**Exercise 5: Employee Management System - Defining Query Methods**

**Business Scenario:**

Enhance your repository to support custom queries.

Instructions:

1. **Defining Query Methods:**

Use keywords in method names to create custom query methods.

Implement custom query methods using the @Query annotation.

1. **Named Queries:**

Define and execute named queries with @NamedQuery and @NamedQueries.

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")

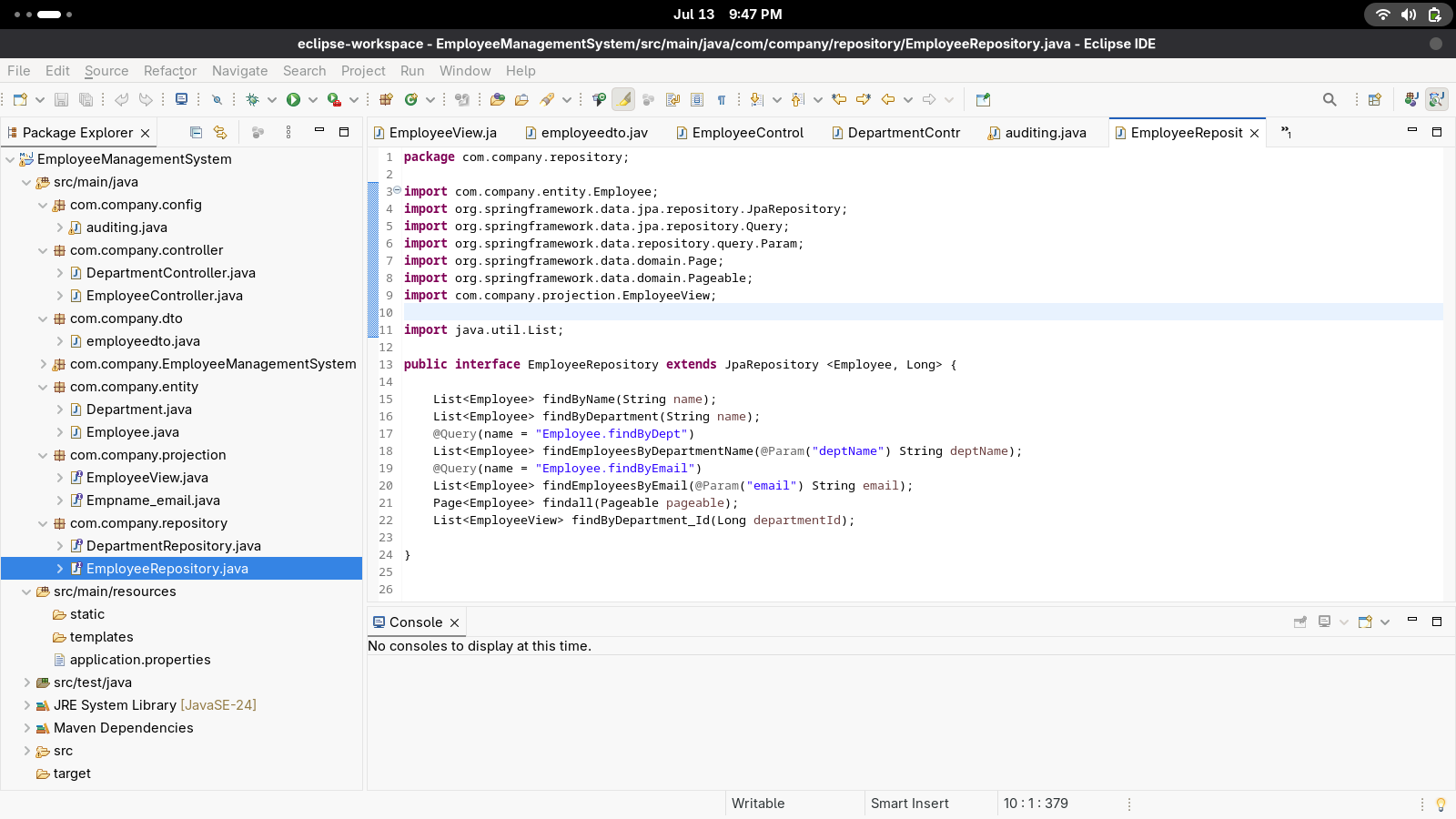
List<Employee> findEmployeesByEmail(@Param("email") String email);

Page<Employee> findall(Pageable pageable);

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

}

package com.company.entity;



import jakarta.persistence.\*;

import lombok.\*;

import org.springframework.data.annotation.CreatedBy;

import org.springframework.data.annotation.LastModifiedBy;

import org.springframework.data.annotation.CreatedDate;

import org.springframework.data.annotation.LastModifiedDate;

import org.springframework.data.jpa.domain.support.AuditingEntityListener;

import java.time.LocalDateTime;

@NamedQueries({

@NamedQuery(

name = "Employee.findByEmail",

query = "SELECT e FROM Employee e WHERE e.email = :email"

),

@NamedQuery(

name = "Employee.findByDeptName",

query = "SELECT e FROM Employee e WHERE e.department.name = :deptName"

)

})

@Entity

@EntityListeners(AuditingEntityListener.class)

@Table(name = "employees")

@Data

@NoArgsConstructor

@AllArgsConstructor

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String email;

@ManyToOne

@JoinColumn(name="department\_id")

private Department department;

public String getName() {

return name;

}

@CreatedDate

private LocalDateTime createdDate;

@LastModifiedDate

private LocalDateTime modifiedDate;

@CreatedBy

private String createdBy;

@LastModifiedBy

private String modifiedBy;

}

