**Hands-on 1: Introduction to HQL and JPQL**

**Overview**

* **HQL** stands for **Hibernate Query Language**
* **JPQL** stands for **Java Persistence Query Language**
* Both HQL and JPQL are **object-focused** query languages, similar in syntax to SQL
* **JPQL is a subset of HQL**
* All **JPQL queries** are valid **HQL queries**, but **not all HQL queries** are valid JPQL queries
* Both HQL and JPQL support the following operations:
  + **SELECT**
  + **UPDATE**
  + **DELETE**
* **HQL** additionally supports the **INSERT** statement (which is not available in JPQL)

**Key Differences**

| **Feature** | **HQL** | **JPQL** |
| --- | --- | --- |
| Full Form | Hibernate Query Language | Java Persistence Query Language |
| Focus | Hibernate-specific ORM queries | JPA-standard ORM queries |
| INSERT Statement | Supported | Not Supported |
| Compliance | Vendor-specific (Hibernate) | JPA specification compliant |
| Usage Context | Hibernate | Any JPA implementation |

**Example Syntax Comparison**

**JPQL / HQL SELECT:**

SELECT e FROM Employee e WHERE e.department.name = 'HR'

**JPQL / HQL UPDATE:**

UPDATE Employee e SET e.salary = e.salary + 1000 WHERE e.department.id = :id

**JPQL / HQL DELETE:**

DELETE FROM Employee e WHERE e.permanent = false

**HQL-only INSERT:**

INSERT INTO EmployeeBackup (id, name, salary)

SELECT e.id, e.name, e.salary FROM Employee e WHERE e.permanent = false

**Conclusion**

* Use **JPQL** when working with JPA-compliant providers (e.g., EclipseLink, Hibernate in JPA mode).
* Use **HQL** when using Hibernate's native ORM capabilities.
* Prefer JPQL for **portability**, HQL for **Hibernate-specific features** like INSERT.

**Hands on 2**

**Get all permanent employees using HQL**

**Department.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.List;

*@Entity*

public class Department {

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

*@Column*(name = "dp\_id")

private int id;

*@Column*(name = "dp\_name")

private String name;

*@OneToMany*(mappedBy = "department")

private List<Employee> employeeList;

// Getters and Setters

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

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

}

**Employee.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.Date;

import java.util.List;

*@Entity*

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;

*@Temporal*(*TemporalType*.***DATE***)

*@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;

// Getters and Setters

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; }

}

**Skill.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.List;

*@Entity*

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 List<Employee> employeeList;

// Getters and Setters

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

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

}

**EmployeeRepository.java**

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();

}

**EmployeeService.java**

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Employee;

import com.cognizant.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;

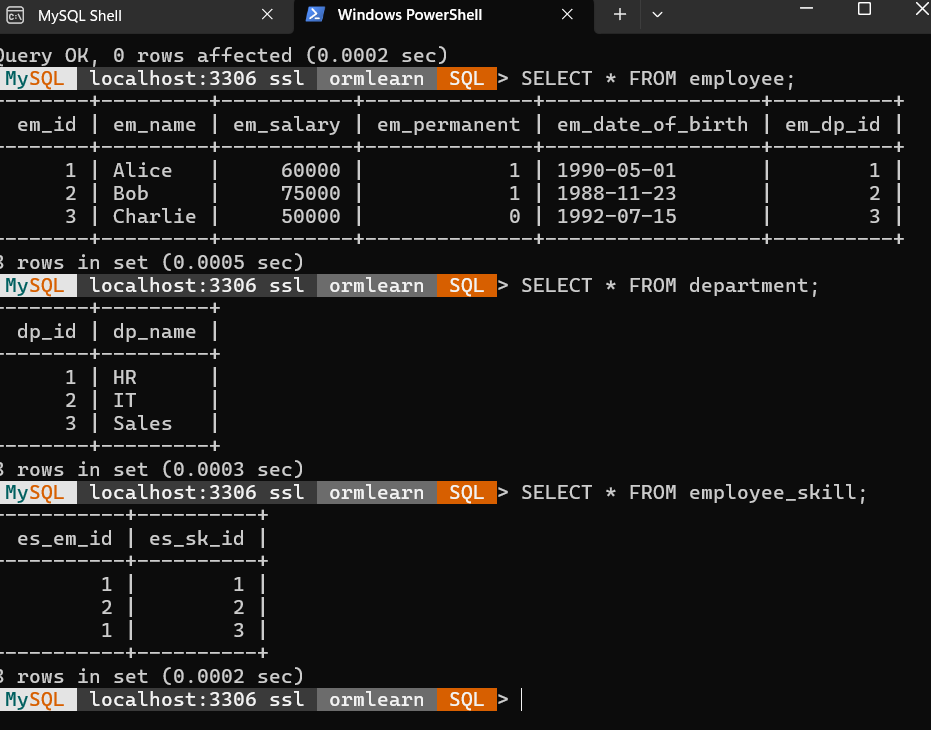
public List<Employee> getAllPermanentEmployees() {

return employeeRepository.getAllPermanentEmployees();

