

### 3. spring-data-jpa-handson

## Hands on 1

## Introduction to HQL and JPQL

The screenshot shows the Eclipse IDE with the `HqJpqlLearnApplication.java` file open. The code defines a `run` method that calls `testGetAllEmployees()`, which in turn calls `employeeService.getAllEmployees()` and prints the results. The console output shows the application starting successfully and printing the list of employees:

```
2025-07-11T10:28:02.101+05:30 INFO 20468 --- [hql-jpql-learn] [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set 'hiber
2025-07-11T10:28:02.107+05:30 INFO 20468 --- [hql-jpql-learn] [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persist
2025-07-11T10:28:02.366+05:30 WARN 20468 --- [hql-jpql-learn] [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. T
2025-07-11T10:28:02.753+05:30 INFO 20468 --- [hql-jpql-learn] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with context
2025-07-11T10:28:02.765+05:30 INFO 20468 --- [hql-jpql-learn] [main] c.hql.jpql.demo.HqJpqlLearnApplication : Started HqJpqlLearnApplication in 4.858 seconds

Hibernate: select e1_0.em_id,e1_0.em_dp_id,e1_0.em_name,e1_0.emPermanent,e1_0.em_salary from employee e1_0
Hibernate: select d1_0.dp_id,d1_0.dp_name from department d1_0 where d1_0.dp_id=?
Hibernate: select s11_0.es_em_id,s11_1.sk_id,s11_1.sk_name from employee_skill s11_0 join skill s11_1 on s11_1.sk_id=s11_0.es_sk_id where s11_0.es_em_id=?
Hibernate: select s11_0.es_em_id,s11_1.sk_id,s11_1.sk_name from employee_skill s11_0 join skill s11_1 on s11_1.sk_id=s11_0.es_sk_id where s11_0.es_em_id=?
Employee list ---
101 - John - 50000.0
102 - Jane - 60000.0
103 - David - 70000.0
105 - Updated Name - 90000.0
```

## Hands on 2

## Get all permanent employees using HQL

The screenshot shows the Eclipse IDE with the `HqJpqlLearnApplication.java` file open. The code defines a `run` method that calls `testGetAllPermanentEmployees()`, which in turn calls `employeeService.getAllPermanentEmployees()` and prints the results. The console output shows the application starting successfully and printing the list of permanent employees:

```
2025-07-11T10:57:02.141+05:30 INFO 17476 --- [hql-jpql-learn] [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000489: No JTA platform available (set 'hiber
2025-07-11T10:57:02.148+05:30 INFO 17476 --- [hql-jpql-learn] [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persist
2025-07-11T10:57:02.474+05:30 INFO 17476 --- [hql-jpql-learn] [main] o.s.d.j.r.query.QueryEnhancerFactory : Hibernate is in classpath; If applicable, HQL pa
2025-07-11T10:57:03.129+05:30 INFO 17476 --- [hql-jpql-learn] [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. T
2025-07-11T10:57:03.580+05:30 INFO 17476 --- [hql-jpql-learn] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8081 (http) with context
2025-07-11T10:57:03.590+05:30 INFO 17476 --- [hql-jpql-learn] [main] c.hql.jpql.demo.HqJpqlLearnApplication : Started HqJpqlLearnApplication in 6.111 seconds

Permanent Employees ---
101 - John - 50000.0 - HR
No Skills Assigned.
103 - David - 70000.0 - IT
No Skills Assigned.
105 - Updated Name - 90000.0 - IT
No Skills Assigned.
```

## Hands on 3

### Fetch quiz attempt details using HQL

The screenshot shows the Eclipse IDE with the `QuizApplication.java` file open. The code is a Spring Boot application that uses JPA and Hibernate. The console output shows the application starting up, with various logs from Spring Boot, Hibernate, and the database driver. The application is running on port 8082.

