**Spring Data JPA with Spring Boot, Hibernate**

**Spring Data JPA - Quick Example**

**Step 1: Spring Boot Project Setup**

* Created Spring Boot project
* Group: com.cognizant
* Artifact: orm-learn
* Dependencies added:
  + Spring Boot DevTools
  + Spring Data JPA
  + MySQL Driver
* Project imported into Eclipse as a Maven project

**Step 2: MySQL Schema and Table Setup**

* Created schema: ormlearn
* Created table:
* create table country (
* co\_code varchar(2) primary key,
* co\_name varchar(50)
* );
* Inserted records:

insert into country values ('IN', 'India');

insert into country values ('US', 'United States of America');

**Step 3: application.properties Configuration**

Configured database, dialect, logging:

properties

Copy code

spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn

spring.datasource.username=root

spring.datasource.password=root

spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

logging.level.org.springframework=info

logging.level.com.cognizant=debug

logging.level.org.hibernate.SQL=trace

**Step 4: Entity Class – Country.java**

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

}

Step 6: Service Layer – CountryService.java

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

**}**

**Step 7: Main Class – OrmLearnApplication.java**

@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);

testGetAllCountries();

} private static void testGetAllCountries() {

LOGGER.info("Start");

List<Country> countries = countryService.getAllCountries();

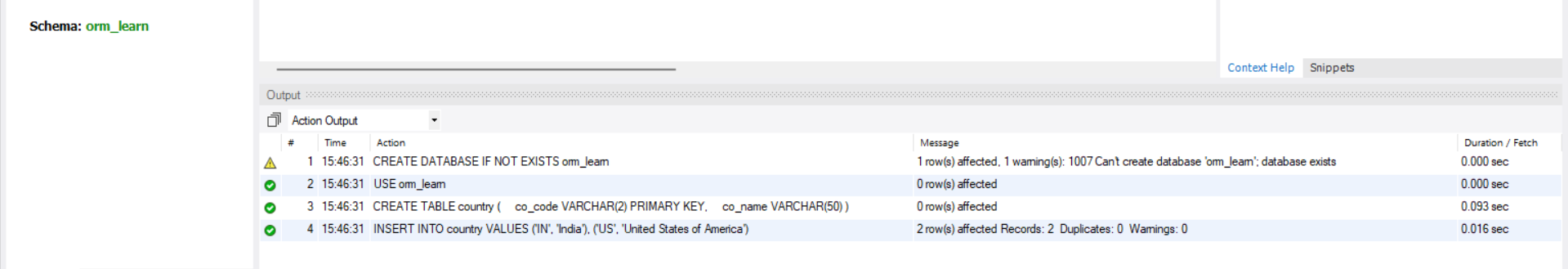
LOGGER.debug("countries={}", countries);