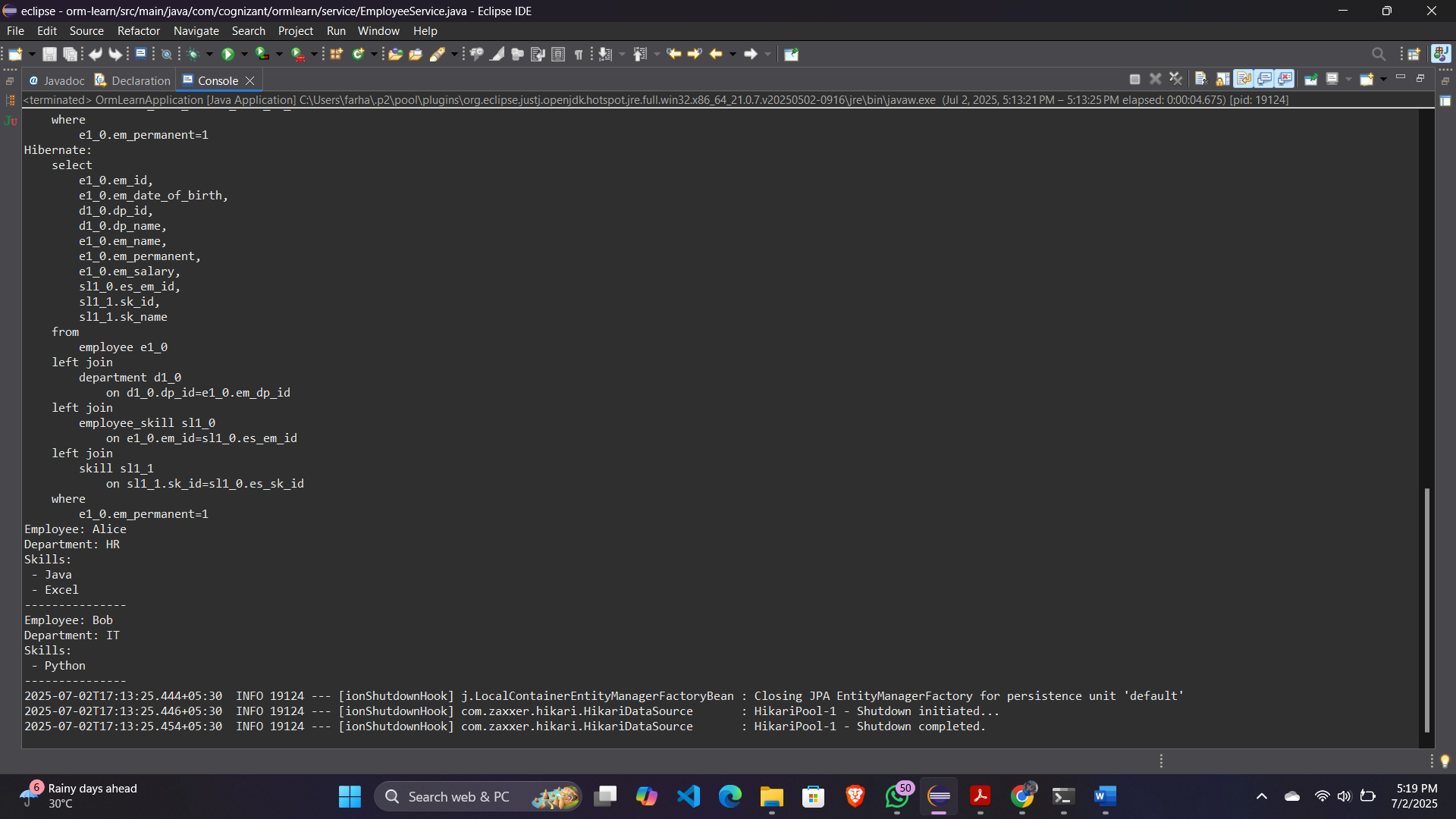
}

}

**MySQL Database:**

****

**Output:**



**Hands on 3**

**Fetch quiz attempt details using HQL**

**User.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.Set;

@Entity

@Table(name = "user")

public class User {

@Id

private int id;

private String username;

@OneToMany(mappedBy = "user")

private Set<Attempt> attempts;

public int getId() { return id; }

public String getUsername() { return username; }

}

**Question.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.Set;

@Entity

public class Question {

@Id

private int id;

private String description;

private double score;

@OneToMany(mappedBy = "question")

private Set<QuizOption> options;

public int getId() { return id; }

public String getDescription() { return description; }

public double getScore() { return score; }

public Set<QuizOption> getOptions() { return options; }

}

**QuizOption.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

@Entity

@Table(name = "quiz\_option")

public class QuizOption {

@Id

private int id;

private String text;

private boolean correct;

@ManyToOne

@JoinColumn(name = "question\_id")

private Question question;

public int getId() { return id; }

public String getText() { return text; }

public boolean isCorrect() { return correct; }

}

**Attempt.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.Date;

import java.util.Set;

@Entity

public class Attempt {

@Id

private int id;

@Temporal(TemporalType.TIMESTAMP)

private Date attemptedDate;

@ManyToOne

@JoinColumn(name = "user\_id")

private User user;

@OneToMany(mappedBy = "attempt")

private Set<AttemptQuestion> attemptQuestions;

public int getId() { return id; }

public Date getAttemptedDate() { return attemptedDate; }

public User getUser() { return user; }

public Set<AttemptQuestion> getAttemptQuestions() { return attemptQuestions; }

}

**AttemptQuestion.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

import java.util.Set;