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;
import org.springframework.boot.autoconfigure.domain.EntityScan;

@SpringBootApplication(scanBasePackages = "com.quiz")
@EnableJpaRepositories(basePackages = "com.quiz.repository")
@EntityScan(basePackages = "com.quiz.model") // Tells Spring where to find @Entity classes
public class QuizApplication {

    public static void main(String[] args) {
        SpringApplication.run(QuizApplication.class, args);
    }
}
```

Console Output:

```
2025-07-11T17:45:46.393+05:30 INFO 20424 --- [hql-jpql-learn] [main] o.hiber.nate.jpa.interna.util.Loggeipeer : HH000004: Processing persistenceunitinfo [name:
2025-07-11T17:45:46.442+05:30 INFO 20424 --- [hql-jpql-learn] [main] o.hiber.nate.Version : HH0000412: Hibernate ORM core version 6.6.18.Fin
2025-07-11T17:45:46.469+05:30 INFO 20424 --- [hql-jpql-learn] [main] o.h.c.internal.RegionFactoryInitiator : HH0000026: Second-level cache disabled
2025-07-11T17:45:46.848+05:30 INFO 20424 --- [hql-jpql-learn] [main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA class tran
2025-07-11T17:45:46.881+05:30 INFO 20424 --- [hql-jpql-learn] [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2025-07-11T17:45:47.241+05:30 INFO 20424 --- [hql-jpql-learn] [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection com.mysql.cj.jdbc
2025-07-11T17:45:47.243+05:30 INFO 20424 --- [hql-jpql-learn] [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2025-07-11T17:45:47.244+05:30 INFO 20424 --- [hql-jpql-learn] [main] org.hiber.nate.orm.connections.pooling : HH10001005: Database Info:
Database JDBC URL [connecting through datasource 'hikariPool-1']
Database driver: undefined/unknown
Database version: 8.0.42
Autocommit mode: undefined/unknown
Isolation level: undefined/unknown
Minimum pool size: undefined/unknown
Maximum pool size: undefined/unknown
2025-07-11T17:45:48.588+05:30 INFO 20424 --- [hql-jpql-learn] [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HH0000489: No JTA platform available (set 'hiber
2025-07-11T17:45:48.594+05:30 INFO 20424 --- [hql-jpql-learn] [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persist
2025-07-11T17:45:49.155+05:30 WARN 20424 --- [hql-jpql-learn] [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Th
2025-07-11T17:45:49.580+05:30 INFO 20424 --- [hql-jpql-learn] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8082 (http) with context
2025-07-11T17:45:49.593+05:30 INFO 20424 --- [hql-jpql-learn] [main] c.quiz.quizapplication.QuizApplication : Started QuizApplication in 3.993 seconds (proces
```

## Hands on 4

### Get average salary using HQL

The screenshot shows the Eclipse IDE with the `HqlJpqlLearnApplication.java` file open. The code is a Spring Boot application that uses JPA and Hibernate. The console output shows the application starting up, with various logs from Spring Boot, Hibernate, and the database driver. The application is running on port 8082.

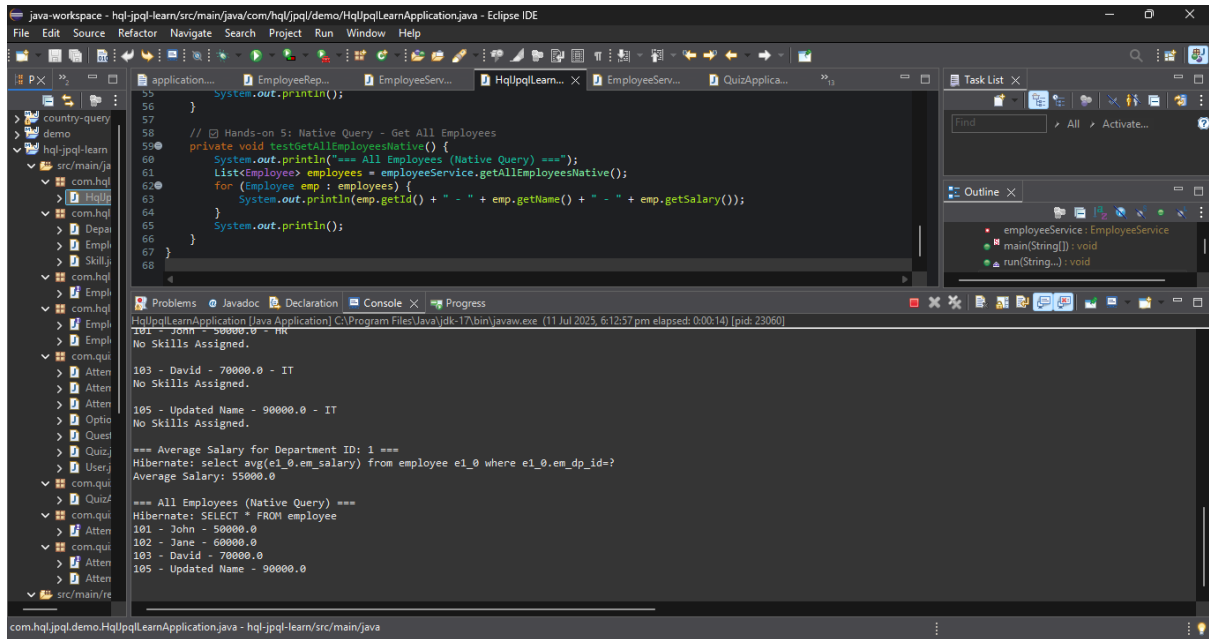
```
private void testGetAverageSalaryByDepartment(int deptId) {
    System.out.println("=== Average Salary for Department ID: " + deptId + " ===");
    double avgSalary = employeeService.getAverageSalary(deptId);
    System.out.println("Average Salary: " + avgSalary);
    System.out.println();
}
```

Console Output:

