**Implement services for managing Country**

**Find a country based on country code**

**Add a new country**

**Update a country based on code**

**Delete a country based on code**

**-> I have implemented all the above mentioned handson in single application which is as follows:**

**Country.class (POJO)**  
package com.cognizant.spring\_learn.model;

import jakarta.persistence.Entity;

import jakarta.persistence.Id;

import jakarta.persistence.Table;

@Entity

@Table(name="country")

public class Country {

@Id

private String coCode;

private String coName;

public Country() {}

public Country(String coCode, String coName) {

this.coCode = coCode;

this.coName = coName;

}

public String getCoCode() {

return coCode;

}

public void setCoCode(String coCode) {

this.coCode = coCode;

}

public String getCoName() {

return coName;

}

public void setCoName(String coName) {

this.coName = coName;

}

@Override

public String toString() {

return "Country [coCode=" + coCode + ", coName=" + coName + "]";

}

}

**CountryService.class**

package com.cognizant.spring\_learn.service;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

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

import com.cognizant.spring\_learn.repository.CountryRepository;

import com.cognizant.spring\_learn.service.exception.CountryNotFoundException;

import jakarta.transaction.Transactional;

@Service

public class CountryService {

@Autowired

private CountryRepository countryrepository;

@Transactional

public Country findCountryByCode(String countryCode) throws CountryNotFoundException{

Optional<Country> result=countryrepository.findById(countryCode);

if(!result.isPresent())

throw new CountryNotFoundException("country not found with code: "+countryCode);

return result.get();

}

@Transactional

public void addCountry(Country country) {

countryrepository.save(country);

}

@Transactional

public void updateCountry(String co\_code,String co\_name) throws CountryNotFoundException{

Optional<Country> result=countryrepository.findById(co\_code);

if(!result.isPresent())

throw new CountryNotFoundException("couldnt update");

Country c=result.get();

c.setCoName(co\_name);

countryrepository.save(c);

}

@Transactional

public void deleteCountry(String code) {

countryrepository.deleteById(code);

}

}

**CountryNotFoundException**.**class**

package com.cognizant.spring\_learn.service.exception;

public class CountryNotFoundException extends Exception{

public CountryNotFoundException(String message) {

super(message);

}

}

**CountryRepository.class**

package com.cognizant.spring\_learn.repository;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

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

@Repository

public interface CountryRepository extends JpaRepository<Country,String> {

}

**SpringLearnApplication**.**class**

package com.cognizant.spring\_learn;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

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

import com.cognizant.spring\_learn.service.CountryService;

import com.cognizant.spring\_learn.service.exception.CountryNotFoundException;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@SpringBootApplication

public class SpringLearnApplication {

private static final Logger *log*=LoggerFactory.*getLogger*(SpringLearnApplication.class);

private static CountryService *countryservice*;

public static void main(String[] args) throws Exception{

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

*countryservice*=context.getBean(CountryService.class);

*getAllCountriesTest*();

*addCountryTest*();

*updateCountryTest*();

*deleteCountryTest*();

}

private static void getAllCountriesTest() throws CountryNotFoundException{

*log*.info("Start");

Country country = *countryservice*.findCountryByCode("IN");

*log*.debug("Country:{}", country);

*log*.info("End");

}

private static void addCountryTest() throws CountryNotFoundException{

*log*.info("Start addCountryTest");

*countryservice*.addCountry(new Country("AB", "ABland"));

*log*.debug("Added Country: {}", *countryservice*.findCountryByCode("AB"));

*log*.info("End addCountryTest");

}

private static void updateCountryTest() throws CountryNotFoundException {

*log*.info("Start updateCountryTest");

*countryservice*.updateCountry("AB", "ABResidence");

*log*.debug("Updated Country: {}", *countryservice*.findCountryByCode("AB"));

*log*.info("End updateCountryTest");

}

private static void deleteCountryTest() throws CountryNotFoundException {

*log*.info("Start deleteCountryTest");

*countryservice*.deleteCountry("XY");

try {

*countryservice*.findCountryByCode("XY");

} catch (CountryNotFoundException e) {

*log*.debug("Confirmed deletion: Country not found");