@Entity

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")

private Set<AttemptOption> attemptOptions;

public int getId() { return id; }

public Question getQuestion() { return question; }

public Set<AttemptOption> getAttemptOptions() { return attemptOptions; }

}

**AttemptOption.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

@Entity

public class AttemptOption {

@Id

private int id;

private boolean selected;

@ManyToOne

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

private AttemptQuestion attemptQuestion;

@ManyToOne

@JoinColumn(name = "option\_id")

private QuizOption option;

public boolean isSelected() { return selected; }

public QuizOption getOption() { return option; }

}

**AttemptRepository.java**

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Attempt;

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

import org.springframework.data.repository.CrudRepository;

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

public interface AttemptRepository extends CrudRepository<Attempt, Integer> {

@Query("SELECT 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 op " +

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

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

}

**AttemptService.java**

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Attempt;

import com.cognizant.ormlearn.repository.AttemptRepository;

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

import org.springframework.stereotype.Service;

@Service

public class AttemptService {

@Autowired

private AttemptRepository attemptRepository;

public Attempt getAttempt(int userId, int attemptId) {

return attemptRepository.getAttempt(userId, attemptId);

}

}

**OrmLearnApplication.java**

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Attempt;

import com.cognizant.ormlearn.model.AttemptQuestion;

import com.cognizant.ormlearn.model.AttemptOption;

import com.cognizant.ormlearn.model.Question;

import com.cognizant.ormlearn.model.QuizOption;

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;

import java.util.Set;

import java.util.stream.Collectors;

@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) throws Exception {

testGetAttemptDetail();

}

private void testGetAttemptDetail() {

int userId = 1;

int attemptId = 1;

Attempt attempt = attemptService.getAttempt(userId, attemptId);

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

System.out.println("Attempted Date: " + attempt.getAttemptedDate());

System.out.println("-------------------------------------------");

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

Question q = aq.getQuestion();

System.out.println(q.getDescription());

// Get selected option IDs

Set<Integer> selectedOptionIds = aq.getAttemptOptions().stream()

.filter(AttemptOption::isSelected)

.map(ao -> ao.getOption().getId())

.collect(Collectors.toSet());

for (QuizOption option : q.getOptions()) {

int optionId = option.getId();

String optionText = option.getText();

double score = q.getScore();

boolean selected = selectedOptionIds.contains(optionId);

System.out.printf("%d) %s\t%.1f\t%s\n", optionId, optionText, score, selected);

}

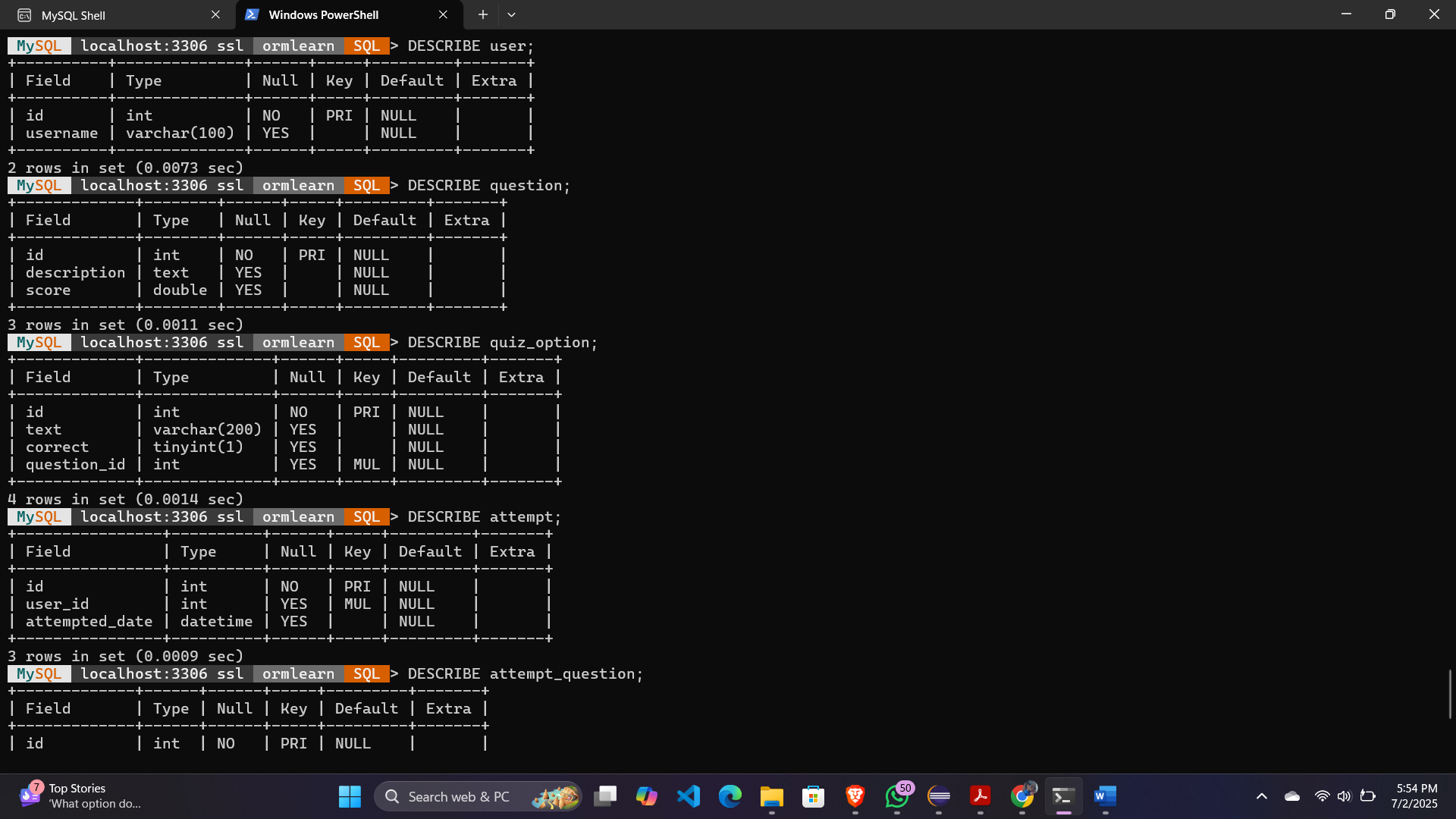
System.out.println();

