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

**Business Scenario:**

Create projections to fetch specific data subsets from the employee and department entities.

**Instructions:**

1. **Projections:**
   * Define interface-based and class-based projections.
   * Use **@Value** and constructor expressions to control the fetched data.

**Solution**

Define Entities

import javax.persistence.Entity;

import javax.persistence.Id;

import javax.persistence.ManyToOne;

@Entity

public class Employee {

@Id

private Long id;

private String name;

private String position;

private Double salary;

@ManyToOne

private Department department;

// Getters and Setters

}

import javax.persistence.Entity;

import javax.persistence.Id;

@Entity

public class Department {

@Id

private Long id;

private String name;

// Getters and Setters

}

Interface-Based Projection

import org.springframework.beans.factory.annotation.Value;

public interface EmployeeProjection {

@Value("#{target.name} - #{target.position}")

String getEmployeeDetails();

String getName();

Double getSalary();

}

import org.springframework.beans.factory.annotation.Value;

public interface DepartmentProjection {

@Value("#{target.name}")

String getDepartmentName();

}

Class-Based Projections

public class EmployeeProjection {

private String name;

private String position;

private Double salary;

public EmployeeProjection(String name, String position, Double salary) {

this.name = name;

this.position = position;

this.salary = salary;

}

// Getters

public String getName() {

return name;

}

public String getPosition() {

return position;

}

public Double getSalary() {

return salary;

}

}

public class DepartmentProjection {

private String name;

public DepartmentProjection(String name) {

this.name = name;

}

// Getter

public String getName() {

return name;

}

}

DepartmentRepository

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

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

public interface DepartmentRepository extends JpaRepository<Department, Long> {

@Query("SELECT new com.example.DepartmentProjection(d.name) FROM Department d")

List<DepartmentProjection> findAllDepartmentProjections();

}