**Hands on 4**

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

**Java Persistence API (JPA)**

* JPA is a specification (JSR 338) for ORM (Object-Relational Mapping).
* It defines annotations, interfaces, and behavior for managing persistence.
* JPA does not have an implementation it just specifies what should be done, not how.

**Hibernate**

* Hibernate is a popular implementation of JPA.
* It is a full ORM framework that translates Java objects into database tables and vice versa.
* Hibernate handles SQL generation, object caching, session management, etc.

**Spring Data JPA**

* Spring Data JPA is not an implementation of JPA.
* It is a Spring module that sits on top of JPA (typically using Hibernate) and reduces boilerplate code.
* It manages repositories, queries, and transactions using simple interfaces like JpaRepository.

**Relationship Diagram**

**Application Code**

**↓**

**Spring Data JPA**

**↓**

**JPA (Interfaces/Specs)**

**↓**

**Hibernate (Implements JPA)**

**↓**

**Database (e.g., MySQL)**

**Hibernate Code (Manual):**

**public Integer addEmployee(Employee employee) {**

**Session session = factory.openSession();**

**Transaction tx = null;**

**Integer employeeID = null;**

**try {**

**tx = session.beginTransaction();**

**employeeID = (Integer) session.save(employee);**

**tx.commit();**

**} catch (HibernateException e) {**

**if (tx != null) tx.rollback();**

**e.printStackTrace();**

**} finally {**

**session.close();**

**}**

**return employeeID;**

**}**

**Spring Data JPA Code (Simplified):**

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

**}**

**@Service**

**public class EmployeeService {**

**@Autowired**

**private EmployeeRepository employeeRepository;**

**@Transactional**

**public void addEmployee(Employee employee) {**

**employeeRepository.save(employee);**

**}**

**}**

**Conclusion**

* JPA is the blueprint (interface/specification) for Java persistence.
* Hibernate is the builder (implementation) that executes JPA's rules.
* Spring Data JPA is the facilitator, which abstracts complexity and simplifies development.
* Spring Data JPA allows developers to focus more on business logic by automating common persistence tasks.