**Hands-on 4: Difference between JPA, Hibernate, and Spring Data JPA**

**1. JPA (Java Persistence API)**

**Type:** Specification / Interface

**What it is:**

JPA is a standard defined by Java EE (Jakarta EE now) for object-relational mapping (ORM). It provides a common interface for ORM tools.

**Who provides it:**

Developed by Oracle as part of the Java EE specification.

Functionality: It defines:

Entity annotations like @Entity, @Table, @Id, etc.

Query language (JPQL)

Entity lifecycle management

Caching and transaction handling

**> Key Point:** JPA is just a set of interfaces; it doesn’t provide an actual implementation.

**2. Hibernate**

**Type:** JPA Implementation + Extra Features

**What it is:**

Hibernate is a popular ORM framework for Java. It implements JPA and also offers additional features not defined in JPA.

**Who provides it:** Developed by Red Hat.

**Functionality:**

Implements all JPA interfaces

**Adds features like:**

Lazy loading, caching, batch fetching

Native SQL queries

Automatic schema generation

Envers for auditing

**> Key Point:** Hibernate can work with or without JPA. When you use @Entity, @Id, etc., Hibernate interprets them and manages your data persistence.

**3. Spring Data JPA**

**Type:** Framework built on JPA + Hibernate (or other JPA providers)

**What it is:**

Spring Data JPA is part of the Spring Data project. It simplifies JPA-based data access by reducing boilerplate code.

**Who provides it:** Spring (Pivotal/VMware)

**Functionality:**

Builds on JPA and uses Hibernate as the default provider

Auto-generates repository implementations using interfaces

Uses method name conventions like findByName() or findByEmailAndStatus()

Adds powerful query support with @Query, Specification, pagination, and sorting

**> Key Point:** Spring Data JPA is not a replacement for JPA or Hibernate, but a higher-level abstraction that makes them easier to use.

**Example in Practice**

**You write:**

@Entity

public class User {

@Id

private Long id;

private String name;

}

JPA defines this structure,

Hibernate manages persistence to DB,

**Spring Data JPA lets you write:**

public interface UserRepository extends JpaRepository<User, Long> {

List<User> findByName(String name);

}