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

**JPA (Java Persistence API)**

* JPA is a Java specification (an interface) that standardizes how Java objects are mapped to relational databases.
* It defines ORM behaviour, but doesn't provide the implementation.
* Includes annotations like @Entity, @Id, @OneToMany, etc.
* Uses JPQL (Java Persistence Query Language) for queries.

**Example:**

@Entity

public class Book {

@Id

private Long id;

private String title;

}

**Hibernate**

* Hibernate is a popular ORM framework that implements the JPA specification.
* It existed before JPA, and later added JPA support.
* Provides additional features beyond JPA:
  + Caching
  + Lazy/eager loading
  + Dirty checking
  + HQL (Hibernate Query Language)

**Example:**

// Using Hibernate-specific annotation

@Cache(usage = CacheConcurrencyStrategy.READ\_ONLY)

public class Book {

//...

}

**Spring Data JPA**

* Spring project that sits on top of JPA and Hibernate.
* Removes boilerplate code by generating implementations for repositories.
* Uses JpaRepository, CrudRepository interfaces.
* Supports query methods like findByTitle(String title) and custom queries with @Query.
* Easily integrates with Spring Boot and auto-configures data source, JPA provider, etc.

**Example:**

public interface BookRepository extends JpaRepository<Book, Long> {

List<Book> findByTitle(String title);

@Query("SELECT b FROM Book b WHERE b.title LIKE %:keyword%")

List<Book> searchByKeyword(@Param("keyword") String keyword);

}