

## Java Persistence API (JPA)

**Definition:** JPA is just a specification. It is like a set of rules and guidelines that explain how Java objects should be stored in a relational database.

- 1) It doesn't actually do anything on its own.
- 2) It defines standard annotations like `@Entity`, `@Id`, and interfaces like `EntityManager`.
- 3) JPA is like the **blueprint** and it needs an **implementation** to actually work.

**Ex:**

`@Entity`

```
public class Employee {
```

```
    @Id
```

```
    private int id;
```

```
    private String name;
```

```
} This class tells JPA how to map a Java object to a database table.
```

## Hibernate

**Definition:** Hibernate is a **popular ORM framework** in Java that provides the **actual implementation** of the JPA specification.

- Implements JPA and also provide additional features.
- Offers advanced feature like **second-level caching**, **lazy fetching**, batch processing.
- JPA tells what to do and Hibernate does the work.

**Ex:**

```
EntityManagerFactory emf = Persistence.createEntityManagerFactory("examplePU");
```

```
EntityManager em = emf.createEntityManager();
```

```
em.getTransaction().begin();
```

```
em.persist(new Employee());
```

```
em.getTransaction().commit();
```

---

## Spring Data JPA

**Definition:** Spring Data JPA is a Spring Framework project that provides an abstraction layer on top of JPA/Hibernate to simplify data access.

- It doesn't implement JPA itself but makes working with JPA and Hibernate much easier.

- It eliminates a lot of boilerplate code.
- With just an interface we can perform database operations like save,delete,find and more without writing a single SQL or JPQL query.

**Ex:**

```
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {  
    List<Employee> findByName(String name);  
};
```

### **Conclusion:**

JPA is just a set of rules that tells us how Java objects should be stored in a database it does not do anything by itself. Hibernate is a tool that follows those rules and actually handles the database operations. Spring Data JPA makes things easier by removing the need to write most of the code, just define a simple interface and it handles all other stuffs.