JPA, Hibernate, and Spring Data JPA are all related to working with databases in Java, but they serve different purposes. JPA (Java Persistence API) is just a specification—it defines how Java objects should be mapped to database tables, but it doesn’t provide any actual implementation. That’s where Hibernate comes in. Hibernate is the most popular implementation of JPA, and it also adds a lot of useful features beyond the basic JPA spec, like caching, lazy loading, and its own query language called HQL. On top of that, Spring Data JPA makes working with JPA and Hibernate even easier by reducing boilerplate code. With Spring Data JPA, you don’t need to write most of the common queries or implementations yourself—it can generate them just from method names, and it integrates smoothly with Spring Boot. In simple terms, JPA is the guideline, Hibernate is the tool that follows that guideline and does the heavy lifting, and Spring Data JPA is the convenience layer that makes the developer’s life easier.