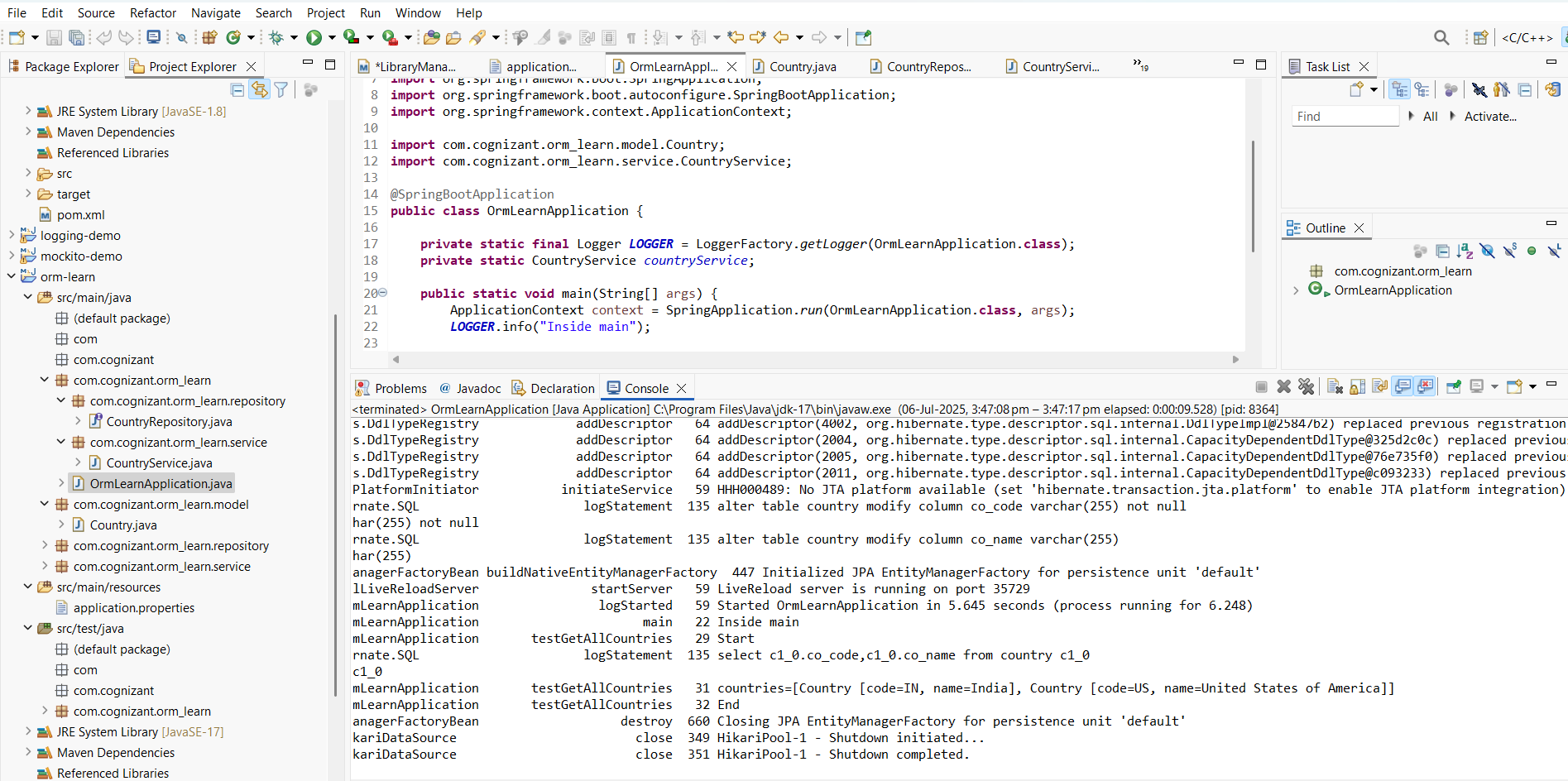
LOGGER.info("End");

}

}

**Output**

****

****

06-07-25 15:47:17.101 restartedMain DEBUG c.c.o.OrmLearnApplication testGetAllCountries countries=[Country [code=IN, name=India], Country [code=US, name=United States of America]]

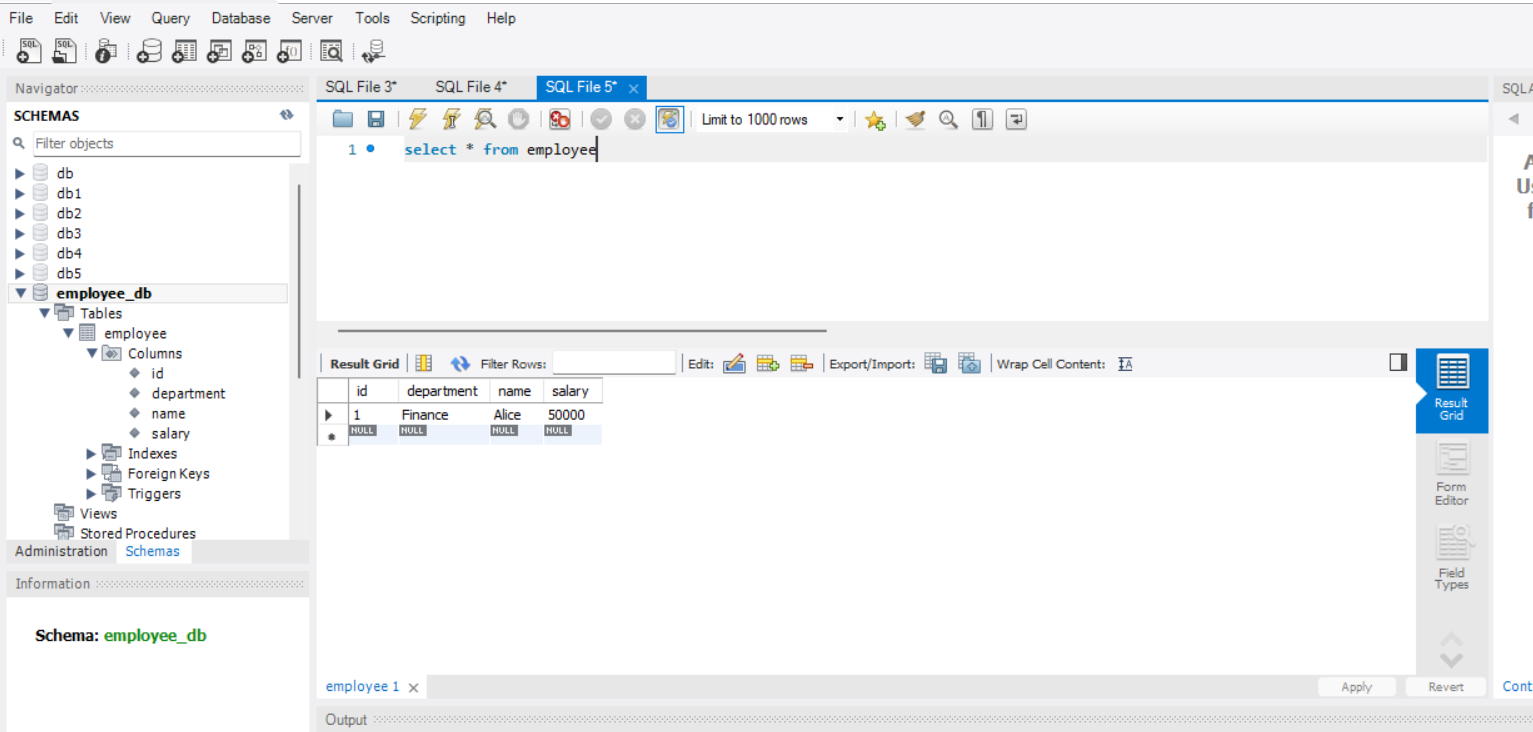
06-07-25 15:47:17.102 restartedMain INFO c.c.o.OrmLearnApplication testGetAllCountries