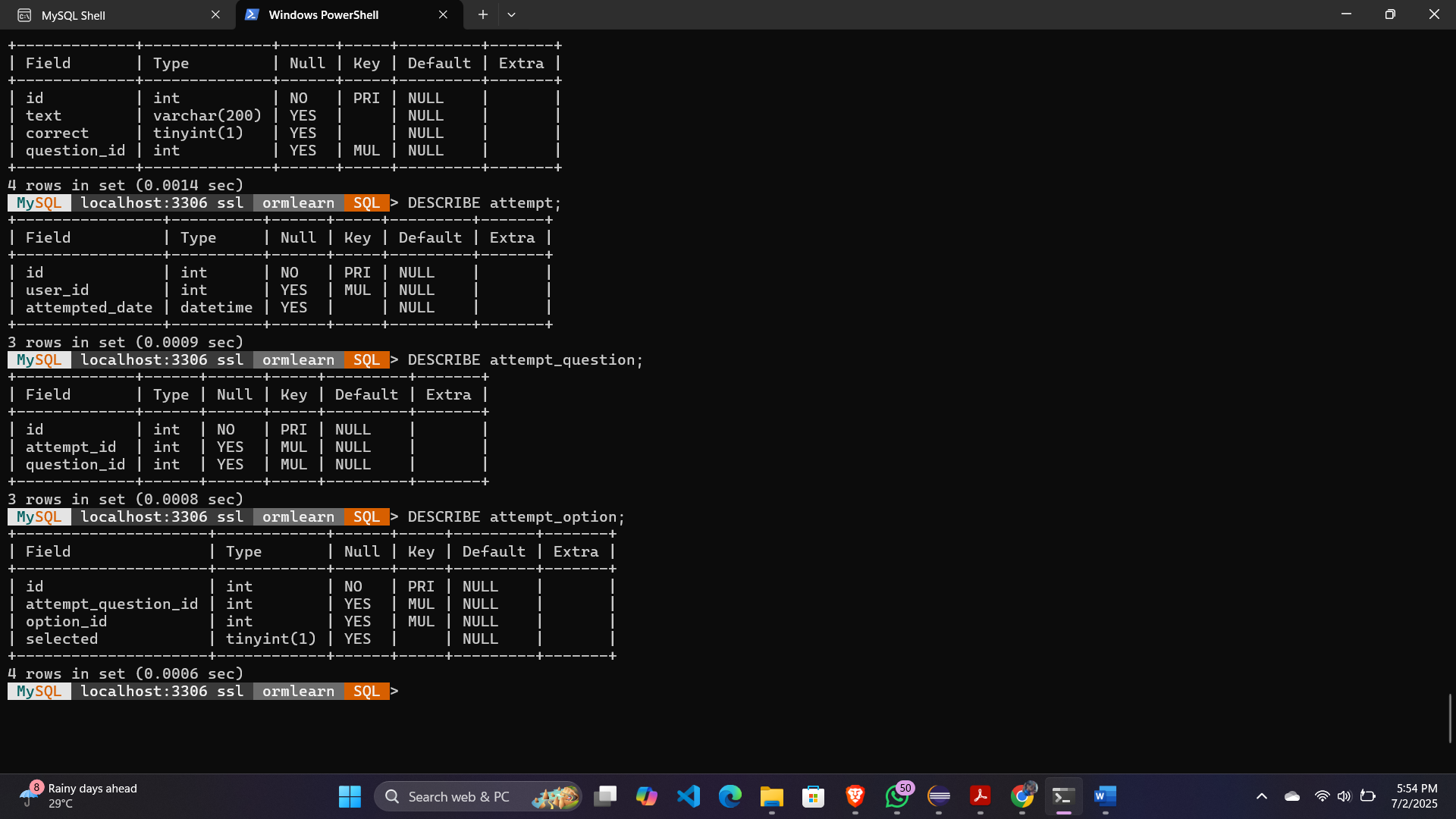
}

}

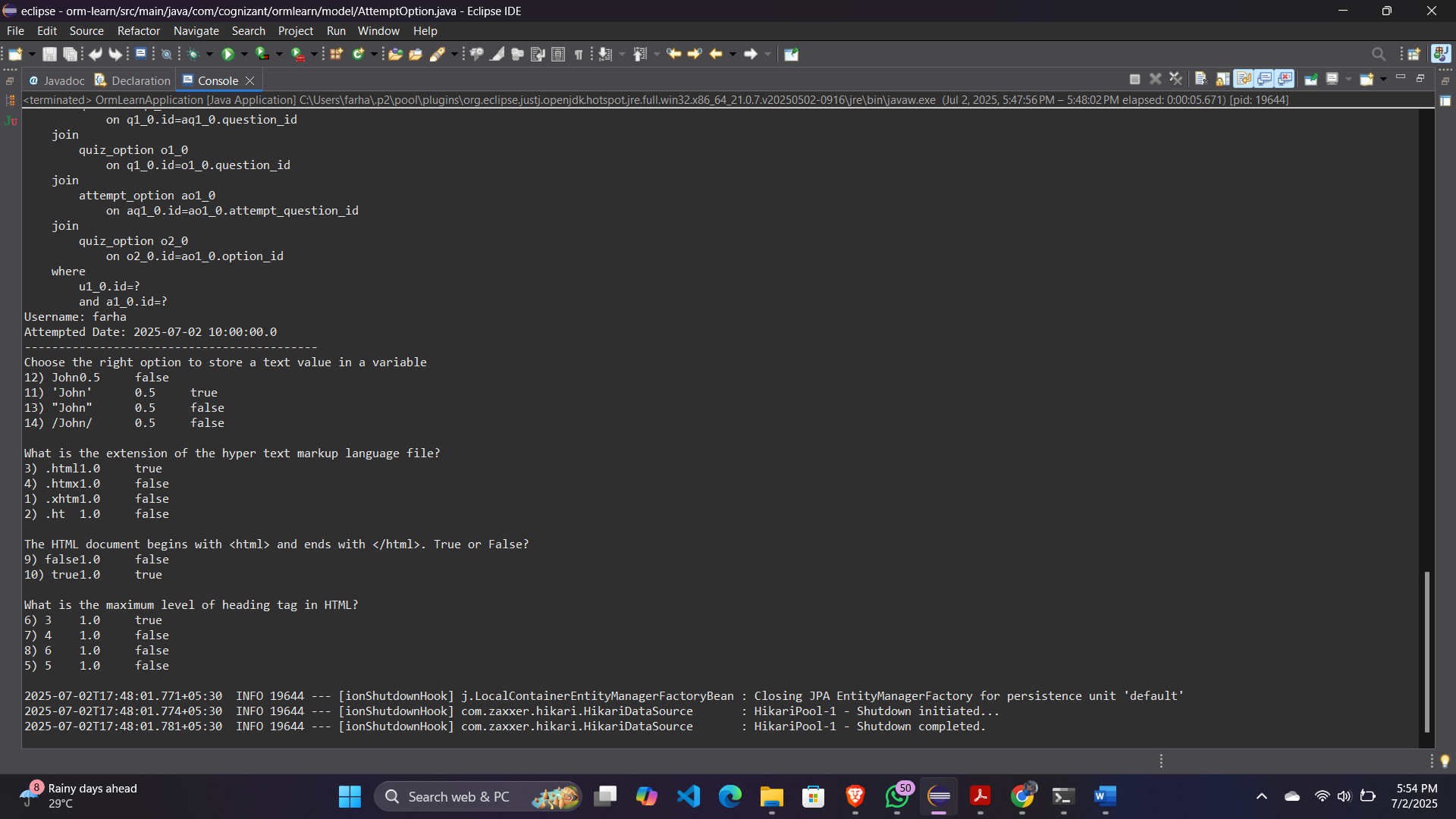
}

**MySQL Database:**





**Output:**



**Hands on 4**

**Get average salary using HQL**

**Employee.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

@Entity

public class Employee {

@Id

@Column(name = "em\_id")

private int id;

@Column(name = "em\_name")

private String name;

@Column(name = "em\_salary")

private double salary;

@ManyToOne

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

private Department department;

// getters

public int getId() { return id; }

public double getSalary() { return salary; }

public Department getDepartment() { return department; }

}

**Departmenr.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

@Entity

public class Department {

@Id

@Column(name = "dp\_id")

private int id;

@Column(name = "dp\_name")

private String name;

public int getId() { return id; }

public String getName() { return name; }

}

**EmployeeRepository.java**

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

public interface EmployeeRepository extends CrudRepository<Employee, Integer> {

@Query("SELECT AVG(e.salary) FROM Employee e WHERE e.department.id = :id")

double getAverageSalary(@Param("id") int departmentId);

}

**EmployeeService.java**

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.repository.EmployeeRepository;

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

import org.springframework.stereotype.Service;

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository employeeRepository;

public double getAverageSalary(int departmentId) {

return employeeRepository.getAverageSalary(departmentId);

}

}

**OrmLearnApplication.java**

package com.cognizant.ormlearn;

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;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private EmployeeService employeeService;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

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

testGetAverageSalary();

