1. JPA (Java Persistence API)

* **What it is**: A **specification** for object-relational mapping (ORM) in Java. It defines a set of interfaces and annotations for managing relational data in Java applications.
* **Key Point**: JPA is not an implementation; it’s just a standard. You need a provider (like Hibernate) to use it.
* **Example**: @Entity, @Table, @Id, @Query are JPA annotations.

2. Hibernate

* **What it is**: A **framework** and the most popular implementation of the JPA specification. It provides additional features beyond JPA, such as caching, lazy loading, and custom query capabilities.
* **Key Point**: Hibernate can work with or without JPA. It has its own native API (e.g., Session, Criteria) but also supports JPA’s API.
* **Example**: Hibernate-specific features include @Fetch, @BatchSize, and SessionFactory.

3. Spring Data JPA

* **What it is**: A **Spring-based abstraction** over JPA. It simplifies database access by reducing boilerplate code and providing repository interfaces.
* **Key Point**: It uses JPA (and its provider, like Hibernate) under the hood but adds features like derived query methods, custom queries, and pagination.
* **Example**: JpaRepository and CrudRepository interfaces allow you to perform CRUD operations without writing SQL or JPQL.