**4: Difference Between JPA, Hibernate, and Spring Data JPA**

**🔹 1. Project Setup**

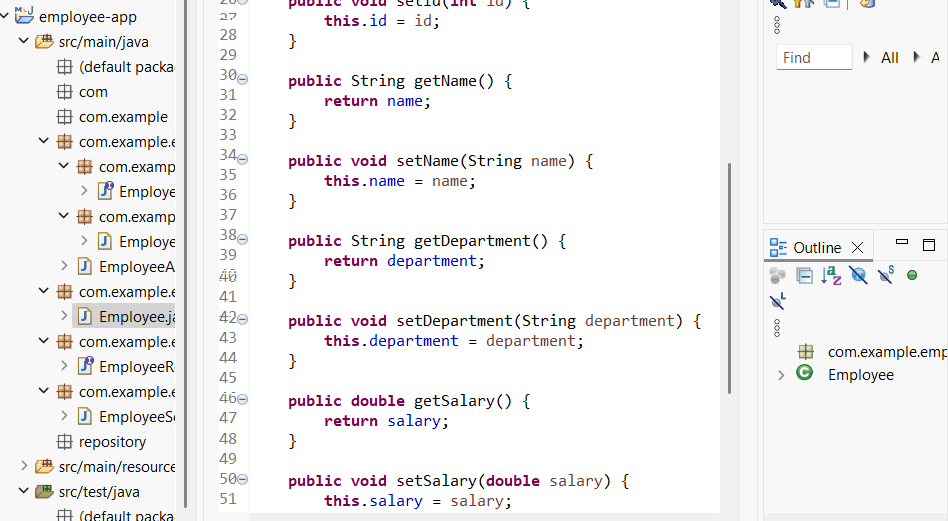
* Created a Spring Boot project employee-app using **Spring Initializr**
* Dependencies added: Spring Data JPA, MySQL Driver, Spring Boot DevTools
* Project imported into Eclipse
* application.properties configured correctly with MySQL database settings
* Database connected: Console shows Hibernate created the table

**Step 2 :Database Setup**



**Step 3: Entity Class: Employee.java**

* Created in com.example.employeeapp.model
* Annotated with @Entity, @Table
* Fields: id, name, department, salary
* Getters, setters, and toString() implemented



**Step 4. Repository Interface: EmployeeRepository.java**

* Created in com.example.employeeapp.repository
* Interface extends JpaRepository<Employee, Integer>
* Annotated with @Repository