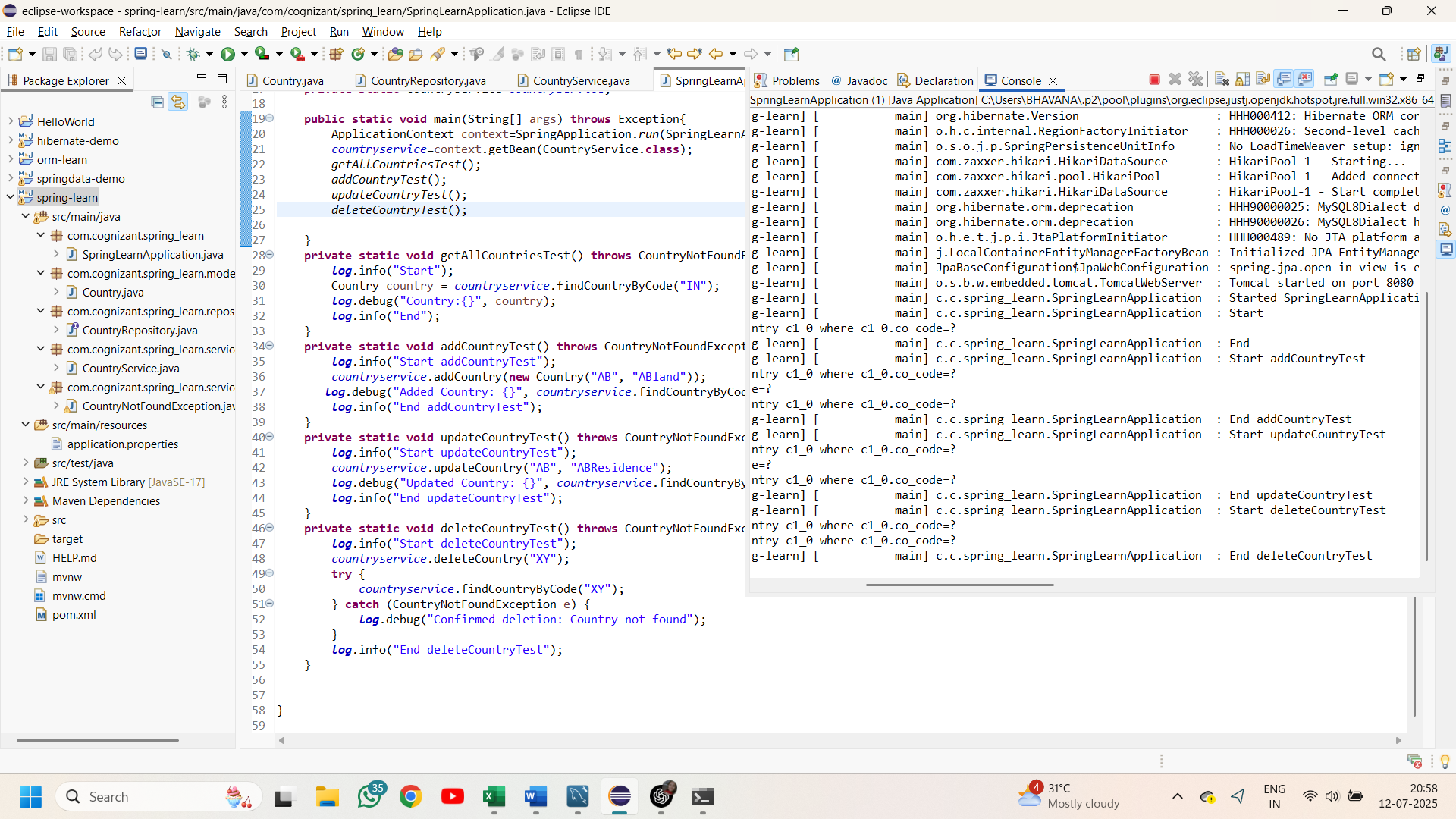
}

*log*.info("End deleteCountryTest");

