**📄 Exercise 1: Setting up Spring Boot with JPA and Retrieving Country Data**

**1. Problem Statement**

Set up a Spring Boot application with JPA and MySQL integration to fetch records from a country table using service and repository layers.

**2. Steps**

* Created a MySQL table country with columns code and name, and inserted sample records.
* Created Country model class with JPA annotations.
* Created CountryRepository interface extending JpaRepository.
* Implemented CountryService to fetch all countries.
* Modified OrmLearnApplication to call getAllCountries() using service layer.

**3. Codes**

**country Table Creation (MySQL):**

CREATE TABLE country (

co\_code VARCHAR(2) PRIMARY KEY,

co\_name VARCHAR(50)

);

INSERT INTO country VALUES ('IN', 'India');

INSERT INTO country VALUES ('US', 'United States of America');

**Country.java**

@Entity

@Table(name = "country")

public class Country {

@Id

@Column(name = "co\_code")

private String code;

@Column(name = "co\_name")

private String name;

// Getters, setters, toString()

}

**CountryRepository.java**

@Repository

public interface CountryRepository extends JpaRepository<Country, String> {

}

**CountryService.java**

@Service

public class CountryService {

@Autowired

private CountryRepository countryRepository;

@Transactional

public List<Country> getAllCountries() {

return countryRepository.findAll();

}

}

**OrmLearnApplication.java**

@SpringBootApplication

public class OrmLearnApplication {

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();

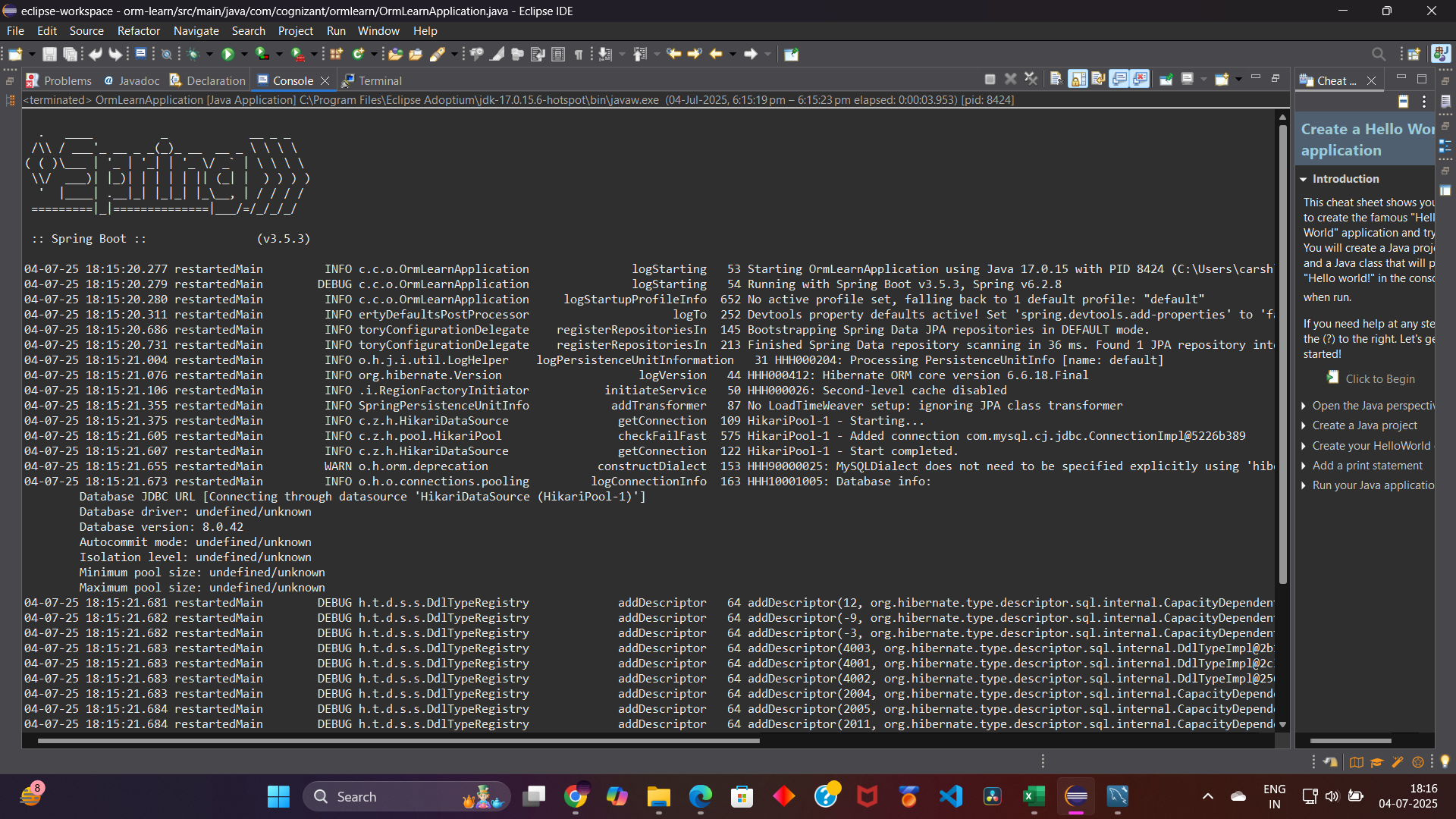
LOGGER.debug("countries={}", countries);

LOGGER.info("End");

}

}

**4. Output:**

****

DEBUG countries=[Country [code=IN, name=India], Country [code=US, name=United States of America]]

**📄 Hands-On 4: Difference between Hibernate and Spring Data JPA – Insert Employee Record**

**1. Problem Statement**

Compare Hibernate and Spring Data JPA by implementing an employee creation scenario using Spring Data JPA and verify the insert operation using MySQL.

