|  |  |  |  |
| --- | --- | --- | --- |
| Feature | JPA (Java Persistence API) | Hibernate | Spring Data JPA |
| Type | Specification (Interface/Standard) | Implementation (ORM Framework) | Abstraction Layer over JPA (Part of Spring ecosystem) |
| Provided By | Oracle (Java EE standard) | Red Hat | Pivotal/VMware (Spring Framework) |
| Purpose | Defines standard for ORM in Java | Implements JPA and adds extra features | Simplifies JPA usage with repository abstraction |
| Is It an Implementation? | No | Yes | No (uses a JPA implementation like Hibernate under the hood) |
| Main Use | ORM specification used by frameworks | Actual ORM library used to map Java objects to DB tables | Reduce boilerplate code and enable quick CRUD with repositories |
| Boilerplate Code | Requires significant code (e.g. writing queries, DAOs) | Some boilerplate needed | Minimal — auto-generates repository methods |
| Query Support | JPQL (Java Persistence Query Language) | JPQL + Native SQL + Criteria API + Hibernate-specific HQL | JPQL + Derived Queries + @Query + Specifications |
| Annotations Provided | @Entity, @Id, @Table, etc. (standard JPA annotations) | Inherits JPA annotations + some Hibernate-specific | Uses JPA annotations; no new ones added |
| Transaction Management | Not included | Provided (or via JTA) | Integrated with Spring’s transaction management |
| Dependency Needed | javax.persistence | Hibernate core libraries | Spring Data JPA + JPA provider (e.g., Hibernate) |
| Custom Extensions | None | Yes (e.g., caching, batch processing, custom types) | Yes (e.g., JpaRepository, CrudRepository, pagination support) |