**Java Persistence API (JPA)**

* **What it is**: A **specification (JSR 338)** provided by Oracle.
* **Purpose**: To define a standard for ORM (Object-Relational Mapping) in Java.
* **Key Point**: **Does not contain implementation**, just interfaces and annotations.
* **Example**: @Entity, @Table, @Id, @Query, etc.
* **Common Implementations**: **Hibernate**, EclipseLink, OpenJPA.

**Hibernate**

* **What it is**: A **concrete ORM framework** that implements the **JPA specification**.
* **Purpose**: To provide an implementation for JPA and also extra features beyond JPA.
* **Can work with or without JPA**.
* **Responsibilities**:
  + Handles session and transaction management.
  + Provides powerful HQL (Hibernate Query Language).
  + Offers caching, lazy/eager loading, etc.

**Hibernate Code Example**

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**

* **What it is**: A **Spring-based framework** that **simplifies JPA usage**.
* **Purpose**: Provides **abstraction over JPA/Hibernate** to reduce boilerplate code.
* **Does not implement JPA itself**, it uses JPA implementations like **Hibernate** behind the scenes.
* **Key Features**:
  + No need to write DAO/Repository code.
  + Automatically handles CRUD, Pagination, Query derivation.
  + Integrates with Spring's transaction management (@Transactional).
  + Allows method names to define queries (e.g., findByName).

**Spring Data JPA Code Example**

// Repository Interface

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

// Service Class

@Autowired

private EmployeeRepository employeeRepository;

@Transactional

public void addEmployee(Employee employee) {

employeeRepository.save(employee);

}