}

private void testGetAverageSalary() {

int departmentId = 1; // Change this ID based on your database data

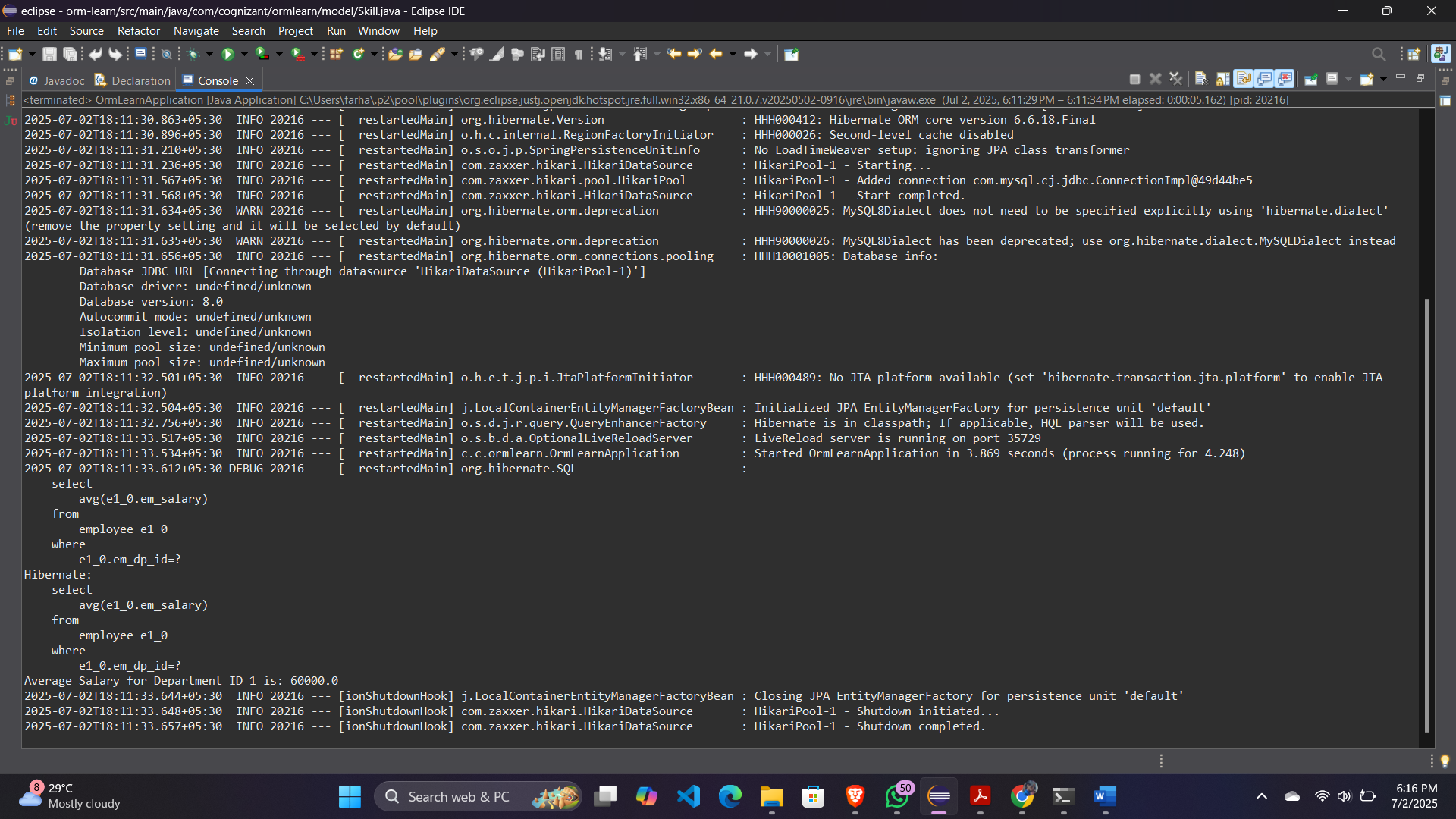
double avgSalary = employeeService.getAverageSalary(departmentId);

System.out.println("Average Salary for Department ID " + departmentId + " is: " + avgSalary);

}

}

**Output:**



**Hands on 5**

**Get all employees using Native Query**

**Employee.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

@Entity

@Table(name = "employee")

public class Employee {

@Id

@Column(name = "em\_id")

private int id;

@Column(name = "em\_name")

private String name;

@Column(name = "em\_salary")

private double salary;

public int getId() {

return id;

}

public String getName() {

return name;

}

public double getSalary() {

return salary;

}

}

**EmployeeRepository.java**

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(value = "SELECT \* FROM employee", nativeQuery = true)

List<Employee> getAllEmployeesNative();

}

**EmployeeService.java**

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Employee;

import com.cognizant.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> getAllEmployeesNative() {

return employeeRepository.getAllEmployeesNative();

}

}

**OrmLearnApplication.java**

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 java.util.List;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private EmployeeService employeeService;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

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

testGetAllEmployeesNative();