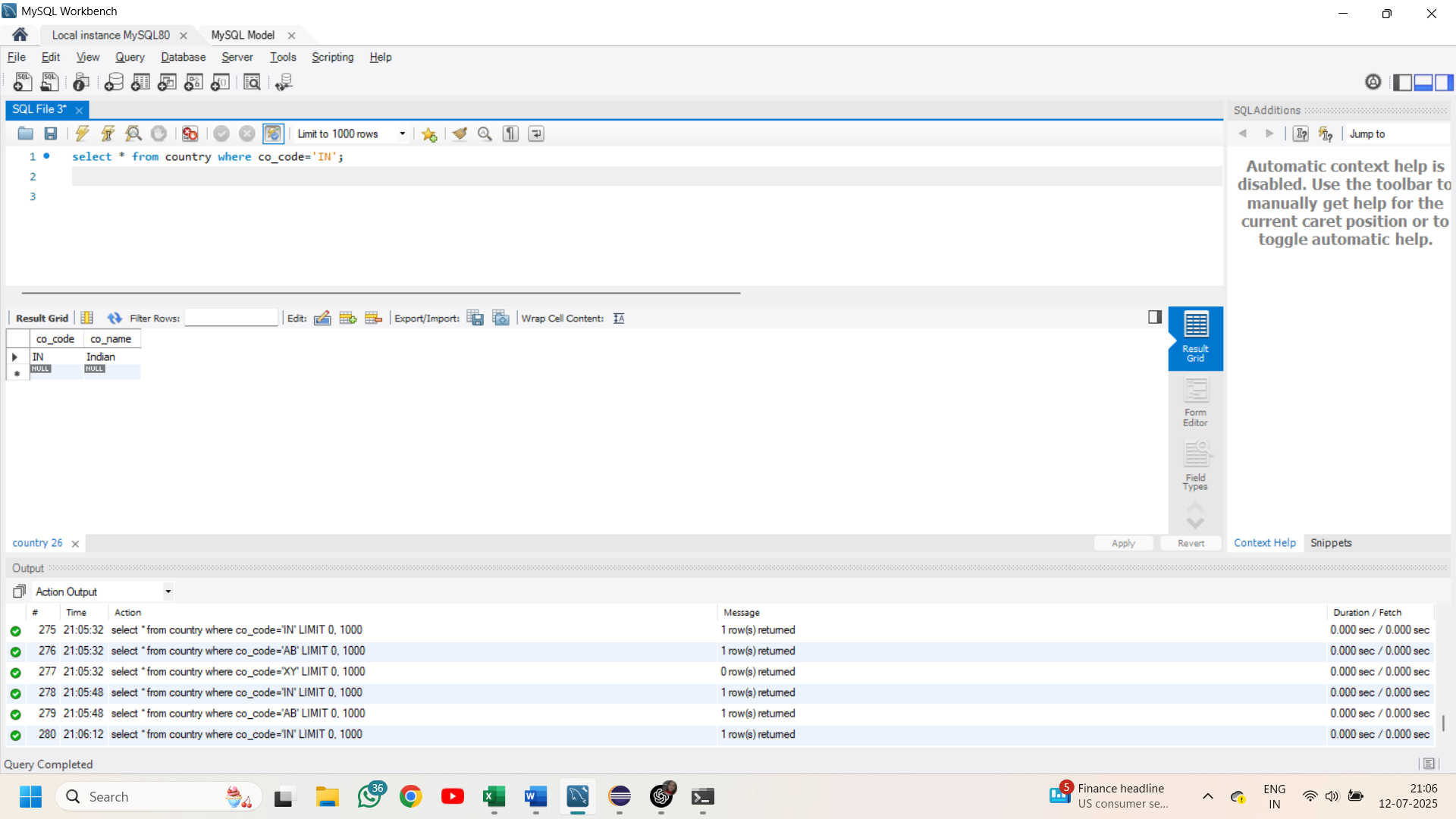
}

}

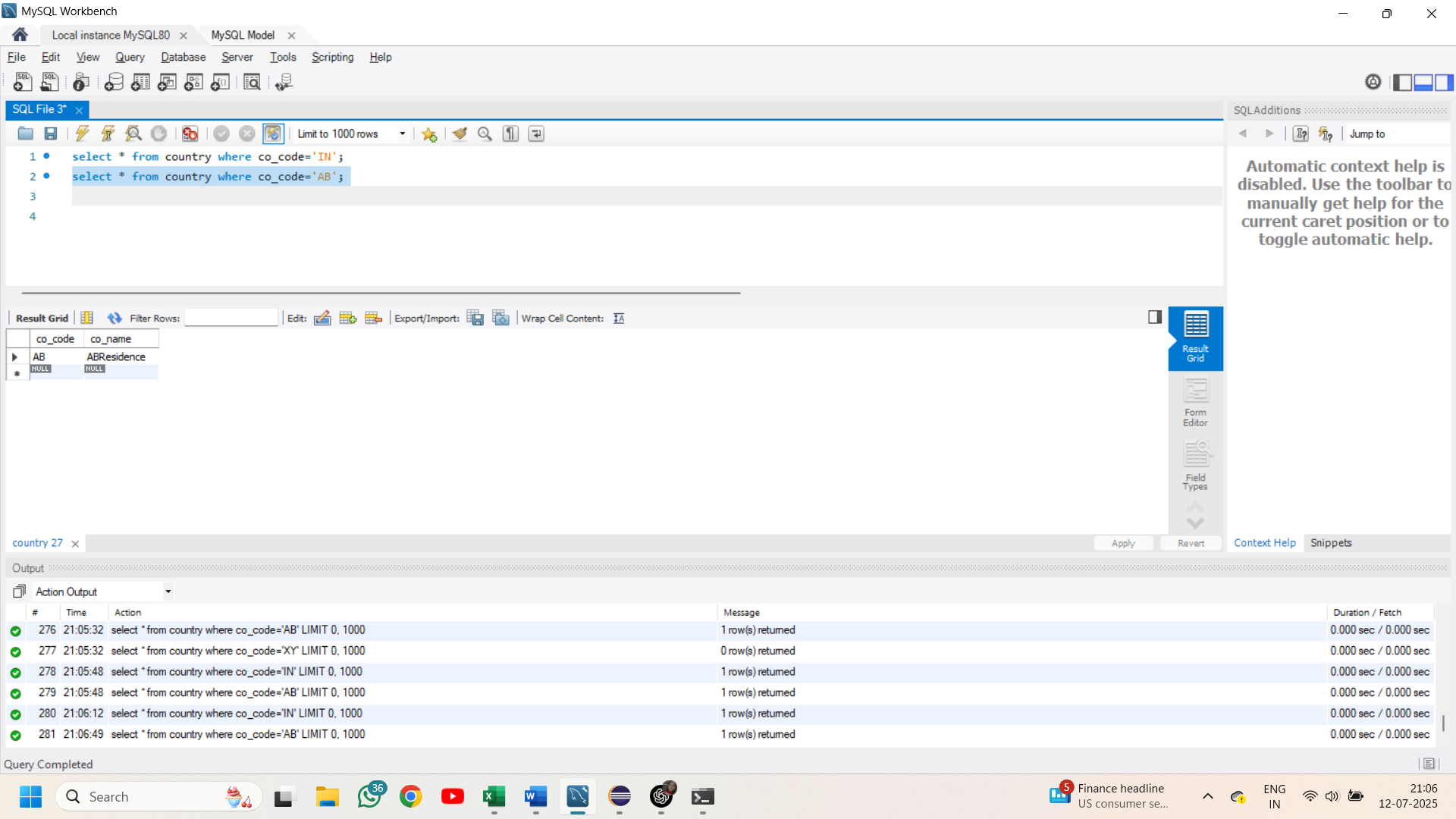
