**Hands On 2: Get all permanent employees using HQL**

**Objective:**

Write an HQL query (using @Query) to fetch:

* All employees where permanent = 1
* Also fetch their associated:
  + Department
  + Skill list

**1. Entity Setup**

**Employee.java**

@Entity

@Table(name = "employee")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private double salary;

private boolean permanent;

@Temporal(TemporalType.DATE)

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

private Date dateOfBirth;

@ManyToOne

@JoinColumn(name = "em\_dp\_id")

private Department department;

@ManyToMany(fetch = FetchType.LAZY)

@JoinTable(name = "employee\_skill",

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

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

private Set<Skill> skillList;

// Getters and setters

}

**Department.java**

@Entity

@Table(name = "department")

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

// No eager fetch

@OneToMany(mappedBy = "department")

private Set<Employee> employeeList;

}

**Skill.java**

@Entity

@Table(name = "skill")

public class Skill {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

@ManyToMany(mappedBy = "skillList")

private Set<Employee> employeeList;

}

**2. Repository: EmployeeRepository.java**

@Repository

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();

}

**3. Service: EmployeeService.java**

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

public List<Employee> getAllPermanentEmployees() {

return employeeRepository.getAllPermanentEmployees();

}

}

**4. OrmLearnApplication.java — Test Method**

@SpringBootApplication

public class OrmLearnApplication {

private static final Logger LOGGER = LoggerFactory.getLogger(OrmLearnApplication.class);

private static EmployeeService employeeService;

public static void main(String[] args) {

ApplicationContext context = SpringApplication.run(OrmLearnApplication.class, args);

employeeService = context.getBean(EmployeeService.class);

testGetAllPermanentEmployees(); // Call test here

}

public static void testGetAllPermanentEmployees() {

LOGGER.info("Start");

List<Employee> employees = employeeService.getAllPermanentEmployees();

LOGGER.debug("Permanent Employees: {}", employees);

employees.forEach(e -> {

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

LOGGER.debug("Department: {}", e.getDepartment().getName());

LOGGER.debug("Skills: {}", e.getSkillList());

});

LOGGER.info("End");

}

}