**DIFFERENCE BETWEEN JPA , HIBERNATE AND SPRING DATA JPA**

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

**What it is:**  
JPA is a standard specification provided by Oracle as part of Java EE (now Jakarta EE). It defines how Java objects are mapped to relational database tables using Object-Relational Mapping (ORM).

**Key Features:**

* JPA is just a set of rules (interfaces and annotations).
* It provides annotations like @Entity, @Id, @GeneratedValue.
* Includes the EntityManager interface for interacting with the database.
* JPA **does not contain any implementation**; it relies on providers like Hibernate.

**Example:**

@Entity

public class Book {

@Id

@GeneratedValue

private Long id;

private String title;

}

**Summary:**  
JPA provides the **blueprint** but not the working engine.

**2. Hibernate – Implementation**

**What it is:**  
Hibernate is an open-source Java framework and the most popular implementation of the JPA specification.

**Key Features:**

* Implements all JPA rules and APIs.
* Adds its own features such as:
  + Caching
  + Lazy loading
  + Dirty checking
  + Hibernate Query Language (HQL)
* Can be used with JPA (EntityManager) or with its native API (Session).

**Example:**

Session session = sessionFactory.openSession();

session.save(book);

**Summary:**  
Hibernate is the **real engine** that runs the rules defined by JPA and adds extra power.

**3. Spring Data JPA – Abstraction**

**What it is:**  
Spring Data JPA is part of the Spring Framework. It builds on JPA and provides an abstraction layer to reduce boilerplate code.

**Key Features:**

* Automatically implements JPA repository interfaces.
* Provides ready-to-use methods like save(), findAll(), deleteById() without writing queries.
* Integrates smoothly with Spring Boot.
* Still uses JPA underneath, and typically uses Hibernate as the default JPA provider.

**Example:**

public interface BookRepository extends JpaRepository<Book, Long> {}

**Summary:**  
Spring Data JPA is a **helper tool** that makes JPA easier and faster to use with Spring applications.

**Comparison Table**

| **Feature** | **JPA (Specification)** | **Hibernate (Implementation)** | **Spring Data JPA (Abstraction)** |
| --- | --- | --- | --- |
| Type | Interface/API | Framework/Library | Spring Module |
| Code Implementation | No | Yes | Yes (auto-generated) |
| Boilerplate Reduction | No | Partial | Yes (major reduction) |
| Query Writing | Manual | HQL or manual | Mostly auto / derived |
| Integration | With any Java app | Standalone or with Spring | Only in Spring ecosystem |
| Example API | EntityManager | SessionFactory | JpaRepository |