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

**EmployeeRepository**

package com.employee.management.system.repositories;

import com.employee.management.system.entities.Employee;

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

import org.springframework.stereotype.Repository;

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

List<Employee> findByName(String name);

List<Employee> findByDepartment(Department department);

List<Employee> findByEmail(String email);

}

**DepartmentRepository**

package com.employee.management.system.repositories;

import com.employee.management.system.entities.Department;

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

import org.springframework.stereotype.Repository;

@Repository

public interface DepartmentRepository extends JpaRepository<Department, Long> {

Department findByName(String name);

}

In the above code, we have created two repository interfaces: EmployeeRepository and DepartmentRepository. These interfaces extend JpaRepository and provide methods for performing CRUD operations on the Employee and Department entities, respectively.

We have also defined derived query methods in these repositories using Spring Data JPA's query method naming convention. These methods allow us to perform specific queries on the data without writing explicit query code.

For example, the findByName method in EmployeeRepository will automatically generate a query to find employees by name, while the findByDepartment method will find employees by department.

Similarly, the findByName method in DepartmentRepository will find departments by name.