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

**JPA (Java Persistence API) – Specification**

* Developed by Oracle as part of **Java EE (now Jakarta EE)**.
* Defines **annotations** and interfaces like @Entity, @Id, EntityManager, etc.
* **Portable**: You can switch between implementations (e.g., from Hibernate to EclipseLink) with minimal code change.
* **Needs an implementation** to function — like Hibernate, EclipseLink, or OpenJPA.
* Encourages **standardized ORM practices**.

**Hibernate – Implementation**

* Most **popular JPA provider** and also works as a standalone ORM tool.
* Adds **additional non-JPA features** like:
  + @Filter, @BatchSize, @Fetch, and more.
  + **Second-level caching** support using EhCache, Infinispan, etc.
  + **Lazy/eager loading** strategies.
* Provides its **own Query Language** called **HQL (Hibernate Query Language)**.
* Can be used without JPA — Hibernate existed before JPA.

**Spring Data JPA – Abstraction Over JPA**

* Part of **Spring Data** project (built on Spring Boot & Spring Framework).
* Eliminates the need to write repository/DAO boilerplate code.
* Supports:
  + **Query Methods** (e.g., findByAuthor(String author))
  + **@Query annotation** for custom queries (JPQL/native SQL)
  + **Pagination and Sorting**
  + **Auditing** support (created/modified date auto-handling)
* Automatically implements **CRUD operations** behind the scenes.

**Comparative Feature Matrix**

| **Feature** | **JPA** | **Hibernate** | **Spring Data JPA** |
| --- | --- | --- | --- |
| Type | Specification (JSR 338) | Implementation | Framework/Abstraction layer |
| Boilerplate Reduction | No | Partial | Yes |
| Used Without Spring | Yes | Yes | No (Spring Required) |
| CRUD Repositories (e.g., save) | No | Manual | Auto-generated |
| Query by Method Name | No | No | Yes |
| Custom Queries with Annotation | No | (HQL) | (@Query with JPQL/Native) |
| Vendor-Specific Extensions | N/A | Yes | Relies on JPA provider |
| Switching JPA Implementations | Easy | Tied to Hibernate | If using standard JPA usage |
| Performance Optimization | Minimal | Yes (Caching, Fetching) | Via Hibernate under the hood |