## 1. Java Persistence API (JPA)

# What it is: JPA is a specification in Java for object-relational mapping (ORM). It provides a standard set of interfaces and annotations to map Java objects to database tables.

# Key Point: JPA is just a spec, not an implementation. It defines what should be done, but not how to do it.

# Example: Interfaces like EntityManager, annotations like @Entity, @Id, @OneToMany.

## 2. Hibernate

# What it is: Hibernate is a concrete implementation of the JPA specification. It is also a powerful standalone ORM framework that existed even before JPA was introduced.

# Extra Features: Hibernate provides additional functionalities beyond the JPA spec, like:

# Caching (first-level and second-level)

# Dirty checking

# Advanced query capabilities (HQL, Criteria API)

# Relationship to JPA: Hibernate can be used as a JPA provider or standalone.

## 3. Spring Data JPA

# What it is: Spring Data JPA is a higher-level abstraction built on top of JPA and a JPA provider (like Hibernate). It is part of the Spring Data project.

# Goal: To reduce boilerplate code when working with data persistence in Spring apps.

# Features:

# Automatically generates repository implementations for interfaces

# Methods like findByName(), deleteByEmail() are created from method names

# Supports pagination, sorting, custom queries with JPQL or native SQL

# Dependencies: Spring Data JPA uses Hibernate internally as the default provider.