**Difference Between JPA, Hibernate, and Spring Data JPA**

This hands-on exercise demonstrates how Spring Data JPA simplifies database operations compared to traditional Hibernate usage. **JPA (Java Persistence API)** is a specification (interface) that defines how Java objects map to database tables. It doesn’t provide any implementation.

* **Hibernate** is the most commonly used implementation of JPA. It handles the actual SQL generation and object-relational mapping.
* **Spring Data JPA** is a Spring-based abstraction built on top of JPA (and typically uses Hibernate underneath). It reduces boilerplate by providing ready-made repository interfaces.

With Spring Data JPA, we don’t need to write all the session or transaction handling like we do in Hibernate. We can just use simple methods like save() or findAll() and it takes care of everything behind the scenes. Objectives Understand the difference between JPA, Hibernate, and Spring Data JPA.

See how boilerplate code is reduced using Spring Data JPA.

Use Spring Boot, Hibernate, and Spring Data JPA to fetch country data from the database.

Steps Performed

Imported the existing orm-learn-difference Spring Boot project.

Verified database configuration in application.properties.

Ensured Country entity is mapped correctly.

Injected CountryService and used a JPA repository. Compared traditional Hibernate approach with simplified Spring Data JPA usage.

Conclusion Spring Data JPA significantly reduces the complexity of data access layers, making it easier and cleaner to work with databases.