Difference between JPA, Hibernate and Spring Data JPA

# 1. Java Persistence API (JPA)

Definition: JPA is a specification (JSR 338) for mapping Java objects to relational database tables.  
Key Point: JPA does not provide an implementation. It defines how to persist, read, and manage data but not how to do it technically.  
Usage: You manually use EntityManager to interact with the database.

Example Code (JPA):

EntityManager em = entityManagerFactory.createEntityManager();  
em.getTransaction().begin();  
Country c = em.find(Country.class, "IN");  
em.getTransaction().commit();  
em.close();

# 2. Hibernate

Definition: Hibernate is an ORM (Object-Relational Mapping) framework that implements the JPA specification.  
Key Point: Adds its own features like caching, lazy loading, and provides more control.  
Usage: Uses Session and Transaction, and you handle sessions manually.

Example Code (Hibernate):

Session session = sessionFactory.openSession();  
Transaction tx = session.beginTransaction();  
Country c = session.get(Country.class, "IN");  
tx.commit();  
session.close();

# 3. Spring Data JPA

Definition: A Spring module built on top of JPA and Hibernate that simplifies database access.  
Key Point: Removes boilerplate code. Uses interfaces and method names to generate queries.  
Usage: Extend JpaRepository and let Spring handle everything.

Example Code (Spring Data JPA):

@Autowired  
private CountryRepository countryRepository;  
  
public void getCountry() {  
 Country c = countryRepository.findById("IN").get();  
}

# Conclusion

JPA is a standard interface that defines how ORM should work but doesn’t do anything by itself. Hibernate is a concrete implementation of JPA that provides powerful features for ORM operations, but it requires more manual configuration and code. Spring Data JPA sits on top of both and makes data access easier by reducing boilerplate code, automatically generating queries based on method names, and managing all the underlying complexity. If you're working on a modern Spring Boot project and want to build quickly with minimal configuration, Spring Data JPA is the best and most developer-friendly choice.