**Output**:

**  
MySql workbench:**

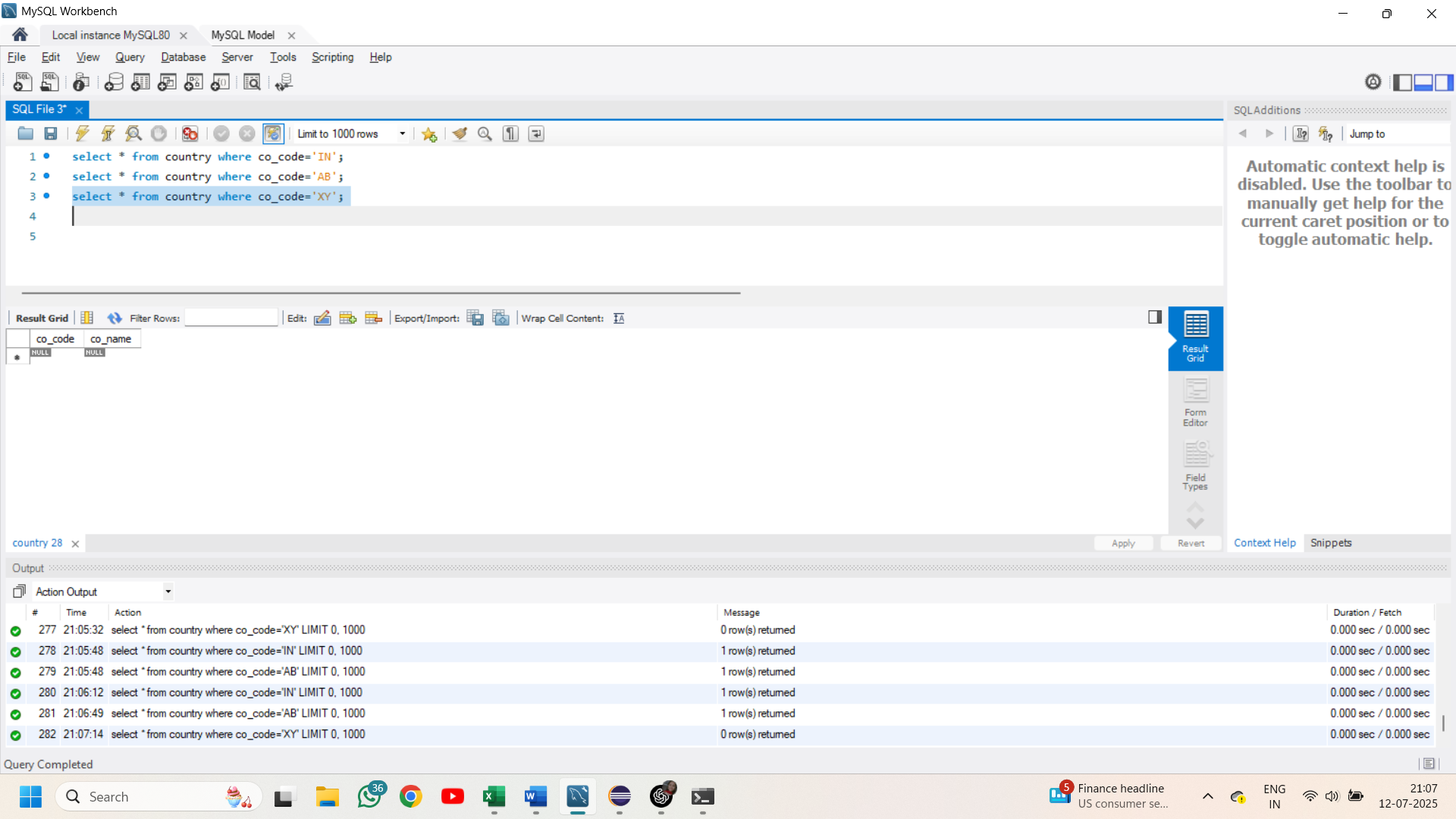
Finding a country based on its code:



