**Hands on Exercise 6 - Implement many to many relationship between Employee and Skill**

**CODE:**

**Employee Java: -**

package com.cognizant.ex2handson3.model;  
import jakarta.persistence.\*;  
import java.util.Date;  
import java.util.Set;  
  
@Entity  
@Table(name = "employee")  
public class Employee {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private int id;  
  
 @Column(name = "name")  
 private String name;  
  
 @Column(name = "salary")  
 private double salary;  
  
 @Column(name = "permanent")  
 private boolean permanent;  
  
 @Column(name = "date\_of\_birth")  
 @Temporal(TemporalType.*DATE*)  
 private Date dateOfBirth;  
  
 // ManyToOne: many employees belong to one department  
 @ManyToOne  
 @JoinColumn(name = "department\_id")  
 private Department department;  
  
 // ManyToMany: employee can have multiple skills  
 @ManyToMany(fetch = FetchType.*EAGER*)  
 @JoinTable(  
 name = "employee\_skill",  
 joinColumns = @JoinColumn(name = "employee\_id"),  
 inverseJoinColumns = @JoinColumn(name = "skill\_id")  
 )  
 private Set<Skill> skillList;  
  
 public Employee() {  
 }  
  
 public Employee(int id, String name, double salary, boolean permanent, Date dateOfBirth, Department department, Set<Skill> skillList) {  
 this.id = id;  
 this.name = name;  
 this.salary = salary;  
 this.permanent = permanent;  
 this.dateOfBirth = dateOfBirth;  
 this.department = department;  
 this.skillList = 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;  
 }  
}

**Skill Class: -**

package com.cognizant.ex2handson3.model;  
import jakarta.persistence.\*;  
import java.util.Set;  
  
@Entity  
@Table(name = "skill")  
public class Skill {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private int id;  
  
 @Column(name = "name")  
 private String name;  
  
 @ManyToMany(mappedBy = "skillList")  
 private Set<Employee> employeeList;  
  
 public Skill() {  
 }  
  
 public Skill(int id, String name, Set<Employee> employeeList) {  
 this.id = id;  
 this.name = name;  
 this.employeeList = 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;  
 }  
}

**SkillService Class: -**

package com.cognizant.ex2handson3.service;  
  
import com.cognizant.ex2handson3.model.Skill;  
import com.cognizant.ex2handson3.repository.SkillRepository;  
import jakarta.transaction.Transactional;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class SkillService {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(SkillService.class);  
  
 @Autowired  
 private SkillRepository skillRepository;  
  
 @Transactional  
 public Skill get(int id) {  
 *LOGGER*.info("Start");  
 return skillRepository.findById(id).orElse(null);  
 }  
  
 @Transactional  
 public void save(Skill skill) {  
 *LOGGER*.info("Start");  
 skillRepository.save(skill);  
 *LOGGER*.info("End");  
 }  
}

**SkillRepository Interface: -**

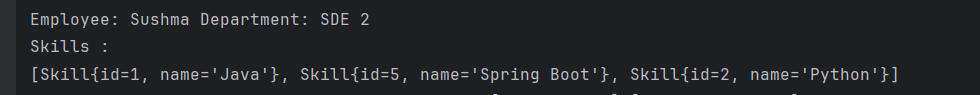
package com.cognizant.ex2handson3.repository;  
import com.cognizant.ex2handson3.model.Skill;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface SkillRepository extends JpaRepository<Skill,Integer> {  
}

**Main Class: -**

package com.cognizant.ex2handson3;  
import com.cognizant.ex2handson3.model.Department;  
import com.cognizant.ex2handson3.model.Employee;  
import com.cognizant.ex2handson3.model.Skill;  
import com.cognizant.ex2handson3.service.DepartmentService;  
import com.cognizant.ex2handson3.service.EmployeeService;  
import com.cognizant.ex2handson3.service.SkillService;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.context.ApplicationContext;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
import java.sql.Date;  
import java.util.Set;  
  
@SpringBootApplication  
public class Ex2handson3Application {  
  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(Ex2handson3Application.class);  
  
 public static void main(String[] args) {  
 ApplicationContext context = SpringApplication.*run*(Ex2handson3Application.class, args);  
  
 EmployeeService employeeService = context.getBean(EmployeeService.class);  
 DepartmentService departmentService = context.getBean(DepartmentService.class);  
 SkillService skillService = context.getBean(SkillService.class);  
  
 *testAddSkillToEmployee*(employeeService, skillService);   
 }  
  
 private static void testAddEmployee(EmployeeService employeeService, DepartmentService departmentService) {  
 *LOGGER*.info("Start");  
  
 Employee employee = new Employee();  
 employee.setName("Ravi Kumar");  
 employee.setSalary(75000);  
 employee.setPermanent(true);  
 employee.setDateOfBirth(Date.*valueOf*("1990-08-15"));  
  
 Department department = departmentService.get(1);  
 employee.setDepartment(department);  
  
 employeeService.save(employee);  
  
 *LOGGER*.debug("Added Employee: {}", employee);  
 System.*out*.println("Added Employee: " + employee.getName() +  
 ", Dept: " + employee.getDepartment().getName());  
  
 *LOGGER*.info("End");  
 }  
  
  
 private static void testUpdateEmployee(EmployeeService employeeService, DepartmentService departmentService) {  
 *LOGGER*.info("Start");  
  
 Employee employee = employeeService.get(1);  
 Department newDept = departmentService.get(2);  
  
 employee.setDepartment(newDept);  
 employeeService.save(employee);  
  
 *LOGGER*.debug("Updated Employee: {}", employee);  
 System.*out*.println("Updated Employee: " + employee.getName() +  
 ", New Dept: " + employee.getDepartment().getName());  
 *LOGGER*.info("End");  
 }  
 private static void testGetEmployee(EmployeeService employeeService) {  
 *LOGGER*.info("Start");  
 Employee employee = employeeService.get(1);  
 *LOGGER*.debug("Employee: {}", employee);  
 *LOGGER*.debug("Department: {}", employee.getDepartment());  
 *LOGGER*.debug("Skills: {}", employee.getSkillList());  
 *LOGGER*.info("End");  
 }  
  
 private static void testAddSkillToEmployee(EmployeeService employeeService, SkillService skillService) {  
 *LOGGER*.info("Start");  
  
 Employee employee = employeeService.get(1);  
 Skill skill = skillService.get(2);  
  
 Set<Skill> skills = employee.getSkillList();  
 skills.add(skill);  
  
 employee.setSkillList(skills);  
 employeeService.save(employee);  
  
 *LOGGER*.debug("Updated Employee with new Skill: {}", employee);  
 *LOGGER*.info("End");  
 }  
  
  
}

**OUTPUT:**

Before adding skill:



After adding skill:

