**Hands on 5: Get all employees using Native Query**

**Employee.java**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.\*;

import java.util.Date;

import java.util.List;

@Entity

@Table(name = "employee")

public class Employee {

@Id

@Column(name = "em\_id")

private int id;

@Column(name = "em\_name")

private String name;

@Column(name = "em\_salary")

private double salary;

@Column(name = "em\_permanent")

private boolean permanent;

@Column(name = "em\_date\_of\_birth")

private Date dateOfBirth;

@ManyToOne

@JoinColumn(name = "em\_dp\_id") // employee.department\_id references department.id

private Department department;

@ManyToMany

@JoinTable(name = "employee\_skill",

joinColumns = @JoinColumn(name = "es\_em\_id"),

inverseJoinColumns = @JoinColumn(name = "es\_sk\_id"))

private List<Skill> skillList;

// Getters and setters

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

public boolean isPermanent() {

return permanent;

}

public void setPermanent(boolean permanent) {

this.permanent = permanent;

}

public Date getDateOfBirth() {

return dateOfBirth;

}

public void setDateOfBirth(Date dateOfBirth) {

this.dateOfBirth = dateOfBirth;

}

public Department getDepartment() {

return department;

}

public void setDepartment(Department department) {

this.department = department;

}

public List<Skill> getSkillList() {

return skillList;

}

public void setSkillList(List<Skill> skillList) {

this.skillList = skillList;

}

@Override

public String toString() {

return "Employee [id=" + id +

", name=" + name +

", salary=" + salary +

", permanent=" + permanent +

", dateOfBirth=" + dateOfBirth +

", department=" + (department != null ? department.getName() : "null") +

"]";

}

}

**EmployeeRepository**

package com.cognizant.orm\_learn.repository;

import com.cognizant.orm\_learn.model.Employee;

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

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

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

import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

@Query("SELECT e FROM Employee e LEFT JOIN FETCH e.department d LEFT JOIN FETCH e.skillList WHERE e.permanent = true")

List<Employee> getAllPermanentEmployees();

@Query("SELECT AVG(e.salary) FROM Employee e WHERE e.department.id = :id")

Double getAverageSalary(@Param("id") int id);

@Query(value = "SELECT \* FROM employee", nativeQuery = true)

List<Employee> getAllEmployeesNative();

}

**EmployeeService**

package com.cognizant.orm\_learn.service;

import com.cognizant.orm\_learn.model.Employee;

import com.cognizant.orm\_learn.repository.EmployeeRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

public List<Employee> getAllPermanentEmployees() {

return employeeRepository.getAllPermanentEmployees();

}

public Double getAverageSalary(int deptId) {

return employeeRepository.getAverageSalary(deptId);

}

public List<Employee> getAllEmployeesNative() {

return employeeRepository.getAllEmployeesNative();

}

}

**OrmLearnApplication**

LOGGER.info("Start fetching all employees using Native Query");

List<Employee> allEmployeesNative = employeeService.getAllEmployeesNative();

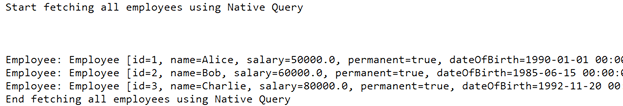
allEmployeesNative.forEach(e -> {

LOGGER.info("Employee: {}", e);

});

LOGGER.info("End fetching all employees using Native Query");

**Output**

****