

# Spring 공부

## JDBC, JPA, Hibernate, Mybatis 에 관하여

### 목차

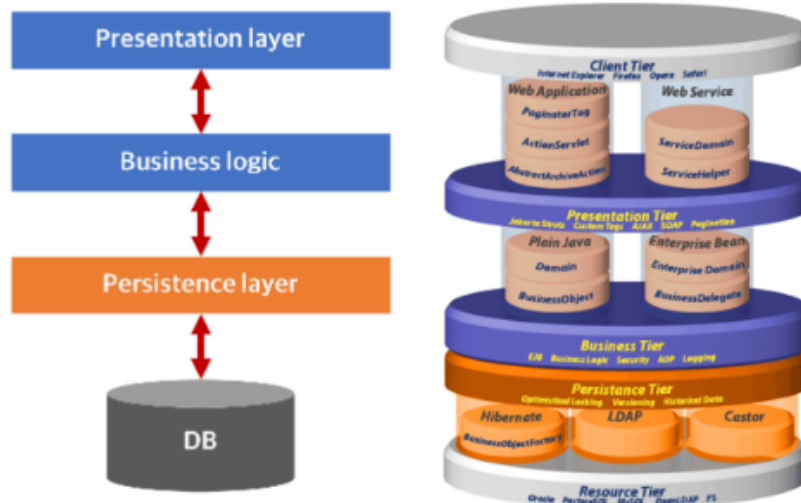
1. 영속성(Persistence) 의 개념
2. SQL Mapper와 ORM의 차이
3. JDBC(Data Transfer Object)란?
4. JPA/Hibernate의 이해
5. Mybatis의 이해

### 1. 영속성(Persistence)

데이터를 생성한 프로그램이 종료되더라도 사라지지 않는 데이터의 특성!

영속성을 갖지 않는 데이터는 메모리에서만 존재하므로 프로그램 종료 시 모두 버려진다. 파일시스템, 데이터베이스 등을 활용하여 프로그램의 데이터에 영속성을 부여한다.

o



o 계층 참고

- o 프레젠테이션 계층 (Presentation layer) - UI 계층 (UI layer) 이라고도 함
- o 애플리케이션 계층 (Application layer) - 서비스 계층 (Service layer) 이라고도 함
- o 비즈니스 논리 계층 (Business logic layer) - 도메인 계층 (Domain layer) 이라고도 함
- o 데이터 접근 계층 (Data access layer) - 영속 계층 (Persistence layer) 이라고도 함

### 2. SQL Mapper와 ORM

Persistence Framework 는 SQL Mapper와 ORM으로 나뉜다.

ORM은 데이터베이스 객체를 자바 객체로 매핑함으로써 객체 간의 관계를 바탕으로 SQL을 자동으로 생성해주지만 SQL Mapper는 SQL을 명시해주어야 한다.

ORM은 관계형 데이터베이스의 "관계"를 object에 반영하자는 것이 목적이다. 반면 SQL Mapper는 단순히 필드를 매핑시키는 것이 목적이다. 지향점의 차이!

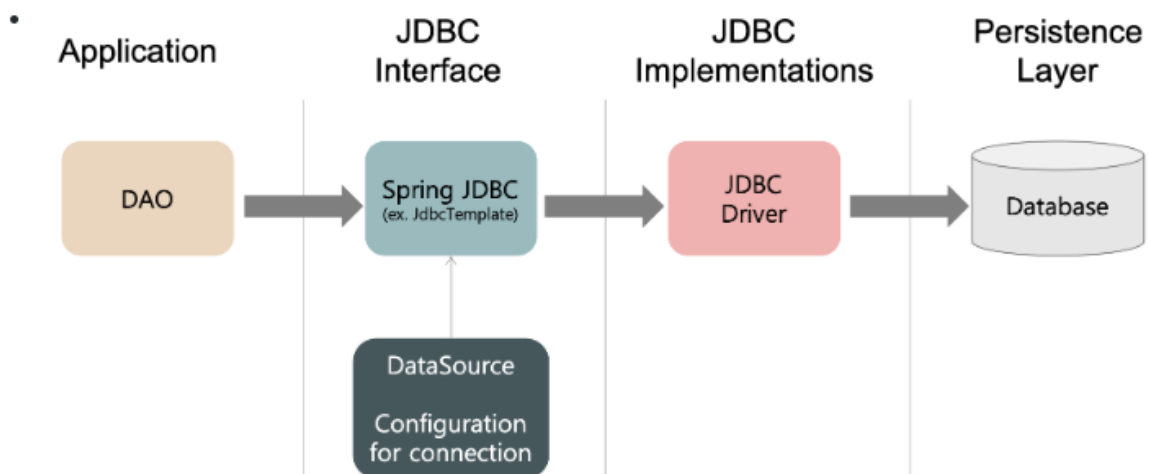
## # SQL Mapper

- SQL <—매핑—> Object 필드
- SQL Mapper는 SQL 문장으로 직접 데이터베이스 데이터를 다룬다.
  - 즉, SQL Mapper는 SQL을 명시해줘야 한다.
  - Ex) Mybatis, JdbcTemplate 등

