**HIBERNATE:**  
**What it is:**

Hibernate is a widely used implementation of the Java Persistence API (JPA). While JPA defines the standard for object-relational mapping (ORM) in Java, Hibernate provides the actual working code that fulfills this specification

**What it does:**

Provides the actual code to persist, query, and manage entities. It can be used with or without JPA.

**Extra features:**

Adds features beyond JPA, like:

Better caching

Native APIs (Session, Criteria)

Extra query options (HQL, native SQL)

**How it’s used:** When you use JPA in Spring Boot, Hibernate is often the default JPA provider under the hood.

Source: https://www.baeldung.com/hibernate-second-level-cache

**JPA:**

**What it is:**

JPA (Java Persistence API) is a specification that’s part of Java EE (now Jakarta EE). It defines a standard approach for mapping Java objects (entities) to relational database tables. This is known as Object-Relational Mapping (ORM).

**What it provides:**

JPA offers a set of interfaces and annotations—such as @Entity, @Id, EntityManager, and @PersistenceContext—that developers use to define and manage how Java objects are persisted to the database.

**What it does NOT do:**

JPA itself doesn’t contain any implementation. It’s just a set of rules and contracts. To actually persist data, you need a provider that implements these rules.

**Who implements it:**

Popular implementations include Hibernate, EclipseLink, and OpenJPA. These frameworks provide the real logic behind the scenes and plug into JPA to make it work.

Source: https://www.geeksforgeeks.org/advance-java/jpa-object-relational-mapping/

**SPRING DATA JPA:**

**What it is:**

Spring Data JPA is a Spring project that builds on top of JPA to simplify database access in Java applications.

**What it does:**

It provides helpful abstractions that reduce boilerplate code:

Automatically generates repository implementations at runtime

Introduces JpaRepository, CrudRepository, and other base interfaces

Supports query methods using naming conventions (e.g., findByEmail, findByAuthorAndTitle)

**How it works:**

Spring Data JPA acts as a layer over JPA and delegates the actual persistence logic to a JPA provider like Hibernate. You define the structure, and Spring wires up the implementation behind the scenes.

Source:https://spring.io/projects/spring-data-jpa