**2.Difference between JPA, Hibernate and Spring Data JPA**

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

* A *specification* (interface) defined by Java EE (now Jakarta EE).  
  Think of it as a *set of guidelines* or *rules* for how Java objects should be mapped to a relational database.
* JPA **does not provide implementation** itself. It only defines APIs and annotations like:

@Entity @Table @Id @GeneratedValue

* **Example Implementations:**
  + Hibernate
  + EclipseLink
  + OpenJPA

**Analogy:**  
JPA is like a *contract or interface* that says “here’s how persistence should work.”

**2. Hibernate**

* A *framework* (library) and one of the most popular implementations of JPA.
  + Provides the actual working code that fulfills JPA’s contracts.
  + Also offers *extra features beyond JPA* (e.g., better caching, performance tuning, native APIs.  
    You can use Hibernate:
  + **As a JPA implementation** (using only the standard JPA APIs).
  + Or **natively** (using Hibernate’s specific APIs like Session, Criteria, etc.).
* **Example:**

Session session = sessionFactory.openSession(); // Hibernate native API

**3. Spring Data JPA**  
 A *Spring project* that sits on top of JPA (and an implementation like Hibernate).

To **simplify** the development of repository (DAO) layers by:

* + Automatically generating query methods.
  + Reducing boilerplate code.
  + Integrating easily with Spring Boot.
  + CrudRepository, JpaRepository, PagingAndSortingRepository.
  + Derived queries by method names.
  + Custom queries with @Query.
* **Example Repository:**

public interface UserRepository extends JpaRepository<User, Long> { List<User> findByLastName(String lastName); }

**Table**

| **Aspect** | **JPA** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| Type | Specification | Implementation / Framework | Spring abstraction over JPA |
| Provides | Interfaces, annotations | Concrete code + extra features | Auto-generated Repositories, Queries |
| Dependency | No runtime code | Hibernate library | Spring Data JPA + Hibernate |
| Usage | EntityManager | Session (native) or JPA | JpaRepository, Spring repositories |

**Quick takeaway:**

* **JPA** = *standard* interface
* **Hibernate** = *implementation* of JPA + more
* **Spring Data JPA** = *Spring helper* that makes JPA/Hibernate easier to use

Top of Form

Bottom of Form