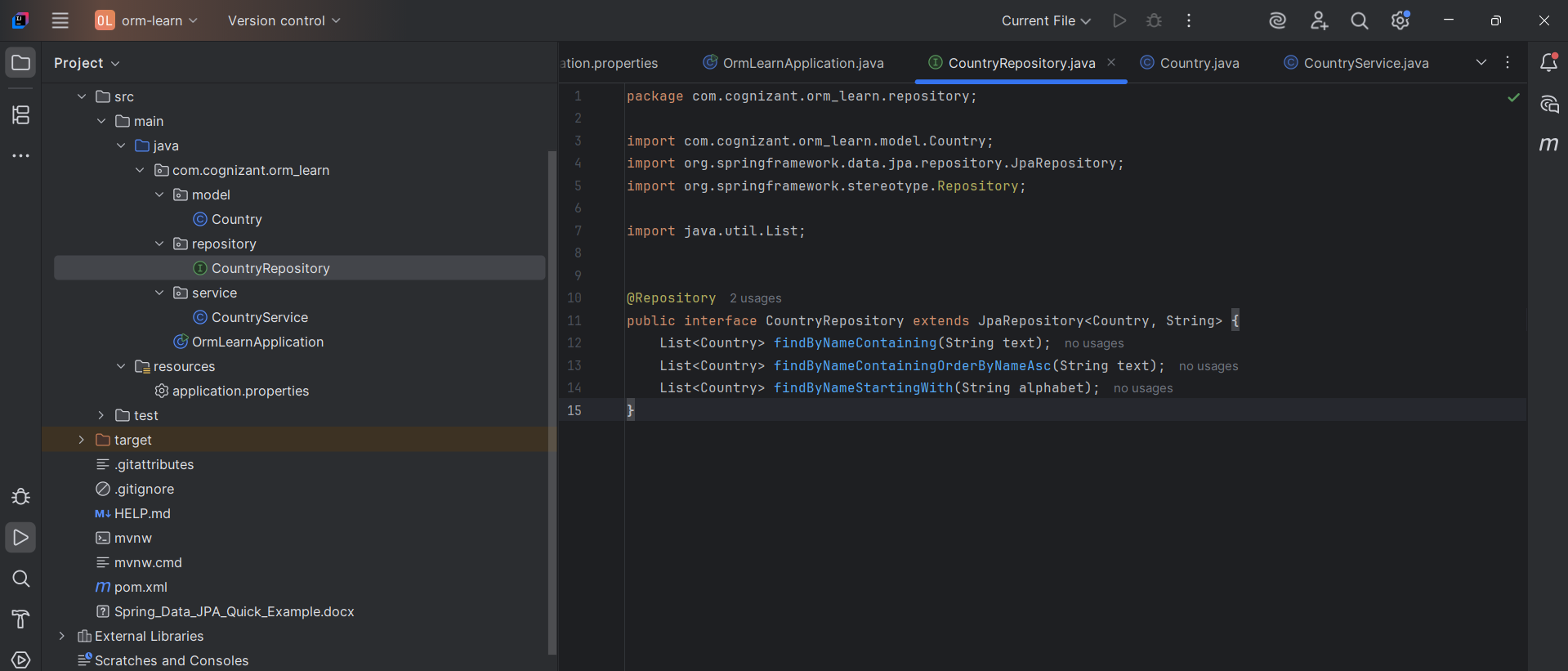
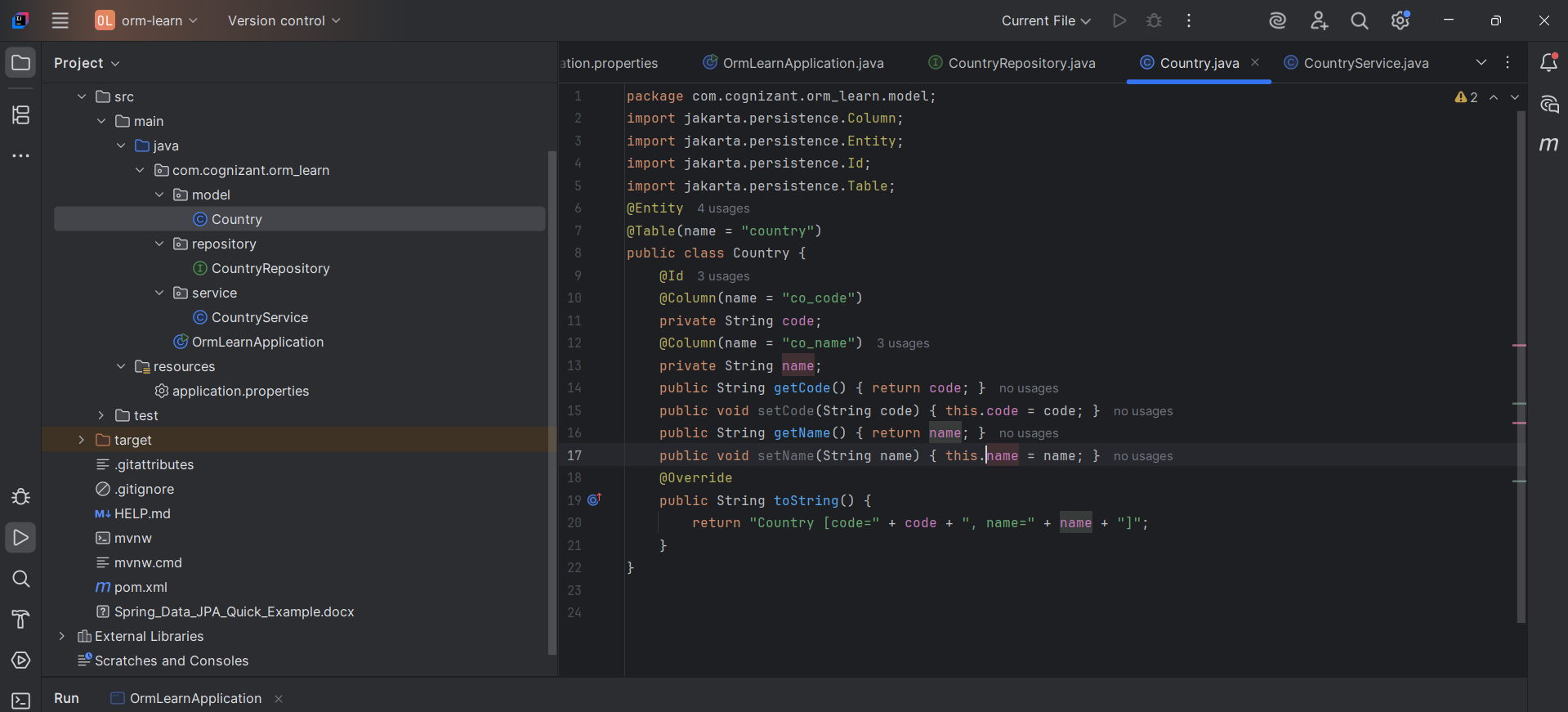
@Autowired

EmployeeRepository repo;

repo.save(employee);

**2. spring-data-jpa-handson**

**Demonstrate implementation of Query Methods feature of Spring Data JPA**   

**Hands on 2**

**Write queries on stock table using Query Methods**   
  
