**Introduction to HQL and JPQL**

**Employee.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.sql.Date;

@Entity

@Table(name = "employees")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String name;

private double salary;

@Column(name = "date\_of\_birth")

private Date dateOfBirth;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

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

}

}

**EmployeeRepository.java**

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Employee;

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

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

import java.util.List;

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

@Query("SELECT e FROM Employee e WHERE e.salary > :salary")

List<Employee> getEmployeesWithSalaryGreaterThan(double salary);

// HQL

@Query("SELECT e FROM Employee e WHERE e.name LIKE %:keyword%")

List<Employee> searchByName(String keyword);

// Native SQL

@Query(value = "SELECT \* FROM employee WHERE salary < :salary", nativeQuery = true)

List<Employee> getLowSalaryEmployees(double salary);

// Aggregate Function

@Query("SELECT AVG(e.salary) FROM Employee e")

Double getAverageSalary();

}

**Ormapplication**

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Employee;

import com.cognizant.ormlearn.repository.EmployeeRepository;

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.sql.Date;

import java.util.List;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private EmployeeRepository employeeRepository;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

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

Employee e = new Employee();

e.setName("Alice");

e.setSalary(70000);

e.setDateOfBirth(Date.valueOf("1995-05-20"));

employeeRepository.save(e);

System.out.println("\n--- High Salary Employees (JPQL) ---");

List<Employee> highPaid = employeeRepository.getEmployeesWithSalaryGreaterThan(60000);

highPaid.forEach(System.out::println);

System.out.println("\n--- Search Employees by Name ---");

employeeRepository.searchByName("Ali").forEach(System.out::println);

System.out.println("\n--- Low Salary Employees (Native SQL) ---");

employeeRepository.getLowSalaryEmployees(80000).forEach(System.out::println);

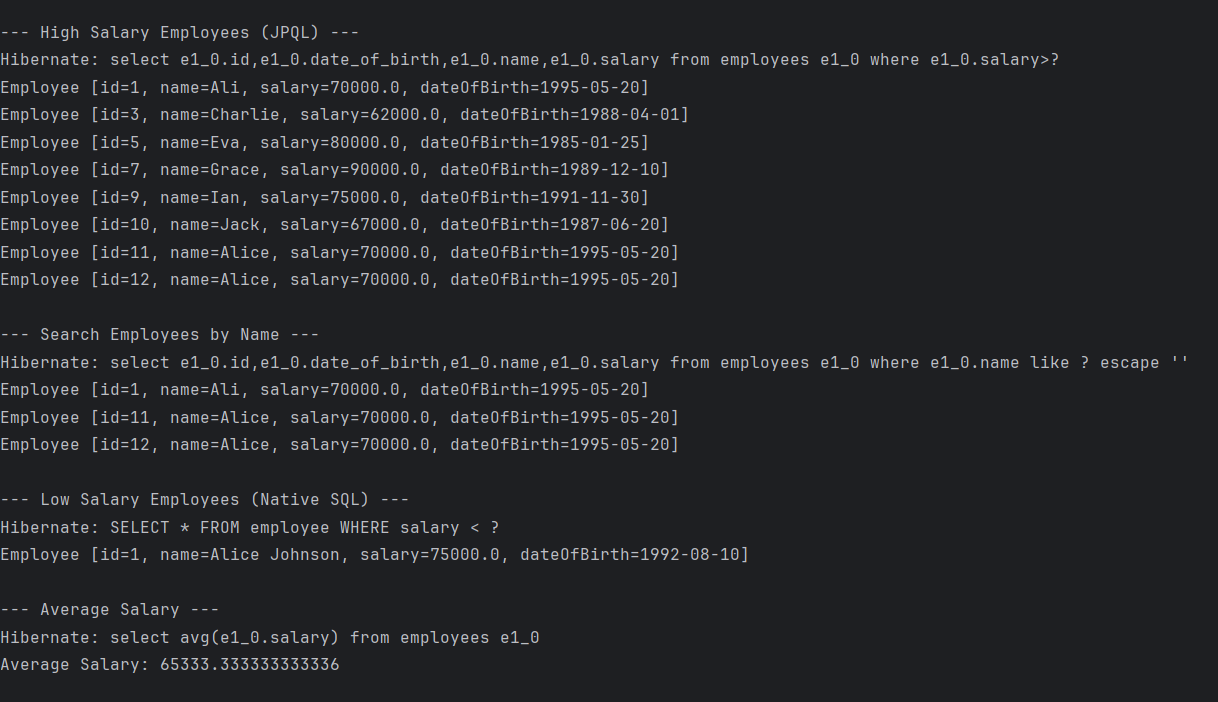
System.out.println("\n--- Average Salary ---");

