EmployeeRepository.java:

package com.example.employeemanagementsystem.repository;

import com.example.employeemanagementsystem.model.Employee;

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

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

import org.springframework.stereotype.Repository;

import java.util.List;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Derived query methods

List<Employee> findByName(String name);

List<Employee> findByEmailContaining(String keyword);

List<Employee> findByDepartmentName(String departmentName);

// Custom @Query using JPQL

@Query("SELECT e FROM Employee e WHERE e.department.name = :deptName")

List<Employee> getEmployeesByDepartmentName(String deptName);

@Query("SELECT e FROM Employee e WHERE e.name LIKE %:pattern%")

List<Employee> searchEmployeesByNamePattern(String pattern);

}

Employee.java:

package com.example.employeemanagementsystem.model;

import jakarta.persistence.\*;

import lombok.\*;

@Entity

@Table(name = "employees")

@Data

@NoArgsConstructor

@AllArgsConstructor

@Builder

@NamedQueries({

@NamedQuery(

name = "Employee.findByDepartmentId",

query = "SELECT e FROM Employee e WHERE e.department.id = :deptId"

),

@NamedQuery(

name = "Employee.searchByNamePrefix",

query = "SELECT e FROM Employee e WHERE e.name LIKE CONCAT(:prefix, '%')"

)

})

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;

}

EmployeeRepository.java:

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

List<Employee> findByDepartmentId(@Param("deptId") Long deptId);

@Query(name = "Employee.searchByNamePrefix")

List<Employee> searchByNamePrefix(@Param("prefix") String prefix);