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

**JPA (Java Persistence API):**

* **JPA is a specification**, not an implementation.
* It defines a set of **interfaces and annotations** for object-relational mapping (ORM) and database operations in Java.
* JPA allows developers to **map Java classes to database tables**, manage relationships, and perform CRUD operations using an entity manager.
* Since JPA is just a specification, it **needs a provider** (like Hibernate) to work.

**Hibernate:**

* **Hibernate is an actual implementation** of the JPA specification (and existed before JPA itself).
* It provides **extra features** on top of JPA, like better caching, lazy loading, and custom SQL support.
* When using Hibernate, you can either use it as a **JPA provider** or directly through its **native APIs**.
* It handles the interaction between your Java objects and the database efficiently.

**Spring Data JPA:**

* **Spring Data JPA is a framework built on top of JPA and Hibernate.**
* It simplifies the data access layer in Spring applications by **removing boilerplate code**.
* With Spring Data JPA, you can create repositories using interfaces and **automatically get implementations** for common operations like save(), findById(), delete() etc.
* It integrates seamlessly with Spring Boot and handles much of the configuration internally.