**Hands-On 3: Create Payroll Tables & Bean Mapping**

**Step 1: SQL Table Creation**

CREATE TABLE department (

dp\_id INT PRIMARY KEY AUTO\_INCREMENT,

dp\_name VARCHAR(45)

);

CREATE TABLE skill (

sk\_id INT PRIMARY KEY AUTO\_INCREMENT,

sk\_name VARCHAR(45)

);

CREATE TABLE employee (

em\_id INT PRIMARY KEY AUTO\_INCREMENT,

em\_name VARCHAR(45),

em\_salary DECIMAL(10,2),

em\_permanent BOOLEAN,

em\_date\_of\_birth DATE,

em\_dp\_id INT,

FOREIGN KEY (em\_dp\_id) REFERENCES department(dp\_id)

);

CREATE TABLE employee\_skill (

es\_id INT PRIMARY KEY AUTO\_INCREMENT,

es\_em\_id INT,

es\_sk\_id INT,

FOREIGN KEY (es\_em\_id) REFERENCES employee(em\_id),

FOREIGN KEY (es\_sk\_id) REFERENCES skill(sk\_id)

);

**Step 2: Model Classes(Bean Mapping)**

**Department.java**

package com.cognizant.orm\_learn.model;

import javax.persistence.\*;

@Entity

@Table(name = "department")

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column(name = "dp\_name")

private String name;

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;

}

@Override

public String toString() {

return "Department [id=" + id + ", name=" + name + "]";

}

}

**Skill.java**

package com.cognizant.orm\_learn.model;

import javax.persistence.\*;

@Entity

@Table(name = "skill")

public class Skill {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column(name = "sk\_name")

private String name;

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;

}

@Override

public String toString() {

return "Skill [id=" + id + ", name=" + name + "]";

}

}

**Employee.java**

package com.cognizant.orm\_learn.model;

import javax.persistence.\*;

import java.util.Date;

@Entity

@Table(name = "employee")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

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;

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;

}

@Override

public String toString() {

return "Employee [id=" + id + ", name=" + name + ", salary=" + salary + ", permanent=" + permanent

+ ", dateOfBirth=" + dateOfBirth + "]";

}

}

**Step 3: Repository Interfaces**

**DepartmentRepository.java**

@Repository

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

**SkillRepository.java**

@Repository

public interface SkillRepository extends JpaRepository<Skill, Integer> {}

**EmployeeRepository.java**

@Repository

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {}

**Step 4: Sample Data Insert SQL**

INSERT INTO department (dp\_name) VALUES ('HR'), ('IT'), ('Finance');

INSERT INTO skill (sk\_name) VALUES ('Java'), ('SQL'), ('Spring Boot');

INSERT INTO employee (em\_name, em\_salary, em\_permanent, em\_date\_of\_birth, em\_dp\_id)

VALUES ('Ravi', 45000, true, '1998-05-10', 1),

('Priya', 50000, true, '1997-09-15', 2);

**✔ Tables Created: employee, department, skill, employee\_skill  
✔ Primary Keys with AUTO\_INCREMENT used  
✔ Foreign Key relationship defined in employee table (department ID)  
✔ Bean Mapping with @Entity, @Table, @Id, @Column, @GeneratedValue annotations  
✔ No relationships mapped yet in Java, only table mapping completed**

**Hands-on 4 - Many to One Mapping (Employee → Department)**

**Add Department mapping:**

@ManyToOne

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

private Department department;

public Department getDepartment() {

return department;

}

public void setDepartment(Department department) {

this.department = department;

}

**Employee.java**

package com.cognizant.orm\_learn.model;

import javax.persistence.\*;

import java.util.Date;

@Entity

@Table(name = "employee")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

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")

private Department department;

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;

}

@Override

public String toString() {

return "Employee [id=" + id + ", name=" + name + ", salary=" + salary + ", permanent=" + permanent

+ ", dateOfBirth=" + dateOfBirth + "]";

}

}

**Repositories**

**EmployeeRepository.java**

package com.cognizant.orm\_learn.repository;

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

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

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**DepartmentRepository.java**

package com.cognizant.orm\_learn.repository;

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

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

public interface DepartmentRepository extends JpaRepository<Department, Integer> {

}

**Service Classes**

**EmployeeService.java**

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 javax.transaction.Transactional;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public Employee get(int id) {

return employeeRepository.findById(id).get();

}

@Transactional

public void save(Employee employee) {

employeeRepository.save(employee);

}

}

**DepartmentService.java**

package com.cognizant.orm\_learn.service;

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

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

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

import org.springframework.stereotype.Service;

import javax.transaction.Transactional;

@Service

public class DepartmentService {

@Autowired

private DepartmentRepository departmentRepository;

@Transactional

public Department get(int id) {

return departmentRepository.findById(id).get();

}

@Transactional

public void save(Department department) {

departmentRepository.save(department);

}

}

**Application Testing**

**OrmLearnApplication.java**

@Autowired

private EmployeeService employeeService;

@Autowired

private DepartmentService departmentService;

@Override

public void run(String... args) throws Exception {

testGetEmployee();

testAddEmployee();

testUpdateEmployee();

}

private void testGetEmployee() {

Employee employee = employeeService.get(1);

System.out.println("Employee: " + employee);

System.out.println("Department: " + employee.getDepartment());

