# Difference Between JPA, Hibernate & Spring Data JPA

When working with databases in Java applications, we often hear about JPA, Hibernate, and Spring Data JPA. While they are closely related, each one plays a different role. Here's a simple explanation to understand them better:

## 1. JPA (Java Persistence API)

- JPA is a specification, not a tool or library.  
- It defines a standard way to map Java objects (like classes) to database tables — this concept is called ORM (Object Relational Mapping).  
- JPA gives us a set of interfaces and annotations (like @Entity, @Id, @OneToMany) that describe how to interact with databases in an object-oriented way.  
- But JPA itself doesn’t do anything unless you use an actual implementation like Hibernate.

## 2. Hibernate

- Hibernate is a real working tool — it is the most popular implementation of JPA.  
- You can use Hibernate with or without JPA.  
- It provides additional features that JPA doesn’t have, such as:  
 - Lazy loading, caching, custom query language (HQL), and more.  
- If JPA defines the rules, Hibernate is the one following and adding extra tools to make development easier.

## 3. Spring Data JPA

- Spring Data JPA is built on top of JPA and Hibernate.  
- It is part of the Spring Framework and is used to simplify database operations.  
- With Spring Data JPA, we can:  
 - Create a repository interface like UserRepository extends JpaRepository  
 - Automatically get methods like save(), findAll(), deleteById(), etc.  
 - Write custom queries using method names (like findByNameAndAge()), or use @Query for JPQL.

## Summary Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature | JPA | Hibernate | Spring Data JPA | Works Alone? |
| Type | Specification (Rulebook) | Implementation (ORM Framework) | Abstraction Layer (Simplifier) | No |
| Provider | Java (Jakarta EE) | Red Hat | Spring Framework | No |
| What it does | Defines ORM rules | Implements JPA + adds features | Simplifies DB access in Spring | No |
| Can it work alone? | No | Yes | No | Varies |
| Example Use | @Entity, EntityManager | Session, HQL, @Cache | JpaRepository, @Query, findBy... |  |

Conclusion

- JPA tells us what we can do with Java and databases.  
- Hibernate shows us how we can do it and adds extra features.  
- Spring Data JPA makes our life easier by doing most of the work automatically.  
  
Together, they form a powerful trio for building robust and clean Java applications that interact with databases.