**JPA Criteria API vs Query API (JPQL & Native Queries)**

**When working with JPA, you can retrieve data using different approaches:**

* **JPQL (Java Persistence Query Language): Object-oriented queries written in a SQL-like syntax.**
* **Native Queries: Raw SQL queries executed directly on the database.**
* **Criteria API: A programmatic way to build queries dynamically.**

**🔹 1. JPQL (Query API)**

**JPQL is similar to SQL but operates on entities and attributes instead of tables and columns.**

**Advantages:**

* **Simple to read and write.**
* **Type-safe compared to native queries.**
* **Database-independent (since it's translated into SQL by JPA).**

**Disadvantages:**

* **Hard to build dynamically.**
* **Not as flexible as Criteria API for complex queries.**

**2. Native Queries (Query API)**

**Native queries execute raw SQL directly on the database.**

**Advantages:**

* **Full SQL power (joins, window functions, stored procedures).**
* **Can use database-specific functions.**
* **Best performance for complex queries.**

**Disadvantages:**

* **Database-dependent (may not be portable).**
* **Not type-safe.**
* **More risk of SQL injection (if parameters are not handled properly).**

**Criteria API (Dynamic Queries)**

**The Criteria API provides a type-safe, programmatic way to build queries dynamically.**

**Advantages:**

* **Type-safe (prevents runtime errors).**
* **Dynamic query building (great for filters & search criteria).**
* **Database-independent like JPQL.**

**Disadvantages:**

* **Complex and verbose syntax.**
* **Harder to read than JPQL.**
* **More boilerplate code.**

| **Feature** | **JPQL (Query API)** | **Native Query** | **Criteria API** |
| --- | --- | --- | --- |
| **Type Safety** | ❌ No | ❌ No | ✅ Yes |
| **Performance** | ✅ Good | ✅ Best | ⚠️ Slightly Slower |
| **Portability** | ✅ Yes | ❌ No | ✅ Yes |
| **Flexibility** | ❌ Low | ✅ High | ✅ High |
| **Best For** | Simple Queries | Complex SQL | Dynamic Queries |

| **Use Case** | **Recommended Approach** |
| --- | --- |
| Simple queries | **JPQL** |
| Database-specific queries | **Native SQL** |
| Performance-critical complex queries | **Native SQL** |
| Dynamic query generation | **Criteria API** |
| Type-safe queries | **Criteria API** |

* **JPQL**: Best for **basic object-oriented queries** that are easy to read.
* **Native Queries**: Best for **performance-heavy, database-specific queries**.
* **Criteria API**: Best for **dynamic, type-safe queries**.