**O/R Mapping Implementation using Spring Data JPA**

**Entity Definitions:**

**Department.java**

java

CopyEdit

@Entity

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

@OneToMany(mappedBy = "department")

private List<Employee> employeeList;

}

**Skill.java**

java

CopyEdit

@Entity

public class Skill {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

@ManyToMany(mappedBy = "skillList")

private List<Employee> employeeList;

}

**Employee.java**

java

CopyEdit

@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private double salary;

private boolean permanent;

@Temporal(TemporalType.DATE)

private Date dateOfBirth;

@ManyToOne

@JoinColumn(name = "department\_id")

private Department department;

@ManyToMany

@JoinTable(

name = "employee\_skill",

joinColumns = @JoinColumn(name = "employee\_id"),

inverseJoinColumns = @JoinColumn(name = "skill\_id")

)

private List<Skill> skillList;

}

**Repository:**

**EmployeeRepository.java**

java

CopyEdit

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();

}

**Service Layer:**

**EmployeeService.java**

java

CopyEdit

public interface EmployeeService {

List<Employee> getAllPermanentEmployees();

double getAverageSalary(int deptId);

List<Employee> getAllEmployeesNative();

}

**EmployeeServiceImpl.java**

java

CopyEdit

@Service

public class EmployeeServiceImpl implements 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.java (Test Method):**

java

CopyEdit

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

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

@Autowired

private EmployeeService employeeService;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String... args) {

testGetAllPermanentEmployees();

}

private void testGetAllPermanentEmployees() {

LOGGER.info("Start");

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

for (Employee e : employees) {

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

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

e.getSkillList().forEach(skill -> LOGGER.debug("Skill: {}", skill.getName()));

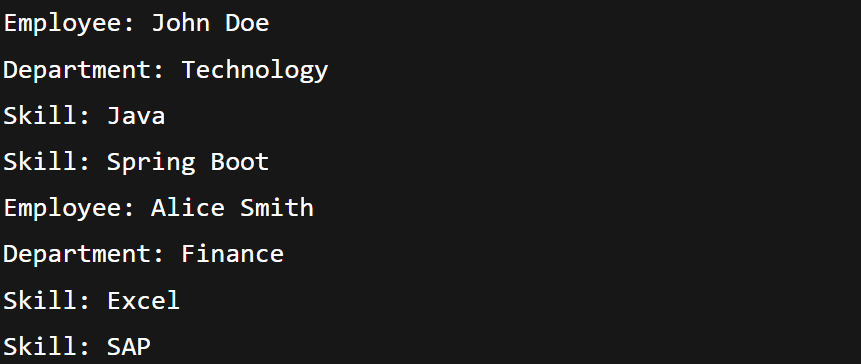
}

LOGGER.info("End");

}

}

**Output:**



**Generated SQL Query:**

select employee0\_.em\_id as em\_id1\_2\_0\_,

department1\_.dp\_id as dp\_id1\_1\_1\_,

skill3\_.sk\_id as sk\_id1\_4\_2\_,

employee0\_.em\_date\_of\_birth as em\_date\_2\_2\_0\_,

employee0\_.em\_dp\_id as em\_dp\_id6\_2\_0\_,

employee0\_.em\_name as em\_name3\_2\_0\_,

employee0\_.em\_permanent as em\_perma4\_2\_0\_,

employee0\_.em\_salary as em\_salar5\_2\_0\_,

department1\_.dp\_name as dp\_name2\_1\_1\_,

skill3\_.sk\_name as sk\_name2\_4\_2\_,

skilllist2\_.es\_em\_id as es\_em\_id1\_3\_0\_\_,

skilllist2\_.es\_sk\_id as es\_sk\_id2\_3\_0\_\_

from employee employee0\_

left outer join department department1\_ on employee0\_.em\_dp\_id = department1\_.dp\_id

left outer join employee\_skill skilllist2\_ on employee0\_.em\_id = skilllist2\_.es\_em\_id

left outer join skill skill3\_ on skilllist2\_.es\_sk\_id = skill3\_.sk\_id

where employee0\_.em\_permanent = 1

**output:**

