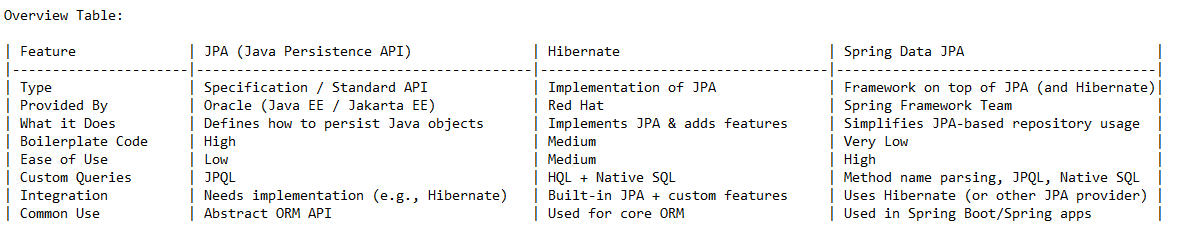
**Difference between JPA, Hibernate and Spring Data JPA**



**Detailed Explanation:**

**1. JPA (Java Persistence API)**

* JPA is a specification.
* It defines how Java objects map to database tables.
* It needs a provider (like Hibernate) to work.
* Defines annotations: @Entity, @Id, etc., and APIs like EntityManager.

**2. Hibernate**

* A popular JPA provider and ORM tool.
* Implements JPA and adds extra features (e.g., HQL, caching).
* Provides both JPA and native APIs (SessionFactory, Session, etc).

**3. Spring Data JPA**

* Built on top of JPA and Hibernate.
* Reduces boilerplate by using Spring Repositories.
* Supports derived queries, @Query annotations, and native queries.
* Integrated with Spring Boot, widely used in modern applications.

**Example without Spring Data JPA (using only JPA + Hibernate):**

EntityManager em = entityManagerFactory.createEntityManager();

TypedQuery<User> query = em.createQuery("SELECT u FROM User u", User.class);

List<User> users = query.getResultList();

**Example With Spring Data JPA:**

public interface UserRepository extends JpaRepository<User, Long> {

List<User> findByEmail(String email);

}