}

private void testAddEmployee() {

Employee employee = new Employee();

employee.setName("John");

employee.setSalary(50000);

employee.setPermanent(true);

employee.setDateOfBirth(new Date());

Department department = departmentService.get(1);

employee.setDepartment(department);

employeeService.save(employee);

System.out.println("Added Employee: " + employee);

}

private void testUpdateEmployee() {

Employee employee = employeeService.get(1);

Department department = departmentService.get(2);

employee.setDepartment(department);

employeeService.save(employee);

System.out.println("Updated Employee: " + employee);

}

**SQL Example**

INSERT INTO department (dp\_name) VALUES ('IT'), ('HR');

INSERT INTO employee (em\_name, em\_salary, em\_permanent, em\_date\_of\_birth, em\_dp\_id)

VALUES ('Alice', 45000, true, '1990-05-15', 1);

**Hands-on 5: One to Many Relationship (Department → Employees)**

In this, one Department has many Employees.

**Department.java**

@OneToMany(mappedBy = "department", fetch = FetchType.EAGER)

private Set<Employee> employeeList;

public Set<Employee> getEmployeeList() {

return employeeList;

}

public void setEmployeeList(Set<Employee> employeeList) {

this.employeeList = employeeList;

}

**Department.java**

package com.cognizant.orm\_learn.model;

import javax.persistence.\*;

import java.util.Set;

@Entity

@Table(name = "department")

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column(name = "dp\_name")

private String name;

@OneToMany(mappedBy = "department", fetch = FetchType.EAGER)

private Set<Employee> employeeList;

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 Set<Employee> getEmployeeList() {

return employeeList;

}

public void setEmployeeList(Set<Employee> employeeList) {

this.employeeList = employeeList;

}

@Override

public String toString() {

return "Department [id=" + id + ", name=" + name + "]";

}

}

**OrmLearnApplication.java for Testing**

package com.cognizant.orm\_learn;

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

import com.cognizant.orm\_learn.service.DepartmentService;

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

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.CommandLineRunner;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private DepartmentService departmentService;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String... args) throws Exception {

testGetDepartment();

}

private void testGetDepartment() {

Department department = departmentService.get(1); // Get department with ID 1

System.out.println("Department: " + department);

System.out.println("Employees: " + department.getEmployeeList());

}

}

**Database Example**

INSERT INTO department (dp\_name) VALUES ('IT'), ('HR');

INSERT INTO employee (em\_name, em\_salary, em\_permanent, em\_date\_of\_birth, em\_dp\_id)

VALUES ('Alice', 45000, true, '1990-05-15', 1),

('Bob', 50000, false, '1985-03-20', 1),

('Eve', 55000, true, '1992-11-11', 2);

**Output**

**Department: Department [id=1, name=IT]**

**Employees: [Employee [id=1, name=Alice, salary=45000.0, permanent=true, dateOfBirth=Mon May 15 00:00:00 IST 1990], Employee [id=2, name=Bob, salary=50000.0, permanent=false, dateOfBirth=Wed Mar 20 00:00:00 IST 1985]]**

**Hands on 6**

**Implement many to many relationship between Employee and Skill**

**Employee.java**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.\*;

import java.util.Date;

import java.util.Set;

@Entity

@Table(name = "employee")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@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")

private Department department;

@ManyToMany(fetch = FetchType.EAGER)

@JoinTable(name = "employee\_skill",

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

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

private Set<Skill> skillList;

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 Set<Skill> getSkillList() { return skillList; }

public void setSkillList(Set<Skill> skillList) { this.skillList = skillList; }

@Override

public String toString() {

return "Employee [id=" + id + ", name=" + name + ", salary=" + salary + ", permanent=" + permanent + "]";

}

}

**Skill.java**

package com.cognizant.orm\_learn.model;

import jakarta.persistence.\*;

import java.util.Set;

@Entity

@Table(name = "skill")

public class Skill {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "sk\_id")

private int id;

@Column(name = "sk\_name")

private String name;

@ManyToMany(mappedBy = "skillList")

private Set<Employee> employeeList;

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 Set<Employee> getEmployeeList() { return employeeList; }

public void setEmployeeList(Set<Employee> employeeList) { this.employeeList = employeeList; }

@Override

public String toString() {

return "Skill [id=" + id + ", name=" + name + "]";

}

}

**OrmLearnApplication.java**

package com.cognizant.orm\_learn;

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

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

import com.cognizant.orm\_learn.service.EmployeeService;

import com.cognizant.orm\_learn.service.SkillService;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import java.util.Set;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private EmployeeService employeeService;

@Autowired

private SkillService skillService;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String... args) throws Exception {

testAddSkillToEmployee();

}

private void testAddSkillToEmployee() {

Employee employee = employeeService.get(1);

Skill skill = skillService.get(2);

Set<Skill> skillList = employee.getSkillList();

skillList.add(skill);

employee.setSkillList(skillList);

employeeService.save(employee);

System.out.println("Skill added to Employee Successfully!");

}

}

1. **@ManyToMany mapping established between Employee and Skill.**
2. **Employee holds Set of Skill (skillList), Skill holds Set of Employee (employeeList).**
3. **We use testAddSkillToEmployee() to:**
   * **Fetch employee by ID**
   * **Fetch skill by ID**
   * **Add skill to employee's skillList**
   * **Save employee**
4. **Database table employee\_skill is automatically updated to reflect the relationship.**