}

private void testGetAllEmployeesNative() {

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

for (Employee emp : employeeList) {

System.out.println("ID: " + emp.getId() +

", Name: " + emp.getName() +

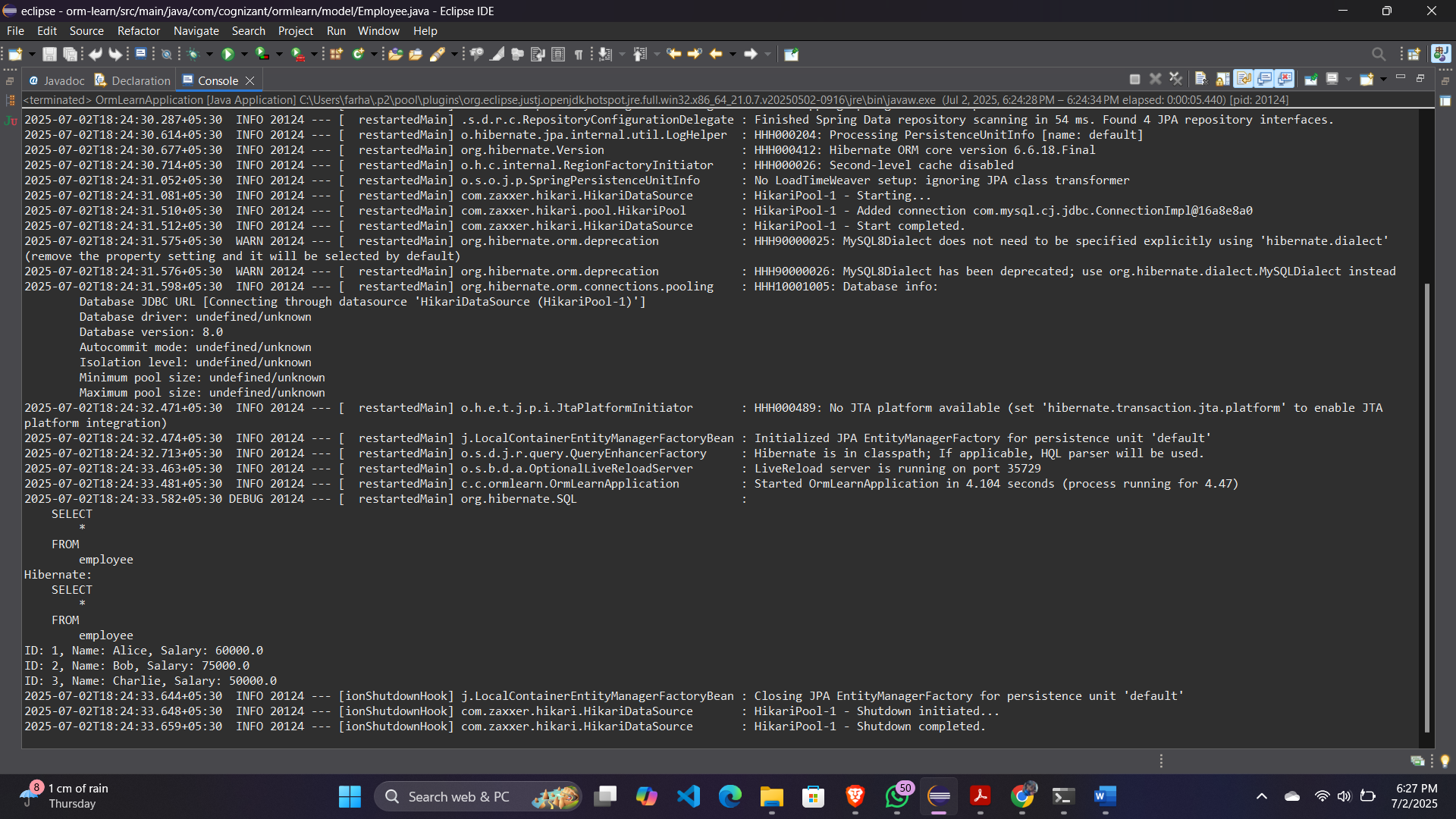
", Salary: " + emp.getSalary());

}

}

}

**Output:**



**Hands on 6 Criteria Query**

**Product.java**

package com.cognizant.ormlearn.model;

import jakarta.persistence.\*;

@Entity

public class Product {

@Id

private int id;

private String name;

private String os;

private String cpu;

private double ram;

private double hdd;

private double weight;

public String getName() { return name; }

}

**ProductRepositoryCustom.java**

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Product;

import java.util.List;

import java.util.Map;

public interface ProductRepositoryCustom {

List<Product> filterProducts(Map<String, Object> filters);

}

**ProductRepositoryCustomImpl.java**

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Product;

import jakarta.persistence.EntityManager;

import jakarta.persistence.PersistenceContext;

import jakarta.persistence.criteria.\*;

import org.springframework.stereotype.Repository;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

@Repository

public class ProductRepositoryCustomImpl implements ProductRepositoryCustom {

@PersistenceContext

private EntityManager em;

@Override

public List<Product> filterProducts(Map<String, Object> filters) {

CriteriaBuilder cb = em.getCriteriaBuilder();

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

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

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

if (filters.containsKey("cpu")) {

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

}

if (filters.containsKey("os")) {

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

}

if (filters.containsKey("ram")) {

predicates.add(cb.ge(root.get("ram"), (Double) filters.get("ram")));

}

if (filters.containsKey("weight")) {

predicates.add(cb.le(root.get("weight"), (Double) filters.get("weight")));

}

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

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

}

}

**ProductRepository.java**

package com.cognizant.ormlearn.repository;

import com.cognizant.ormlearn.model.Product;

import org.springframework.data.repository.CrudRepository;

public interface ProductRepository extends CrudRepository<Product, Integer>, ProductRepositoryCustom {

}

**ProductService.java**

package com.cognizant.ormlearn.service;

import com.cognizant.ormlearn.model.Product;

import com.cognizant.ormlearn.repository.ProductRepository;

