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

**Java Persistence API (JPA)**

**———————————————**

* JSR 338 Specification for persisting, reading and managing data from Java objects
* Does not contain concrete implementation of the specification
* Hibernate is one of the implementation of JPA
* It defines how to map Java objects to database tables using annotations.
* Examples of JPA providers (implementations): Hibernate, EclipseLink, OpenJPA.
* Example Usage (only with JPA): You must use an implementation like Hibernate underneath.

**Hibernate**

**—————**

* Hibernate is a popular ORM (Object Relational Mapping) tool and a JPA implementation.
* It provides all the boilerplate code to manage sessions, transactions, and mapping.
* You manually write code for session handling (open, begin, commit, close, etc.)
* Hibernate provides powerful features like lazy loading, caching, dirty checking, etc.
* My Application <----> Hibernate <----> Database
* (Object)————————> (Object)
* Hibernate handles all low level sql.
* Minimise the amount of JDBC code.
* Example code:

Session session = factory.openSession();

Transaction tx = session.beginTransaction();

session.save(employee);

tx.commit();

session.close();

**Spring Data JPA**

**————————**

* Does not have JPA implementation, but reduces boiler plate code
* This is another level of abstraction over JPA implementation provider like Hibernate
* Manages transaction
* It’s a Spring-based abstraction over JPA that makes working with databases easier.
* It reduces boilerplate code by automatically generating queries using method names.
* You only write repository interfaces, Spring will auto-implement them.
* It internally uses Hibernate (or any JPA provider) but makes your code cleaner and simpler.
* Example Code:

@Repository

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

@Service

public class EmployeeService {

@Autowired

private EmployeeRepository repo;

public void addEmployee(Employee emp) {

repo.save(emp);

}

}