**Exercise 3: Employee Management System - Creating Repositories**

**Business Scenario:**

Create repositories for Employee and Department entities to perform CRUD operations.

**Instructions:**

Overview of Spring Data Repositories:

Learn the benefits of using Spring Data repositories.

**Creating Repositories:**

Create EmployeeRepository and DepartmentRepository interfaces extending JpaRepository.

Define derived query methods in these repositories.

package com.company.repository;

import com.company.entity.Employee;

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

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

import org.springframework.data.repository.query.Param;

import org.springframework.data.domain.Page;

import org.springframework.data.domain.Pageable;

import com.company.projection.EmployeeView;

import java.util.List;

public interface EmployeeRepository extends JpaRepository <Employee, Long> {

List<Employee> findByName(String name);

List<Employee> findByDepartment(String name);

@Query(name = "Employee.findByDept")

List<Employee> findEmployeesByDepartmentName(@Param("deptName") String deptName);

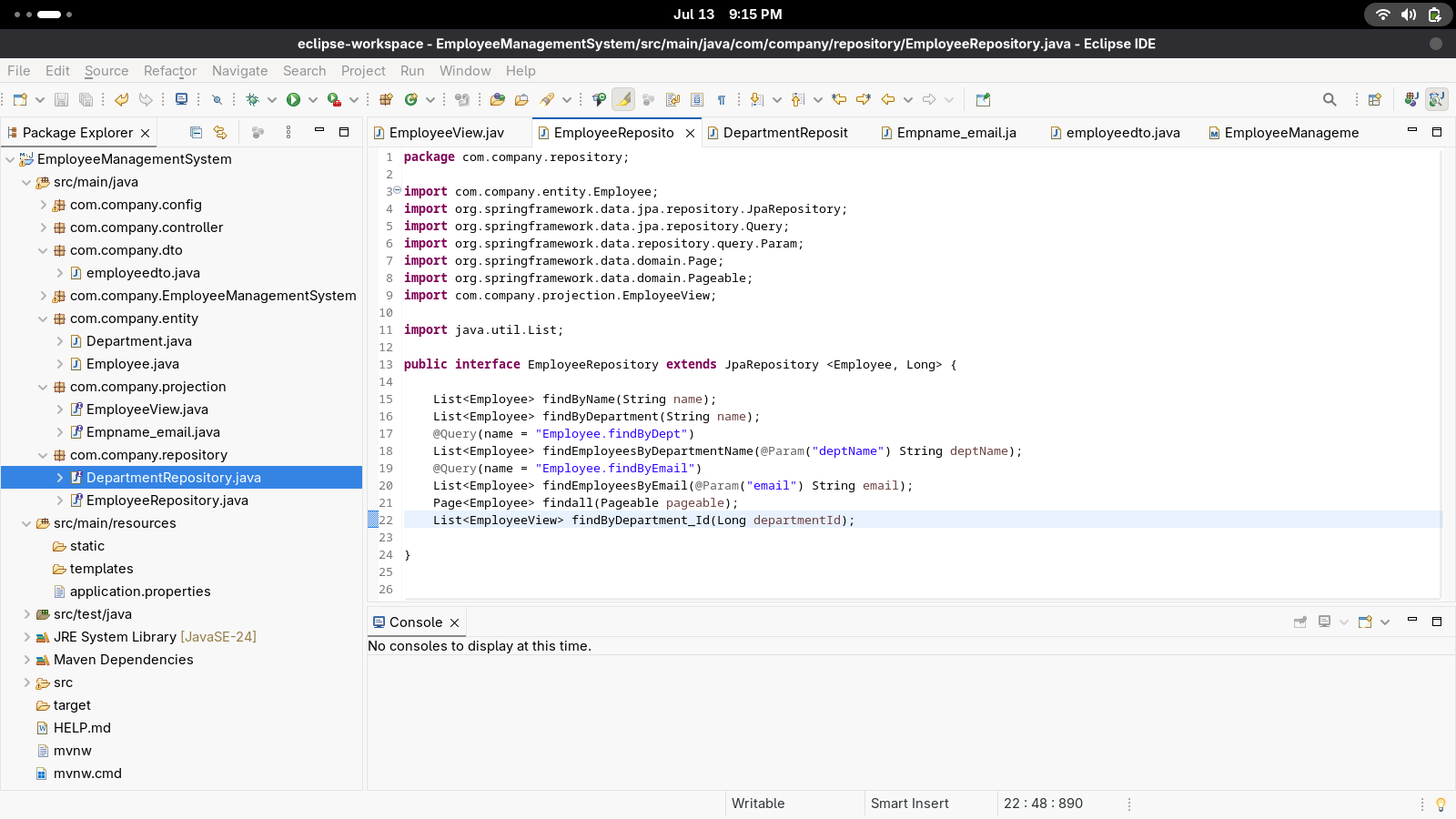
@Query(name = "Employee.findByEmail")

List<Employee> findEmployeesByEmail(@Param("email") String email);

Page<Employee> findall(Pageable pageable);

List<EmployeeView> findByDepartment\_Id(Long departmentId);

}



package com.company.repository;

import com.company.entity.Department;

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

public interface DepartmentRepository extends JpaRepository<Department, Long> {

Department findByName(String name);

}