**2. Steps**

* Created employee table in MySQL.
* Created Employee model class.
* Created EmployeeRepository interface.
* Implemented EmployeeService to save employee record using Spring Data JPA.
* Called the service method from OrmLearnApplication to test insert functionality.

**3. Codes**

**employee Table Creation (MySQL):**

CREATE TABLE employee (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

department VARCHAR(100),

salary DOUBLE

);

**Employee.java**

@Entity

@Table(name = "employee")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String department;

private double salary;

// Getters, setters, toString()

}

**EmployeeRepository.java**

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**EmployeeService.java**

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}

}

**OrmLearnApplication.java**

@SpringBootApplication

public class OrmLearnApplication {

private static EmployeeService employeeService;

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);

employeeService = context.getBean(EmployeeService.class);

testAddEmployee();

}

private static void testAddEmployee() {

LOGGER.info("Start testAddEmployee");

Employee emp = new Employee();

emp.setName("Alice");

emp.setDepartment("Tech");

emp.setSalary(75000.00);

employeeService.addEmployee(emp);

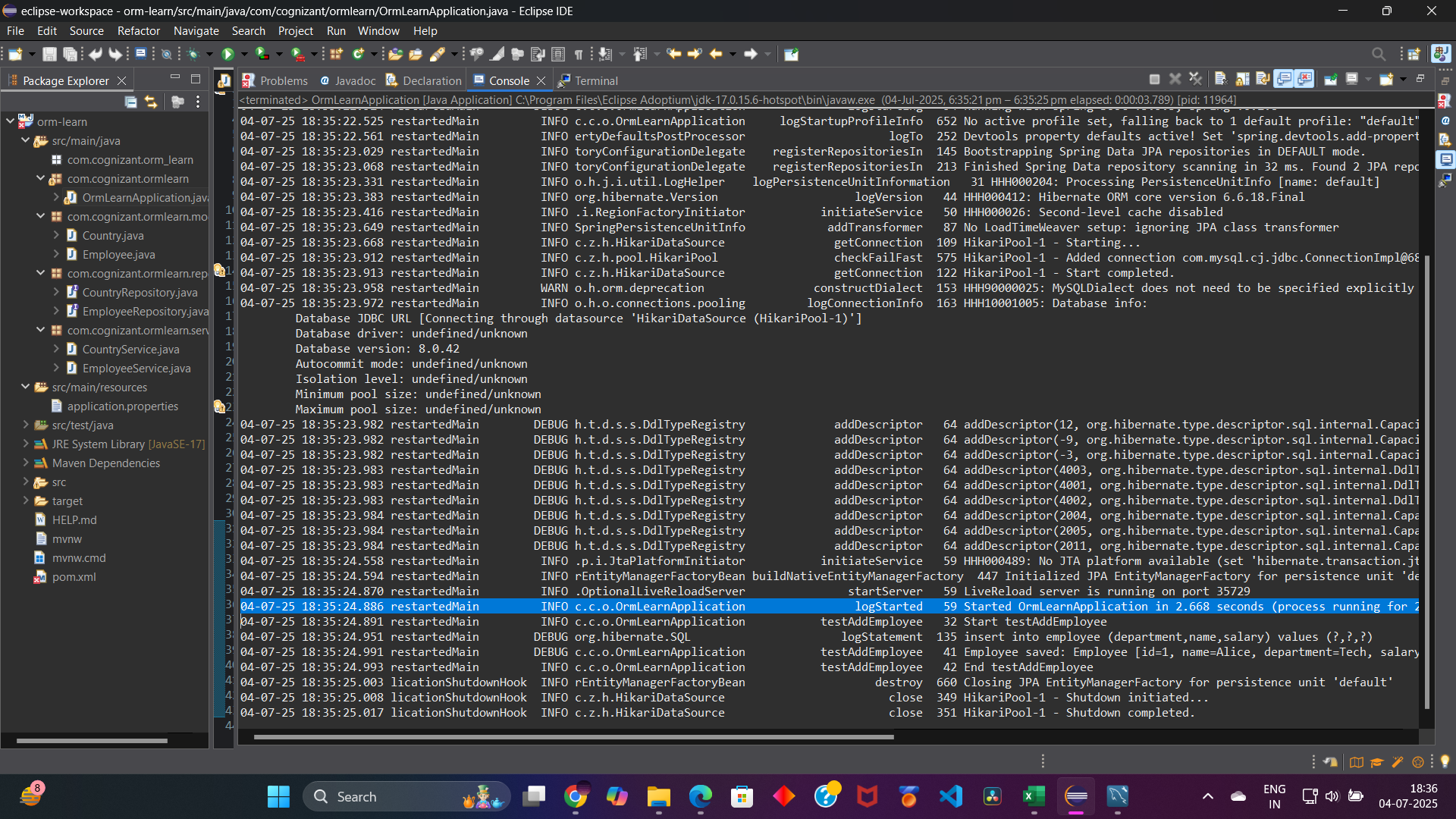
LOGGER.debug("Employee saved: {}", emp);

LOGGER.info("End testAddEmployee");

}

}

**4. Output:**

****

DEBUG org.hibernate.SQL logStatement insert into employee (department,name,salary) values (?,?,?)

DEBUG Employee saved: Employee [id=1, name=Alice, department=Tech, salary=75000.0]