Spring Data JPA - Advanced Hands-On Solutions

# Hands on 1: Introduction to HQL and JPQL

HQL (Hibernate Query Language) and JPQL (Java Persistence Query Language) are object-oriented query languages.  
- JPQL is a subset of HQL.  
- Both support SELECT, UPDATE, DELETE.  
- HQL also supports INSERT.  
- Syntax is similar to SQL but operates on entity objects and their properties.

# Hands on 2: Get all permanent employees using HQL

Repository Method:  
@Query("SELECT e FROM Employee e LEFT JOIN FETCH e.department d LEFT JOIN FETCH e.skillList WHERE e.permanent = 1")  
List<Employee> getAllPermanentEmployees();  
  
Service Method:  
public List<Employee> getAllPermanentEmployees() { return employeeRepository.getAllPermanentEmployees(); }  
  
Test Method in Main Class:  
testGetAllPermanentEmployees() - Logs employee and skill details.  
  
Key Point: Use `fetch` with `join` to eagerly load collections efficiently.

# Hands on 3: Fetch quiz attempt details using HQL

- Create entity classes and map relationships based on schema.  
- Repository HQL:  
@Query("SELECT a FROM Attempt a JOIN FETCH a.user JOIN FETCH a.questions q JOIN FETCH q.options WHERE a.user.id = :userId AND a.id = :attemptId")  
Attempt getAttempt(int userId, int attemptId);  
  
- Display format includes question, options, correct answer, score and selected option.

# Hands on 4: Get average salary using HQL

Repository Method:  
@Query("SELECT AVG(e.salary) FROM Employee e WHERE e.department.id = :id")  
double getAverageSalary(@Param("id") int id);  
  
- Usage of aggregate functions like AVG, SUM, MIN, MAX with filtering.

# Hands on 5: Get all employees using Native Query

Repository Method:  
@Query(value="SELECT \* FROM employee", nativeQuery=true)  
List<Employee> getAllEmployeesNative();  
  
- Native queries are DB-specific and less portable. Use them sparingly.

# Hands on 6: Criteria Query

Criteria queries allow dynamic query building:  
- Useful when filtering conditions vary.  
- Uses CriteriaBuilder, CriteriaQuery, Root, and TypedQuery.  
- Example:  
CriteriaBuilder cb = entityManager.getCriteriaBuilder();  
CriteriaQuery<Product> cq = cb.createQuery(Product.class);  
Root<Product> root = cq.from(Product.class);  
List<Predicate> predicates = new ArrayList<>();  
if (filter.hasRating()) predicates.add(cb.equal(root.get("rating"), filter.getRating()));  
...  
cq.where(cb.and(predicates.toArray(new Predicate[0])));