```
2025-07-11T18:07:15.798+05:30 INFO 9220 --- [hql-jpql-learn] [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persist
2025-07-11T18:07:16.421+05:30 WARN 9220 --- [hql-jpql-learn] [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Th
2025-07-11T18:07:16.798+05:30 INFO 9220 --- [hql-jpql-learn] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8082 (http) with context p
2025-07-11T18:07:16.801+05:30 INFO 9220 --- [hql-jpql-learn] [main] c.hql.jpql.demo.HqlJpqlLearnApplication : Started HqlJpqlLearnApplication in 5.882 seconds
=== Permanent Employees ===
Hibernate: select e1_0.em_id,d1_0.dp_id,d1_0.dp_name,e1_0.em_name,e1_0.em_permanent,e1_0.em_salary,s11_0.es_em_id,s11_1.sk_id,s11_1.sk_name from employee e1_0 left join dep
101 - John - 50000.0 - HR
No Skills Assigned.
103 - David - 70000.0 - IT
No Skills Assigned.
105 - Updated Name - 90000.0 - IT
No Skills Assigned.
=== Average Salary for Department ID: 1 ===
Hibernate: select avg(e1_0.em_salary) from employee e1_0 where e1_0.em_dp_id=?
Average Salary: 55000.0
```

## Hands on 5

### Get all employees using Native Query



```
55      System.out.println();
56  }
57
58  // Hands-on 5: Native Query - Get All Employees
59  private void testGetAllEmployeesNative() {
60      System.out.println("=== All Employees (Native Query) ===");
61      List<Employee> employees = employeeService.getAllEmployeesNative();
62      for (Employee emp : employees) {
63          System.out.println(emp.getId() + " - " + emp.getName() + " - " + emp.getSalary());
64      }
65      System.out.println();
66  }
67
68  }
```

Console Output:

```
101 - John - 50000.0 - HR
No Skills Assigned.

103 - David - 70000.0 - IT
No Skills Assigned.

