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

| **Feature/Aspect** | **JPA (Java Persistence API)** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| Type | Specification (part of Java EE) | Implementation of JPA | Spring abstraction over JPA (uses Hibernate by default) |
| Provided By | Oracle / Java EE | Red Hat | Spring Framework |
| Boilerplate Code | Requires writing queries, entity managers | Still verbose, manual session handling | Removes most boilerplate using repositories |
| Configuration | Manual (persistence.xml, entityManager) | Manual (session factory) | Auto-configured (via Spring Boot) |
| Out-of-the-box CRUD | No | No | Yes (save(), findById(), etc.) |
| Integration with Spring | Needs setup | Needs setup | Seamless (Spring Boot ready) |
| Learning Curve | Moderate | High | Low (easy for beginners) |
| Popular Methods | persist(), merge(), remove() | save(), load(), get() | findById(), save(), deleteById() |
| Querying Style | JPQL | JPQL + HQL + Criteria API | JPQL + Derived Queries + @Query annotation |

**JPA (Standard)**

EntityManager em = emf.createEntityManager();

em.persist(entity);

**Hibernate (Implementation)**

Session session = sessionFactory.openSession();

session.save(entity);

**Spring Data JPA (Abstraction)**

userRepository.save(entity);