Added and updated the country with same code:



Deleting a country that doesn’t exists in the table



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

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

**CountryRepository.class**

package com.cognizant.spring\_learn.repository;

import java.util.List;

import org.springframework.data.jpa.repository.JpaRepository;

import org.springframework.stereotype.Repository;

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

@Repository

public interface CountryRepository extends JpaRepository<Country,String> {

List<Country> findByCoNameContainingIgnoreCase(String coName);

List<Country> findByCoNameContainingIgnoreCaseOrderByCoNameAsc(String coName);

List<Country> findByCoNameStartingWithIgnoreCase(String prefix);

}

**SpringLearnApplication.class**

package com.cognizant.spring\_learn;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import com.cognizant.spring\_learn.repository.CountryRepository;

@SpringBootApplication

public class SpringLearnApplication implements CommandLineRunner{

@Autowired

private CountryRepository countryRepository;

public static void main(String[] args){

SpringApplication.*run*(SpringLearnApplication.class, args);

}

@Override

public void run(String... args) throws Exception {

System.*out*.println("Countries containing 'ou': ");

countryRepository.findByCoNameContainingIgnoreCase("ou")

.forEach(c -> System.*out*.println(c.getCoCode() + " " + c.getCoName()));

System.*out*.println();

System.*out*.println("Countries containing 'ou' sorted: ");

countryRepository.findByCoNameContainingIgnoreCaseOrderByCoNameAsc("ou")

.forEach(c -> System.*out*.println(c.getCoCode() + " " + c.getCoName()));

System.*out*.println();

System.*out*.println("Countries starting with 'z': ");