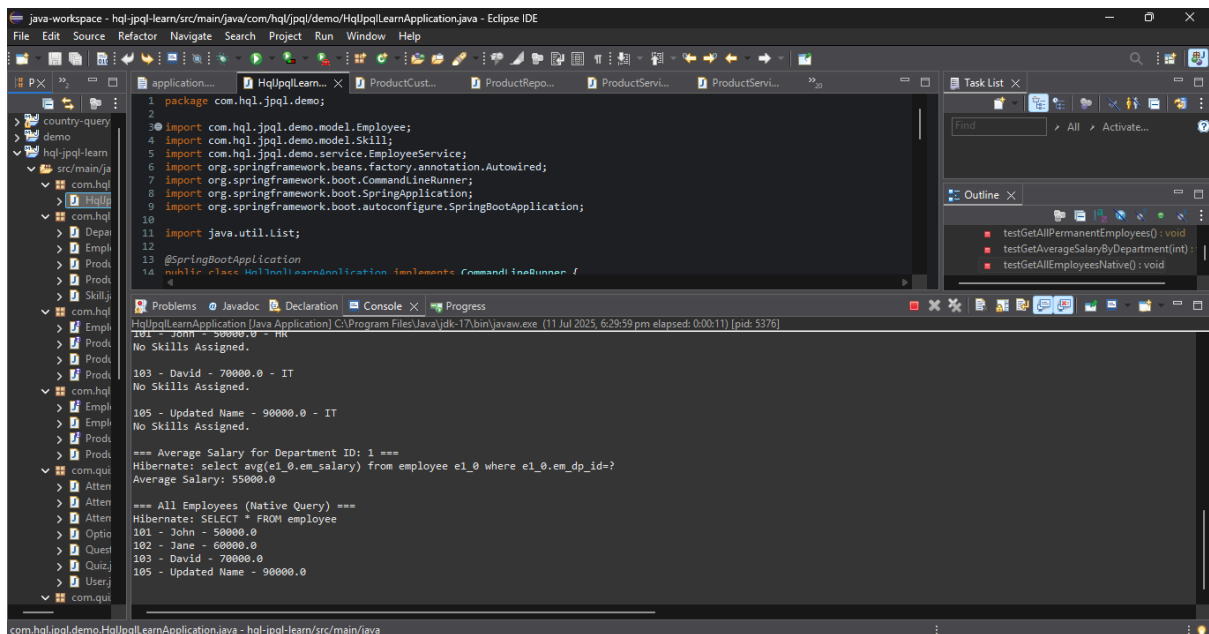
105 - Updated Name - 90000.0 - IT
No Skills Assigned.

=== Average Salary for Department ID: 1 ===
Hibernate: select avg(e1_0.em_salary) from employee e1_0 where e1_0.em_dp_id=?
Average Salary: 55000.0

=== All Employees (Native Query) ===
Hibernate: SELECT * FROM employee
101 - John - 50000.0
102 - Jane - 60000.0
103 - David - 70000.0
105 - Updated Name - 90000.0
```

## Hands on 6

### Criteria Query



```
1 package com.hql.jpql.demo;
2
3 import com.hql.jpql.demo.model.Employee;
4 import com.hql.jpql.demo.model.Skill;
5 import com.hql.jpql.demo.service.EmployeeService;
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.boot.CommandLineRunner;
8 import org.springframework.boot.SpringApplication;
9 import org.springframework.boot.autoconfigure.SpringBootApplication;
10
11 import java.util.List;
12
13 @SpringBootApplication
14 public class HqlJpqlLearnApplication implements CommandLineRunner {
15
16     // ...
17
18     private void testGetAllPermanentEmployees() {
19         // ...
20     }
21
22     // ...
23 }
```

Console Output:

```
101 - John - 50000.0 - HR
No Skills Assigned.

103 - David - 70000.0 - IT
No Skills Assigned.

105 - Updated Name - 90000.0 - IT
No Skills Assigned.

=== Average Salary for Department ID: 1 ===
Hibernate: select avg(e1_0.em_salary) from employee e1_0 where e1_0.em_dp_id=?
Average Salary: 55000.0

=== All Employees (Native Query) ===
Hibernate: SELECT * FROM employee
101 - John - 50000.0
102 - Jane - 60000.0
103 - David - 70000.0
105 - Updated Name - 90000.0
```