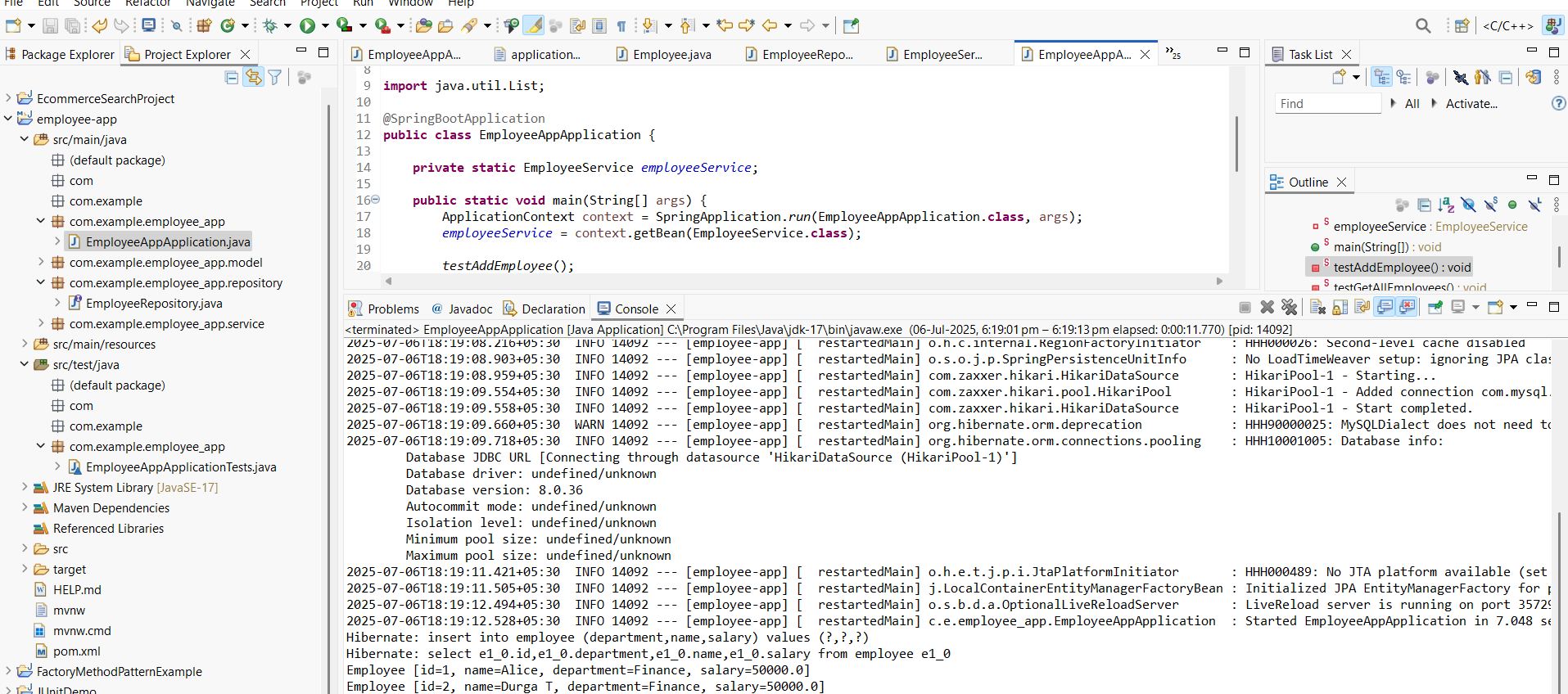
**Step 5 : Service Layer: EmployeeService.java**

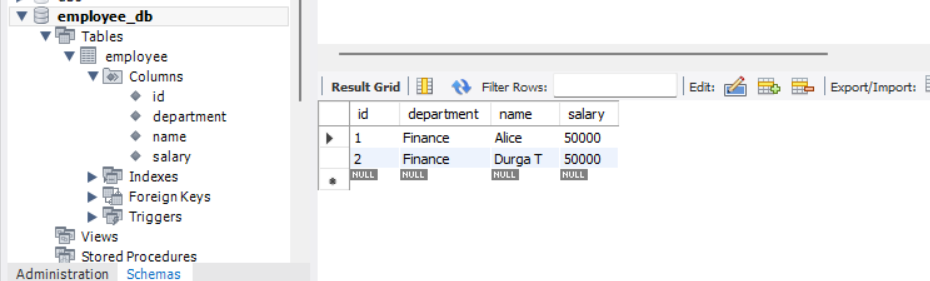
* Created in com.example.employeeapp.service
* Annotated with @Service
* Autowired EmployeeRepository
* Method addEmployee() uses employeeRepository.save()
* Method getAllEmployees() uses employeeRepository.findAll()

**Step 6. Application Class: EmployeeAppApplication.java**

* Main method contains Spring Boot run
* Logs confirm: Inside main
* Created employee (Alice) and displayed list of employees

**Output**

****

****