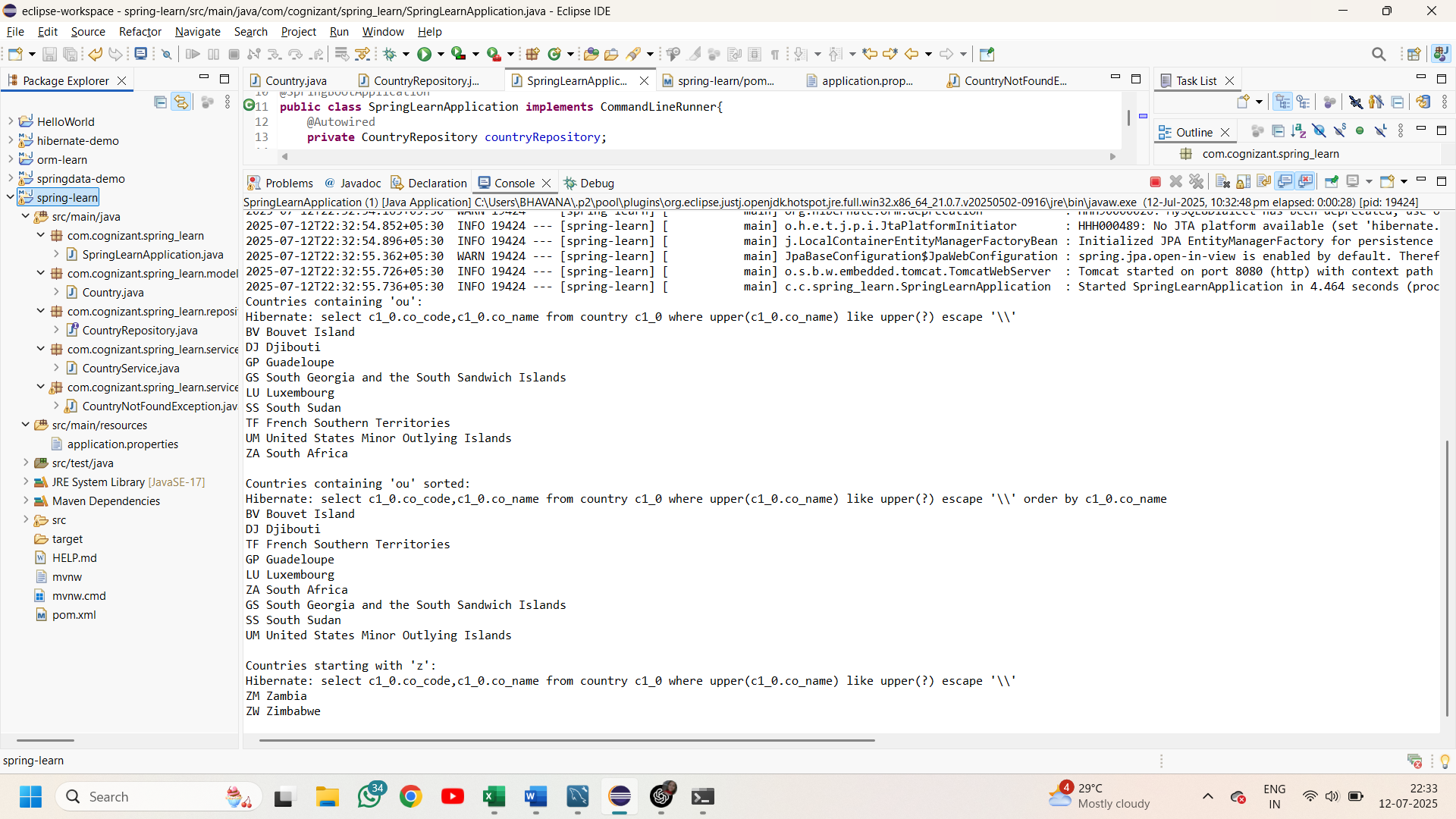
countryRepository.findByCoNameStartingWithIgnoreCase("z")

.forEach(c -> System.*out*.println(c.getCoCode() + " " + c.getCoName()));

}

}

**Output**:



**Demonstrate implementation of O/R Mapping**

**Create payroll tables and bean mapping**

**Implement many to one relationship between Employee and Department**

**->I have created the schema in mysql workbench**

**Department.java**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.\*;

@Entity

@Table(name="department")

public class Department {

@Id

@GeneratedValue(strategy=GenerationType.*IDENTITY*)

@Column(name="dp\_id")

private int id;

@Column(name="dp\_name")

private String name;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Id: "+ id+"Name: "+name;

}

}

**Employee.java**

package com.cognizant.orm\_learn.model;

import java.util.Date;

import jakarta.persistence.Column;

import jakarta.persistence.Entity;

import jakarta.persistence.GeneratedValue;

import jakarta.persistence.GenerationType;

import jakarta.persistence.Id;

import jakarta.persistence.JoinColumn;

import jakarta.persistence.ManyToOne;

import jakarta.persistence.Table;

@Entity

@Table(name="employee")

public class Employee {

@Id

@GeneratedValue(strategy=GenerationType.*IDENTITY*)

@Column(name="em\_id")

private int id;

@Column(name="em\_name")

private String name;

@Column(name="em\_salary")

private double salary;

@Column(name="em\_permanent")

private boolean permanent;

@Column(name="em\_date\_of\_birth")

private Date dateOfBirth;

@ManyToOne

@JoinColumn(name = "em\_dp\_id")

private Department department;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

public boolean isPermanent() {

return permanent;

}

public void setPermanent(boolean permanent) {

this.permanent = permanent;

}

public Date getDateOfBirth() {

return dateOfBirth;

}

public void setDateOfBirth(Date dateOfBirth) {

this.dateOfBirth = dateOfBirth;

