|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | **JPA** | **Hibernate** | **Spring Data JPA** |
| **1. Type** | Specification/  standard | Implementation of JPA | Spring module built on JPA and Hibernate |
| **2. Purpose** | Defines rules for ORM in Java | Provides ORM features and follows JPA rules | Simplifies database access using repositories |
| **3. Ownership** | Java EE / Jakarta EE | Red Hat | VMware (Spring Framework) |
| **4. Level** | High-level contract | Medium-level implementation | Higher-level abstraction over JPA and Hibernate |
| **5. Vendor Neutrality** | Vendor-neutral (works with many providers) | Vendor-specific (one implementation) | Can use any JPA provider (default is Hibernate) |
| **6.Configuration** | Manual setup with EntityManager | Manual setup with SessionFactory | Auto-configured in Spring Boot (minimal setup) |
| **7. Features** | Defines basic annotations and API | Adds advanced features (caching, batch, etc.) | Adds auto-generated queries, custom repositories |
| **8. Boilerplate Code** | More boilerplate (manual queries) | Less boilerplate, but still manual mapping | Minimal boilerplate (method names generate queries) |
| **9. Learning Curve** | Medium | Medium to high | Easiest for beginners |
| **10. Example Use** | @Entity, @Id, EntityManager | Session, advanced caching, native SQL | JpaRepository, custom finder methods |