**pom.xml dependencies:**

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

**application.properties:**

spring.datasource.url=jdbc:mysql://localhost:3306/ormlearn

spring.datasource.username=root

spring.datasource.password=yourpassword

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

**Hands-on 2: Get All Permanent Employees**

package com.example.ormlearn.model;

import javax.persistence.\*;

import java.time.LocalDate;

import java.util.List;

@Entity

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private boolean permanent;

private double salary;

private LocalDate dateOfBirth;

@ManyToOne(fetch = FetchType.LAZY)

@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 List<Skill> skillList;

// Getters, Setters, toString

}

**Department.java**

package com.example.ormlearn.model;

import javax.persistence.\*;

@Entity

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

// Getters, Setters, toString

}

**Skill.java**

package com.example.ormlearn.model;

import javax.persistence.\*;

@Entity

public class Skill {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

// Getters, Setters, toString

}

**EmployeeRepository.java**

package com.example.ormlearn.repository;

import com.example.ormlearn.model.Employee;

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

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

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

import org.springframework.stereotype.Repository;

import java.util.List;

@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();

@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.java**

package com.example.ormlearn.service;

import com.example.ormlearn.model.Employee;

import com.example.ormlearn.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 departmentId) {

return employeeRepository.getAverageSalary(departmentId);

}

public List<Employee> getAllEmployeesNative() {

return employeeRepository.getAllEmployeesNative();

}

}

**OrmLearnApplication.java**

package com.example.ormlearn;

import com.example.ormlearn.model.Employee;

import com.example.ormlearn.service.EmployeeService;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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.List;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private EmployeeService employeeService;

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

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

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

testGetAllPermanentEmployees();

testGetAverageSalary();

testGetAllEmployeesNative();

}

private void testGetAllPermanentEmployees() {

LOGGER.info("Start");

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

employees.forEach(e -> {

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

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

});

LOGGER.info("End");

}

private void testGetAverageSalary() {

LOGGER.info("Average Salary: {}", employeeService.getAverageSalary(2));

}

private void testGetAllEmployeesNative() {

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

employees.forEach(e -> LOGGER.debug("Native Employee: {}", e));

}

}

**Hands-on 6: Criteria Query for Product Filtering**

**Entity: Product.java**

package com.example.ormlearn.model;

import javax.persistence.\*;

@Entity

public class Product {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private String brand;

private int ramSize;

private String operatingSystem;

private double cpuSpeed;

private double weight;

// Getters, Setters

}

**Repository: ProductRepository.java**

package com.example.ormlearn.repository;

import com.example.ormlearn.model.Product;

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

import org.springframework.stereotype.Repository;

@Repository

public interface ProductRepository extends JpaRepository<Product, Integer> {

}

**Service: ProductService.java**

package com.example.ormlearn.service;

import com.example.ormlearn.model.Product;

import org.springframework.stereotype.Service;

import javax.persistence.EntityManager;

import javax.persistence.PersistenceContext;

import javax.persistence.criteria.\*;

import java.util.ArrayList;

import java.util.List;

@Service

public class ProductService {

@PersistenceContext

private EntityManager entityManager;

public List<Product> getFilteredProducts(String brand, Integer ramSize, String os) {

CriteriaBuilder cb = entityManager.getCriteriaBuilder();

CriteriaQuery<Product> cq = cb.createQuery(Product.class);

Root<Product> root = cq.from(Product.class);

List<Predicate> predicates = new ArrayList<>();

if (brand != null) {

predicates.add(cb.equal(root.get("brand"), brand));

}

if (ramSize != null) {

predicates.add(cb.equal(root.get("ramSize"), ramSize));

}

if (os != null) {

predicates.add(cb.equal(root.get("operatingSystem"), os));

}

cq.where(predicates.toArray(new Predicate[0]));

return entityManager.createQuery(cq).getResultList();

}

}

**OrmLearnApplication.java**

@Autowired

private ProductService productService;

private void testProductCriteriaQuery() {

List<Product> products = productService.getFilteredProducts("Dell", 16, "Windows 11");

products.forEach(p -> System.out.println("Product: " + p.getName()));

}