}

public Department getDepartment() {

return department;

}

public void setDepartment(Department department) {

this.department = department;

}

@Override

public String toString() {

return "Employee{" +

"id=" + id +

", name='" + name + '\'' +

", salary=" + salary +

", permanent=" + permanent +

", dateOfBirth=" + dateOfBirth +

", department=" + (department != null ? department.getName() : "null") +

'}';

}

}

**Skill.java**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.\*;

@Entity

@Table(name="skill")

public class Skill {

@Id

@GeneratedValue(strategy=GenerationType.*IDENTITY*)

@Column(name="sk\_id")

private int id;

@Column(name="sk\_name")

private String name;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

@Override

public String toString() {

return "Id: "+ id+"Name: "+name;

}

}

**DepartmentService.java**

package com.cognizant.orm\_learn.service;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

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

import com.cognizant.orm\_learn.repository.DepartmentRepository;

import jakarta.transaction.Transactional;

@Service

public class DepartmentService {

private static final Logger *log*=LoggerFactory.*getLogger*(DepartmentService.class);

@Autowired

private DepartmentRepository departmentRepository;

@Transactional

public Department get(int id) {

*log*.info("Start");

return departmentRepository.findById(id).get();

}

@Transactional

public void save(Department department) {

*log*.info("Start");

departmentRepository.save(department);

*log*.info("End");

}

}

**EmployeeService.java**

package com.cognizant.orm\_learn.service;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

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

import com.cognizant.orm\_learn.repository.EmployeeRepository;

import jakarta.transaction.Transactional;

@Service

public class EmployeeService {

private static final Logger *log*=LoggerFactory.*getLogger*(EmployeeService.class);

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public Employee get(int id) {

*log*.info("Start");

return employeeRepository.findById(id).get();

}

@Transactional

public void save(Employee employee) {

*log*.info("Start");

employeeRepository.save(employee);

*log*.info("End");

}

}

**SkillService.java**

package com.cognizant.orm\_learn.service;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

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

import com.cognizant.orm\_learn.repository.SkillRepository;

import jakarta.transaction.Transactional;

@Service

public class SkillService {

private static final Logger *log*=LoggerFactory.*getLogger*(SkillService.class);

@Autowired

private SkillRepository skillRepository;

@Transactional

public Skill get(int id) {

*log*.info("Start");

return skillRepository.findById(id).get();

}

@Transactional

public void save(Skill skill) {

*log*.info("Start");

skillRepository.save(skill);

*log*.info("End");

}

}

**DepartmentRepository.java**

package com.cognizant.orm\_learn.repository;

import org.springframework.data.jpa.repository.JpaRepository;

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

public interface DepartmentRepository extends JpaRepository<Department,Integer>{

}

**EmployeeRepository.java**

package com.cognizant.orm\_learn.repository;

import org.springframework.data.jpa.repository.JpaRepository;

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

public interface EmployeeRepository extends JpaRepository<Employee, Integer>{

}

**SkillRepository.java**

package com.cognizant.orm\_learn.repository;

import org.springframework.data.jpa.repository.JpaRepository;

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

public interface SkillRepository extends JpaRepository<Skill,Integer> {

}

**OrmLearnApplication.java**

package com.cognizant.orm\_learn;

import java.util.Date;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

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

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

import com.cognizant.orm\_learn.service.DepartmentService;

import com.cognizant.orm\_learn.service.EmployeeService;

import com.cognizant.orm\_learn.service.SkillService;

@SpringBootApplication

public class OrmLearnApplication {

private static final Logger *LOGGER* = LoggerFactory.*getLogger*(OrmLearnApplication.class);

private static EmployeeService *employeeService*;

private static DepartmentService *departmentService*;

private static SkillService *skillService*;

public static void main(String[] args) {

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

*employeeService* = context.getBean(EmployeeService.class);

*departmentService* = context.getBean(DepartmentService.class);

*skillService* = context.getBean(SkillService.class);

// testGetEmployee();

// testAddEmployee();

*testUpdateEmployee*();

}

private static void testGetEmployee() {

*LOGGER*.info("Start");

Employee employee = *employeeService*.get(1);

*LOGGER*.debug("Employee: {}", employee);

*LOGGER*.debug("Department: {}", employee.getDepartment());

*LOGGER*.info("End");

}

private static void testAddEmployee() {

*LOGGER*.info("Start");

Employee employee = new Employee();

employee.setName("Jane Doe");

employee.setSalary(75000);

employee.setPermanent(true);

employee.setDateOfBirth(new Date());