System.out.println("Average Salary: " + employeeRepository.getAverageSalary());

}

}

**Output:**



**Get all permanent employees using HQL** 

**Department.java**

package com.cognizant.ormlearn.model;

import java.util.\*;

import jakarta.persistence.\*;

@Entity

@Table(name = "department")

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

@Column(name = "dp\_id")

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 + "]";

}

}

**Employee.java**

package com.cognizant.ormlearn.model;

import tajakarta.persistence.\*;

import java.sql.Date;

import java.util.List;

@Entity

@Table(name = "employee\_new")

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

@JoinTable(name = "employee\_skill",

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

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

private List<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 List<Skill> getSkillList() { return skillList; }

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

@Override

public String toString() {

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

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

}

}

**Skill.java**

package com.cognizant.ormlearn.model;

import java.util.\*;

import jakarta.persistence.\*;

@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;

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 + "]";

}

}

**EmployeeRepo**

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Employee;

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

import org.springframework.data.repository.CrudRepository;

import java.util.List;

public interface EmployeeRepository extends CrudRepository<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();

}

DepartmentRepo

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Department;

import org.springframework.data.repository.CrudRepository;

public interface DepartmentRepository extends CrudRepository<Department, Integer> {

}

SkillRepo

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Skill;

import org.springframework.data.repository.CrudRepository;

public interface SkillRepository extends CrudRepository<Skill, Integer> {

}

**Ormapplication**

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Employee;

import com.cognizant.ormlearn.service.EmployeeService;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import java.util.List;

@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();

}

public void testGetAllPermanentEmployees() {

LOGGER.info("Start");

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

employees.forEach(e -> {

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

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

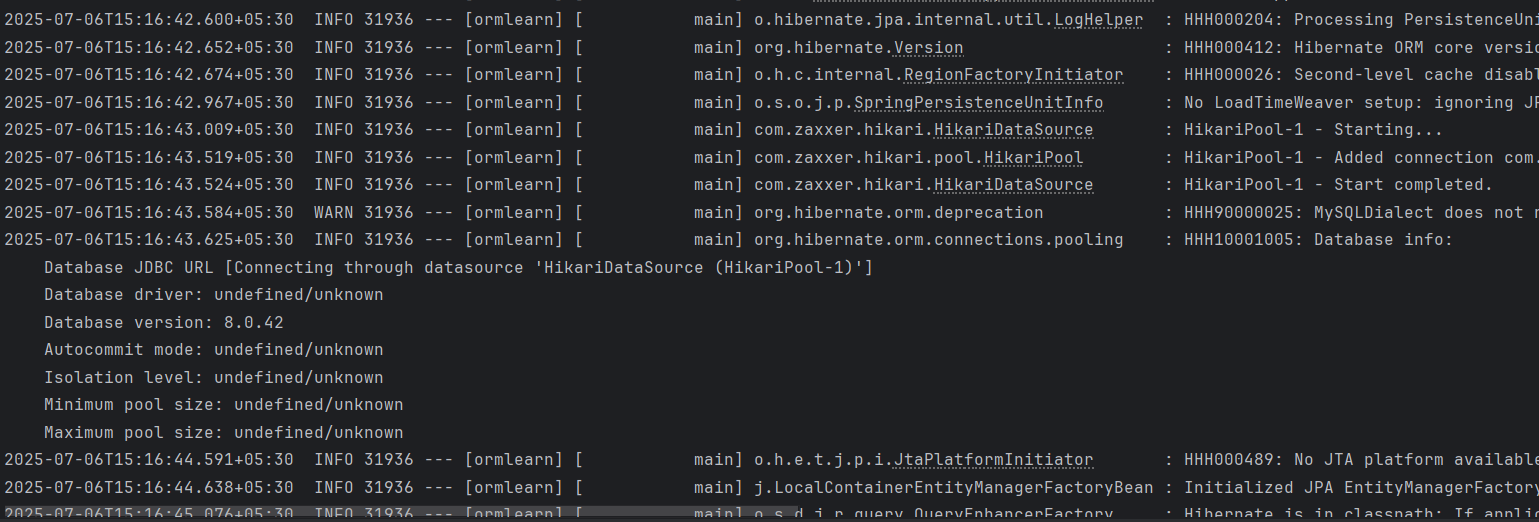
LOGGER.debug("Skills: {}", e.getSkillList());

});

LOGGER.info("End");

}

}



**Fetch quiz attempt details using HQL**

**Attempt.java**

package com.cognizant.ormlearn.model;

import java.util.List;

import java.time.LocalDateTime;

import jakarta.persistence.\*;