## # ORM(Object-Relational Mapping), 객체-관계 매핑

- 데이터베이스 데이터 <—매핑—> Object 필드
  - 객체를 통해 간접적으로 데이터베이스 데이터를 다룬다.
- 객체와 관계형 데이터베이스의 데이터를 자동으로 매핑(연결)해주는 것을 말한다.
  - ORM을 이용하면 SQL Query가 아닌 직관적인 코드(메서드)로 데이터를 조작할 수 있다.
  - 객체 간의 관계를 바탕으로 SQL을 자동으로 생성한다.
- Persistent API라고도 할 수 있다.
  - Ex) JPA, Hibernate 등
- ORM의 장단점 참고

## 3. JDBC(Java Database Connectivity)

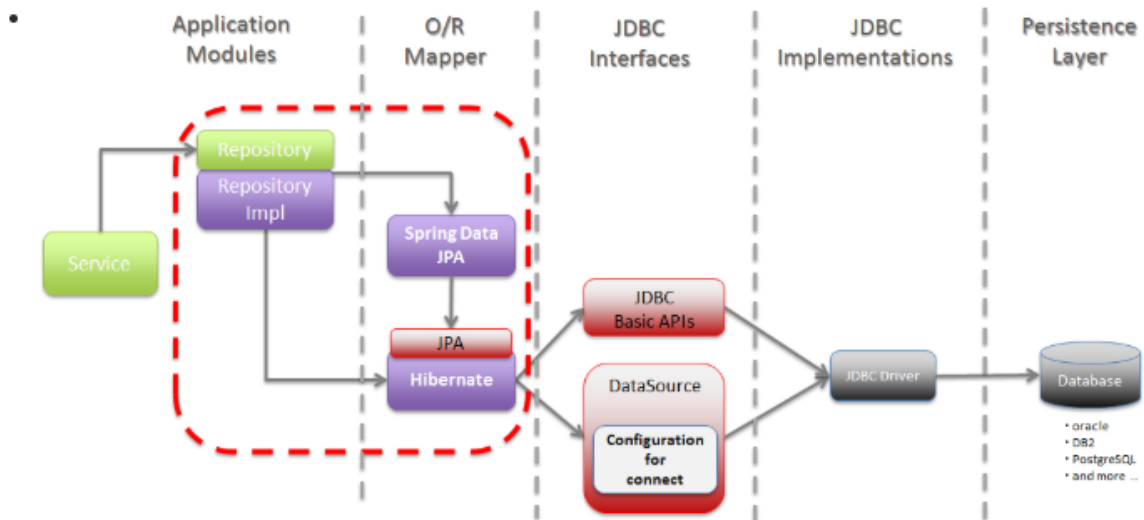


JDBC는 DB에 접근할 수 있도록 Java에서 제공하는 API이다.

모든 Java의 Data Access 기술의 근간

즉! 모든 Persistence Framework는 내부적으로 JDBC API를 이용한다!!!

## 4-1. JPA (Java Persistent API)



자바 ORM 기술에 대한 API 표준 명세로 Java에서 제공하는 API이다.

자바 플랫폼 SE와 자바 플랫폼 EE를 사용하는 응용프로그램에서 관계형 데이터베이스의 관리를 표현하는 자바 API이다.

즉! JPA는 ORM을 사용하기 위한 표준 인터페이스를 모아둔 것이다.

기존에 EJB에서 제공되던 엔티티 빈(Entity Bean)을 대체하는 기술이다.

- 구성요소

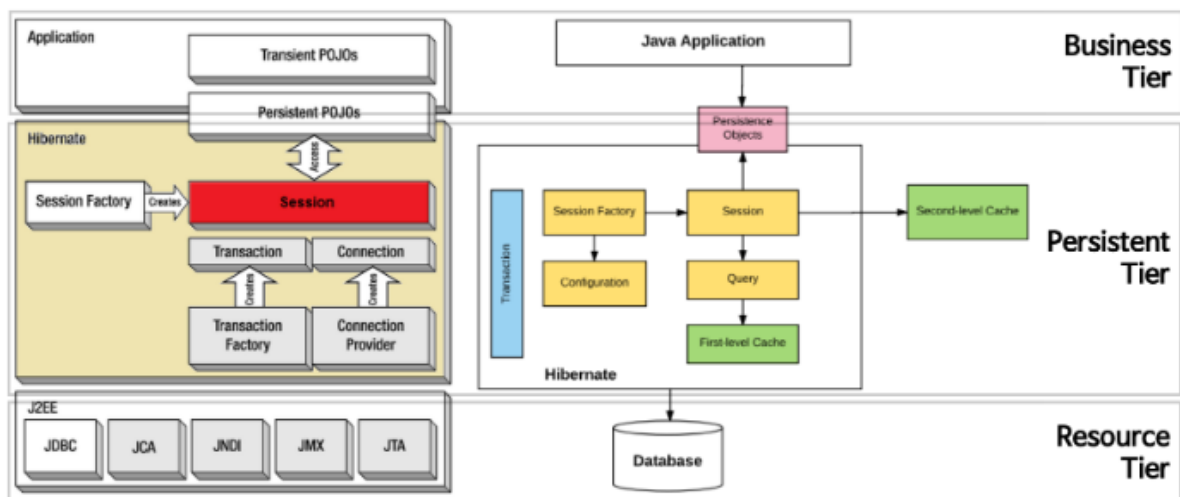
- javax.persistence 패키지로 정의된 API 그 자체
- JPQL(Java Persistence Query Language)
- 객체 / 관계 메타데이터