Department department = *departmentService*.get(1);

employee.setDepartment(department);

*employeeService*.save(employee);

*LOGGER*.debug("Saved Employee: {}", employee);

*LOGGER*.info("End");

}

private static void testUpdateEmployee() {

*LOGGER*.info("Start");

Employee employee = *employeeService*.get(1);

Department newDepartment = *departmentService*.get(2);

employee.setDepartment(newDepartment);

*employeeService*.save(employee);

*LOGGER*.debug("Updated Employee: {}", employee);

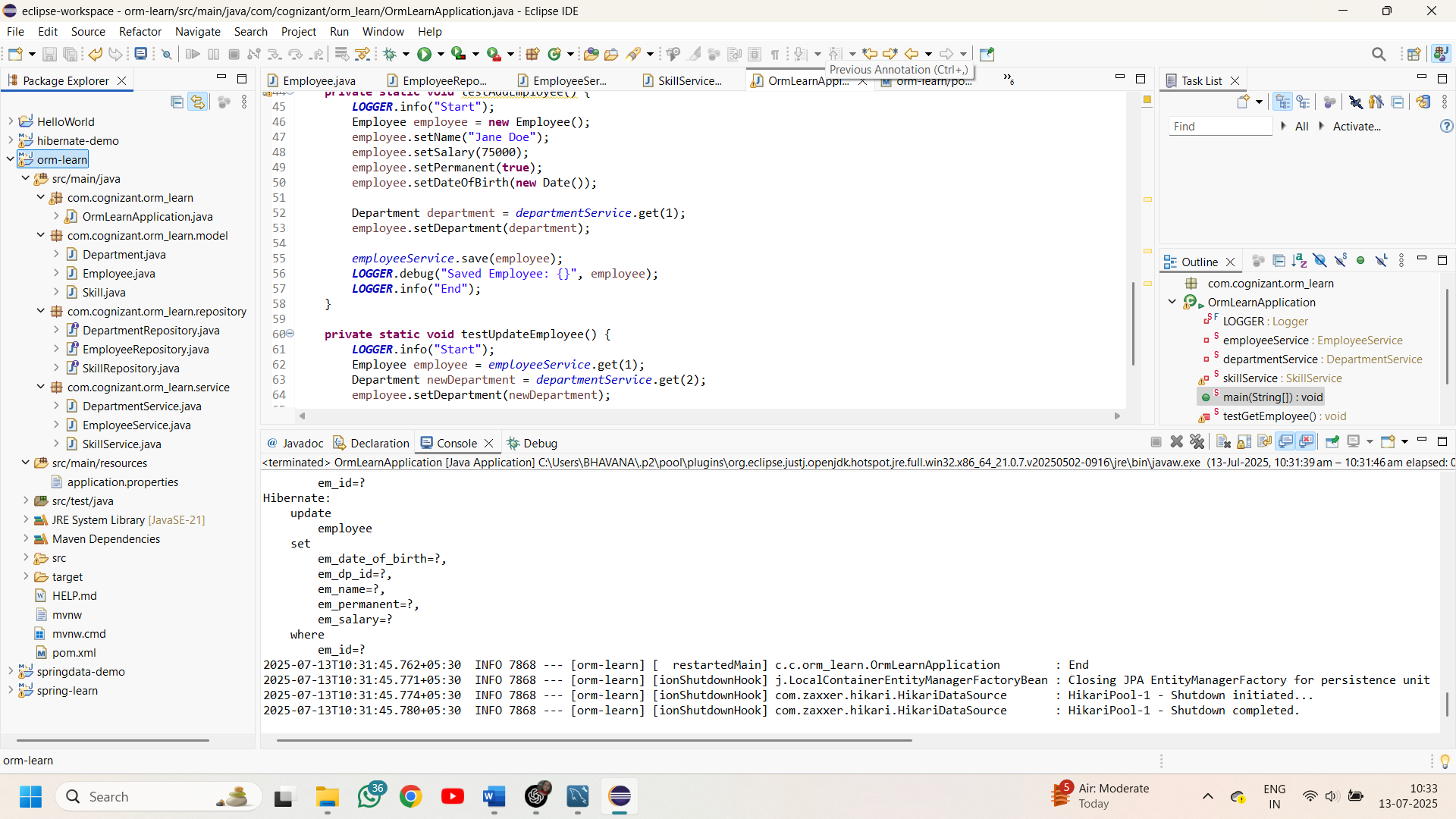
*LOGGER*.info("End");

}

}

**Output:**

Updation:



**Mysql workbench:**

Updation**:**