**Hands on 4**

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

In the world of Java and Spring Boot, we often work with data from databases. To make this easy, we have tools like **JPA**, **Hibernate**, and **Spring Data JPA**. But what exactly are these? How are they different? Let’s break it down in simple language.

**What is JPA?**

* **JPA** stands for **Java Persistence API**.
* It is **not a tool or library**. It is just a **set of rules (interface/specification)** that explains **how Java objects should be saved, updated, deleted, or read from a database**.
* JPA doesn’t do anything on its own. It needs someone to **implement these rules**.

Think of JPA as a **blueprint** for working with databases in Java.

**What is Hibernate?**

* Hibernate is a **library/tool** that **follows the rules of JPA**.
* It is called an **ORM (Object Relational Mapping) tool**.
* It allows you to map Java classes (like Employee) to database tables (like employee).
* Hibernate actually does the work that JPA tells it to do.

Think of Hibernate as the **builder** that follows the **JPA blueprint.**

**What is Spring Data JPA?**

* Spring Data JPA is part of the **Spring ecosystem**.
* It makes working with JPA and Hibernate **even easier**.
* You don’t have to write complex SQL queries or open/close database connections.
* It provides ready-made methods to **save, update, delete, and find data** with just one line of code.

Think of Spring Data JPA as a **smart assistant** that automatically does the work for you.

**Comparison Using Code**

public Integer addEmployee(Employee employee) {

Session session = factory.openSession();

Transaction tx = null;

Integer employeeID = null;

try {

tx = session.beginTransaction(); // Start transaction

employeeID = (Integer) session.save(employee); // Save employee

tx.commit(); // Commit transaction

} catch (HibernateException e) {

if (tx != null) tx.rollback(); // Rollback on error

e.printStackTrace();

} finally {

session.close(); // Close connection

}

return employeeID;

}

**Spring Data JPA Code :**

**public interface EmployeeRepository extends JpaRepository<Employee, Integer> {**

**// No code needed! All methods come from JpaRepository**

**}**