사용자가 원하는 JPA 구현체를 선택해 사용할 수 있다.

대표적인 구현체!! ==> Hibernate, EclipseLink, DataNucleus, OpenJPA, TopLink Essentials 등

이러한 구현체를 ORM Framework 라고 부른다.

## 4-2. Hibernate



JPA의 구현체 중 하나이다.

SQL을 직접 사용하지 않는다고 하여 JDBC API를 사용하지 않는 것은 아니다.  
메서드 내부에는 JDBC가 동작하고 있다. 다만 직접 작성하지는 않을 뿐이다.

HQL(Hibernate Query Language)라는 강력한 쿼리 언어를 포함한다.

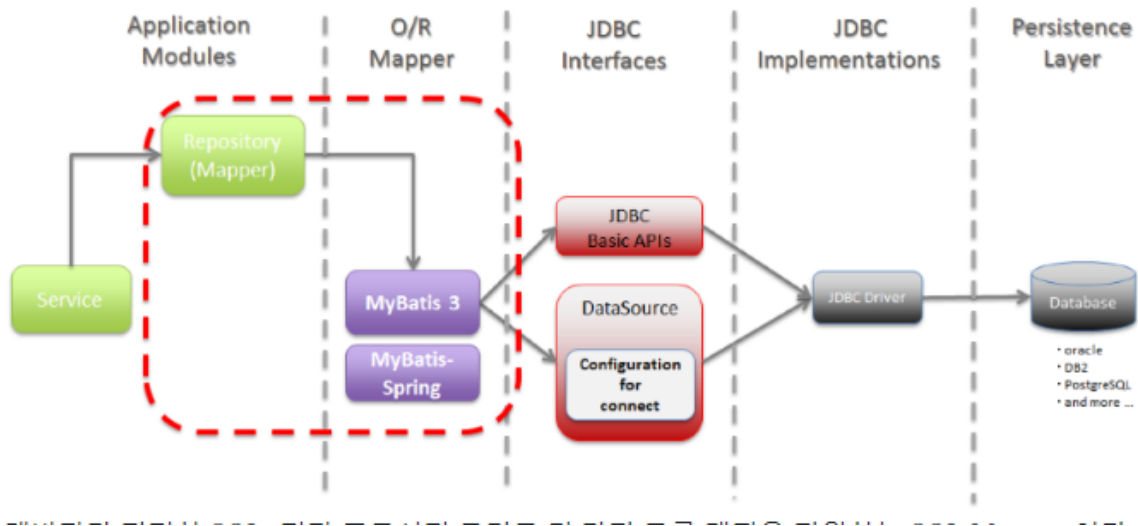
SQL 과 매우 비슷하며 추가적인 컨벤션을 정의할 수 있다.

객체지향적이며 상속,다형성,관계등 객체지향의 강점을 갖는다.

자바 클래스와 프로퍼티의 이름을 제외하고 대소문자를 구분한다.

쿼리결과를 객체로 반환하며 프로그래머에 의해 생성되고 직접 접근 가능하다.

## 5. Mybatis



개발자가 지정한 SQL, 저장 프로시저, 몇가지의 고급매핑을 지원하는 SQL Mapper

JDBC로 처리하는 상당 부분의 코드와 파라미터 설정 및 결과 매핑을 대신한다.

- 기존 JDBC 사용 시 DB와 관련하여 복잡한 설정을 다뤄야했지만 SQL Mapper는 자바 객체를 실제 SQL문에 연결함으로 편리하다.
- 데이터베이스 record에 원시타입과 Map인터페이스 그리고 자바 POJO를 설정하여 매핑하기 위해 xml과 Annotation을 사용할 수 있다.

SQL에 대한 모든 컨트롤을 하고자 할 때 매우 적합하다.

SQL 쿼리들이 최적화가 잘 되어 있을경우 적합하다.