Employee.java:

@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private double salary;

@ManyToOne

private Department department;

// Getters and Setters

}

**Department.java:**

@Entity

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

// Getters and Setters

}

**EmployeeNameSalary.java:**

public interface EmployeeNameSalary {

String getName();

double getSalary();

}

**EmployeeDTO.java:**

public class EmployeeDTO {

private String name;

private String departmentName;

public EmployeeDTO(String name, String departmentName) {

this.name = name;

this.departmentName = departmentName;

}

// Getters and Setters

}

**EmployeeRepository.java:**

public interface EmployeeRepository extends JpaRepository<Employee, Long> {

// Interface-based projection

List<EmployeeNameSalary> findBySalaryGreaterThan(double salary);

// Class-based projection using @Query and constructor expression

@Query("SELECT new com.example.employeemanagement.projection.EmployeeDTO(e.name, d.name) " +

"FROM Employee e JOIN e.department d WHERE d.name = :deptName")

List<EmployeeDTO> findEmployeesByDepartmentName(@Param("deptName") String deptName);

}

**DepartmentRepository.java:**

public interface DepartmentRepository extends JpaRepository<Department, Long> {}

**EmployeeController.java**

@RestController

@RequestMapping("/employees")

public class EmployeeController {

@Autowired

private EmployeeRepository employeeRepository;

@GetMapping("/salary-above/{amount}")

public List<EmployeeNameSalary> getEmployeesWithHighSalary(@PathVariable double amount) {

return employeeRepository.findBySalaryGreaterThan(amount);

}

@GetMapping("/by-department/{name}")

public List<EmployeeDTO> getEmployeesByDepartment(@PathVariable String name) {

return employeeRepository.findEmployeesByDepartmentName(name);

}

}