@Entity

@Table(name = "attempt")

public class Attempt {

@Id

private int id;

@ManyToOne

@JoinColumn(name = "user\_id")

private User user;

@Column(name = "attempt\_date")

private LocalDateTime attemptDate;

@OneToMany(mappedBy = "attempt", fetch = FetchType.LAZY)

private List<AttemptQuestion> attemptQuestions;

public int getId() {

return id;

}

public User getUser() {

return user;

}

public LocalDateTime getAttemptDate() {

return attemptDate;

}

public List<AttemptQuestion> getAttemptQuestions() {

return attemptQuestions;

}

public void setId(int id) {

this.id = id;

}

public void setUser(User user) {

this.user = user;

}

public void setAttemptDate(LocalDateTime attemptDate) {

this.attemptDate = attemptDate;

}

public void setAttemptQuestions(List<AttemptQuestion> attemptQuestions) {

this.attemptQuestions = attemptQuestions;

}

@Override

public String toString() {

return "Attempt{" +

"id=" + id +

", user=" + (user != null ? user.getUsername() : "null") +

", attemptDate=" + attemptDate +

'}';

}

}

**AttemptQuestion.java**

package com.cognizant.ormlearn.model;

import java.util.\*;

import java.time.LocalDateTime;

import jakarta.persistence.Table;

import jakarta.persistence.Entity;

import jakarta.persistence.\*;

@Entity

@Table(name = "attempt\_question")

public class AttemptQuestion {

@Id

private int id;

@ManyToOne

@JoinColumn(name = "attempt\_id")

private Attempt attempt;

@ManyToOne

@JoinColumn(name = "question\_id")

private Question question;

@OneToMany(mappedBy = "attemptQuestion", fetch = FetchType.LAZY)

private List<AttemptOption> attemptOptions;

}

**AttemptOption.java**

package com.cognizant.ormlearn.model;

import java.util.\*;

import java.time.LocalDateTime;

import jakarta.persistence.Table;

import jakarta.persistence.Entity;

import jakarta.persistence.\*;

@Entity

@Table(name = "attempt\_option")

public class AttemptOption {

@Id

private int id;

@ManyToOne

@JoinColumn(name = "attempt\_question\_id")

private AttemptQuestion attemptQuestion;

@ManyToOne

@JoinColumn(name = "option\_id")

private Option option;

}

**Option.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.Entity;

import jakarta.persistence.\*;

import java.time.LocalDateTime;

import jakarta.persistence.Table;

@Entity

@Table(name = "option1")

public class Option {

@Id

private int id;

private String text;

@Column(name = "is\_correct")

private boolean correct;

@ManyToOne

@JoinColumn(name = "question\_id")

private Question question;

}

**Question.java**

package com.cognizant.ormlearn.model;

import java.util.\*;

import jakarta.persistence.Table;

import jakarta.persistence.Entity;

import jakarta.persistence.\*;

import java.time.LocalDateTime;

@Entity

@Table(name = "question")

public class Question {

@Id

private int id;

private String text;

private int score;

@OneToMany(mappedBy = "question")

private List<Option> options;

}

**AttemptRepo**

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Attempt;

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

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

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

public interface AttemptRepository extends JpaRepository<Attempt, Integer> {

@Query("""

SELECT DISTINCT a FROM Attempt a

JOIN FETCH a.user u

JOIN FETCH a.attemptQuestions aq

JOIN FETCH aq.question q

JOIN FETCH q.options o

JOIN FETCH aq.attemptOptions ao

JOIN FETCH ao.option opt

WHERE u.id = :userId AND a.id = :attemptId

""")

Attempt getAttempt(@Param("userId") int userId, @Param("attemptId") int attemptId);

}

**Ormapp**

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.\*;

import com.cognizant.ormlearn.service.AttemptService;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private AttemptService attemptService;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String... args) {

testAttemptDetails();

}

private void testAttemptDetails() {

Attempt attempt = attemptService.getAttempt(1, 3); // Example values

System.out.println("Username: " + attempt.getUser().getUsername());

System.out.println("Attempted On: " + attempt.getAttemptDate());

for (AttemptQuestion aq : attempt.getAttemptQuestions()) {

Question question = aq.getQuestion();

System.out.println("\nQ: " + question.getText());

for (Option option : question.getOptions()) {

boolean selected = aq.getAttemptOptions().stream()

.anyMatch(ao -> ao.getOption().getId() == option.getId());

String mark = selected ? "[Selected]" : "";

String score = option.isCorrect() ? String.valueOf(question.getScore()) : "0";

System.out.printf("- %s [Score: %s] %s\n", option.getText(), score, mark);

}

}

}

}