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

* **JPA (Java Persistence API)**

JPA is a specification (a set of interfaces) for persisting Java objects in a relational database. It provides guidelines and annotations such as @Entity, @Id, and @Query, but it does not provide any actual implementation. It requires an ORM tool (like Hibernate) to work.

* **Hibernate**

Hibernate is a popular open-source ORM (Object-Relational Mapping) framework that implements the JPA specification and provides additional features beyond JPA. It can be used directly without JPA and offers advanced features like caching, lazy loading, and custom query APIs.

* **Spring Data JPA**

Spring Data JPA is a part of the Spring ecosystem that provides an abstraction layer over JPA. It simplifies the development of data access layers by automatically implementing repository interfaces, thus reducing boilerplate code. It uses JPA and Hibernate internally.

* **How to Use Hibernate in a Java Application**

**Step 1:**

Add Hibernate and Database Dependencies in pom.xml:

<dependencies>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-core</artifactId>

<version>5.6.15.Final</version>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.33</version>

</dependency>

</dependencies>

**Step 2:**

Create Hibernate Configuration File (hibernate.cfg.xml)  
Located in src/main/resources

**Step 3:**

Create Entity Class

**Step 4:**

Use Hibernate in Main Class

* **How to Use Spring Data JPA in a Spring Boot Application**

**Step 1:**

Create Spring Boot Project and Add Dependencies:

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

</dependency>

</dependencies>

**Step 2:**

Configure application.properties

**Step 3:**

Create Entity Class

**Step 4:**

Create Repository Interface

**Step 5:**

Use Repository in Service or Main Class