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

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

* **It is a specification**, not a tool.
* Provides a standard way to manage relational data in Java apps.
* Defines annotations like @Entity, @Id, @OneToMany, etc.
* Needs a **provider** (like Hibernate) to work.

Example:

java

CopyEdit

@Entity

public class Student {

@Id

private Long id;

private String name;

}

**2. Hibernate**

* **It is the most popular JPA implementation**.
* Adds extra features on top of JPA.
* Converts Java objects to database tables and vice versa.
* Handles SQL generation, caching, transactions, etc.

Example (Hibernate-specific config):

properties

CopyEdit

hibernate.show\_sql=true

hibernate.format\_sql=true

**3. Spring Data JPA**

* A **Spring framework module**.
* Makes working with JPA (and Hibernate) even easier.
* You only write **interfaces**, no need to write SQL or implementation.

Example:

java

CopyEdit

public interface StudentRepository extends JpaRepository<Student, Long> {

List<Student> findByName(String name);

}

**How They Work Together**

scss

CopyEdit

Spring Data JPA

↓

JPA (Java Persistence API)

↓

Hibernate (Implementation)

↓

Database (MySQL, H2, etc.)