**Exercise 8: Employee Management System - Creating Projections**

**EmployeeSummary (interface-based projection)**

package com.employee.management.system.projections;

public interface EmployeeSummary {

String getName();

String getEmail();

String getDepartmentName();

}

**EmployeeDetails(class-based projection)**

package com.employee.management.system.projections;

public class EmployeeDetails {

private String name;

private String email;

private Department department;

public EmployeeDetails(String name, String email, Department department) {

this.name = name;

this.email = email;

this.department = department;

}

// getters and setters

}

**EmployeeRepository**

package com.employee.management.system.repositories;

import com.employee.management.system.projections.EmployeeSummary;

import com.employee.management.system.projections.EmployeeDetails;

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> {

@Query("SELECT e.name AS name, e.email AS email, d.name AS departmentName FROM Employee e JOIN e.department d")

List<EmployeeSummary> getEmployeeSummaries();

@Query("SELECT new com.employee.management.system.projections.EmployeeDetails(e.name, e.email, e.department) FROM Employee e")

List<EmployeeDetails> getEmployeeDetails();

}

In the above code, I have defined two projections: EmployeeSummary (interface-based) and EmployeeDetails (class-based).

I have used @Query annotation to define custom queries that fetch specific data subsets from the Employee entity.

In the EmployeeSummary projection, I have used @Value to specify the fetched columns.

In the EmployeeDetails projection, I have used a constructor expression to specify the fetched columns and create a new instance of the projection class.

Note that I have also updated the EmployeeRepository to include methods that return these projections.