**Difference Between JPA Hibernate and Spring Data and JPA**

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

* JPA is a **specification** (not a tool).
* It provides **rules and guidelines** to map Java classes with database tables.
* Developed by **Oracle (Java EE)**.
* Requires an **implementation** to work (like Hibernate).
* Common annotations: @Entity, @Table, @Id, @Column.

**💡 Example Use:**

java

Copy code

@Entity

@Table(name = "student")

public class Student {

@Id

private int id;

private String name;

}

**🔹 2. Hibernate**

* Hibernate is an **actual framework** that implements the JPA specification.
* Created by **Red Hat**.
* Converts Java objects into database rows and vice versa.
* Offers additional features like:
  + HQL (Hibernate Query Language)
  + Lazy loading
  + First-level cache
* Can work **with or without JPA**.

**💡 Example Use (Native Hibernate):**

java

Copy code

Session session = sessionFactory.openSession();

Student s = session.get(Student.class, 1);

**🔹 3. Spring Data JPA**

* Spring Data JPA is a **Spring module** built on top of JPA and Hibernate.
* Simplifies JPA by removing boilerplate code (like writing repository methods).
* Automatically generates queries like:
  + findAll()
  + findById()
  + deleteByName(), etc.
* Requires Spring Boot, Hibernate, and JPA internally.

**💡 Example Use:**

java

Copy code

public interface StudentRepository extends JpaRepository<Student, Integer> {

List<Student> findByName(String name);

}