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

import org.springframework.stereotype.Service;

import java.util.List;

import java.util.Map;

@Service

public class ProductService {

@Autowired

private ProductRepository productRepository;

public List<Product> filterProducts(Map<String, Object> filters) {

return productRepository.filterProducts(filters);

}

}

**OrmLearnApplication.java**

package com.cognizant.ormlearn;

import com.cognizant.ormlearn.model.Product;

import com.cognizant.ormlearn.service.ProductService;

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.HashMap;

import java.util.List;

import java.util.Map;

@SpringBootApplication

public class OrmLearnApplication implements CommandLineRunner {

@Autowired

private ProductService productService;

public static void main(String[] args) {

SpringApplication.run(OrmLearnApplication.class, args);

}

@Override

public void run(String... args) {

testDynamicProductFilter();

}

private void testDynamicProductFilter() {

Map<String, Object> filters = new HashMap<>();

filters.put("cpu", "Intel i5");

filters.put("ram", 8.0);

filters.put("weight", 2.5);

List<Product> result = productService.filterProducts(filters);

for (Product p : result) {

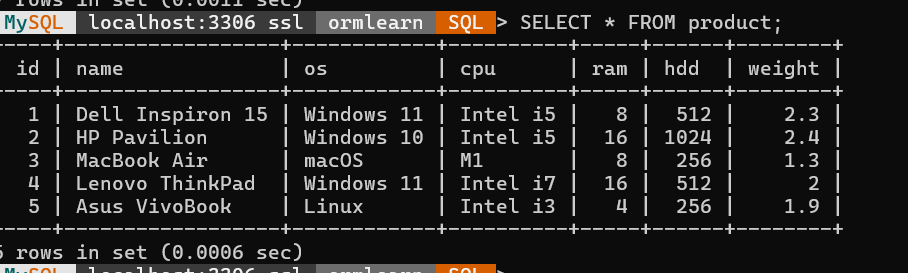
System.out.println("Product: " + p.getName());

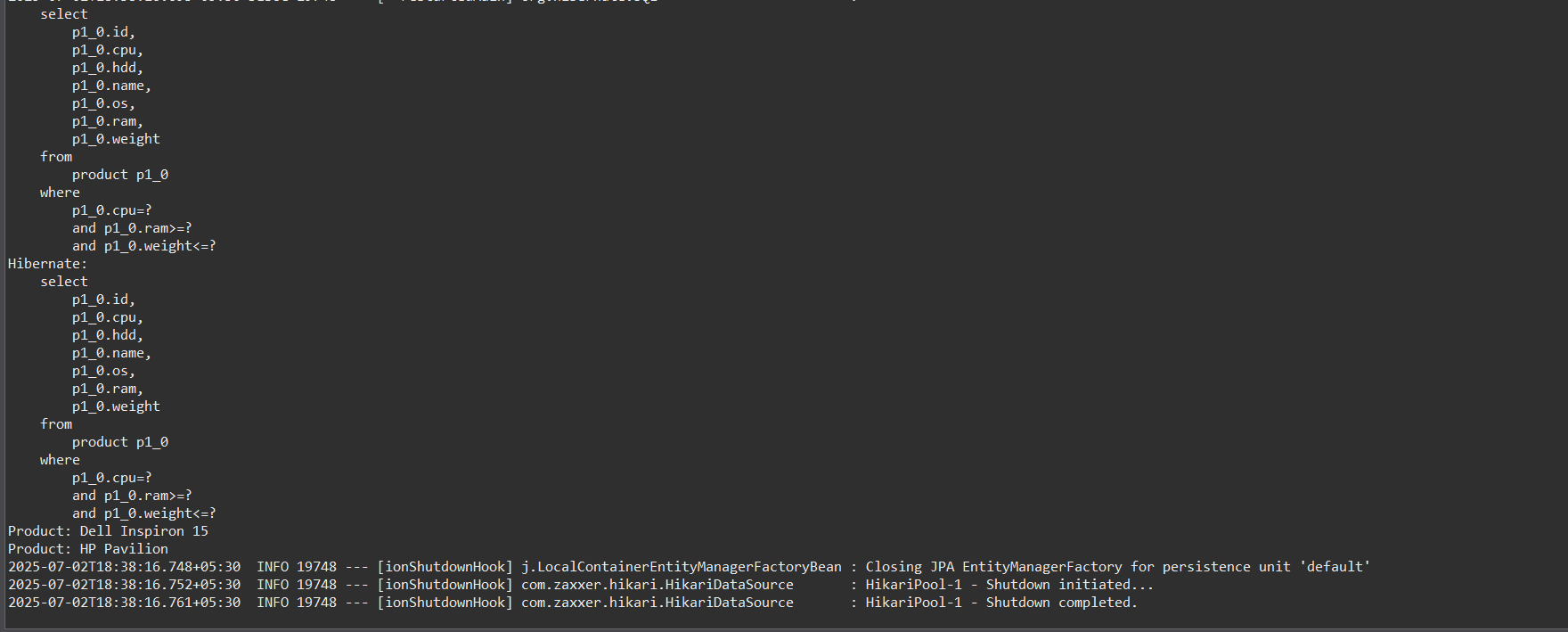
}

}

}

**MySQL Database:**

****

**Output:**