

SPRING BOOT

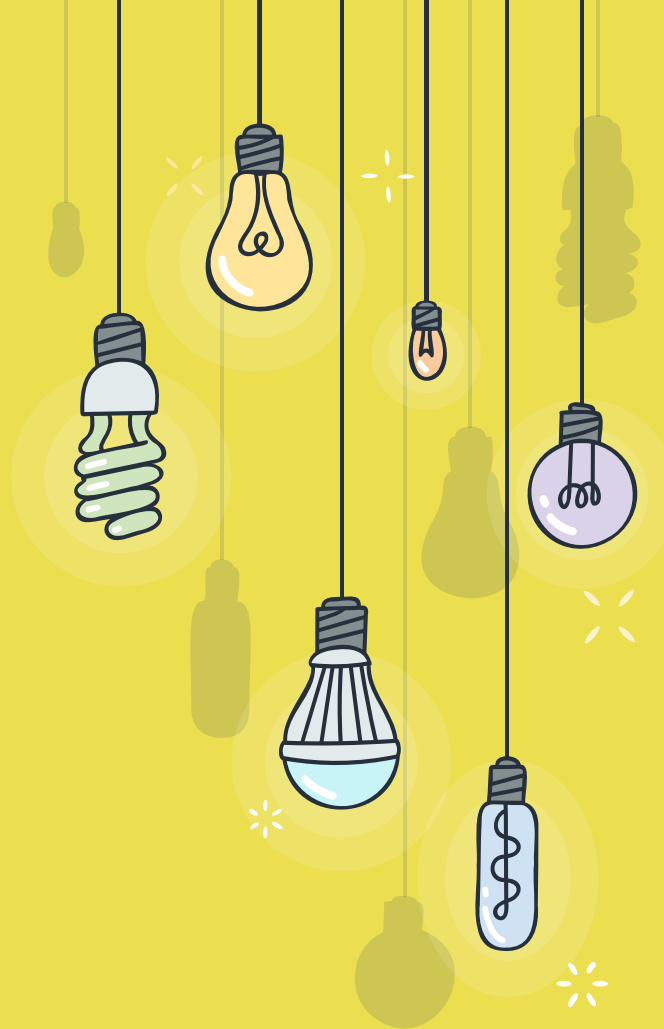
1. JPA

2. 연관매핑



1

JPA 개요



* 데이터베이스 연동 기술

+ 데이터베이스 연동 기술

- × 전통적인 JDBC
- × MyBATIS, Spring JDBC
 - ◆ 데이터 매퍼(Mapper) 기술
- × Hibernate
 - ◆ 대표적인 ORM(Object relational Mapping) 기술

* JPA의 등장

+ JPA(Java Persistence API) 등장

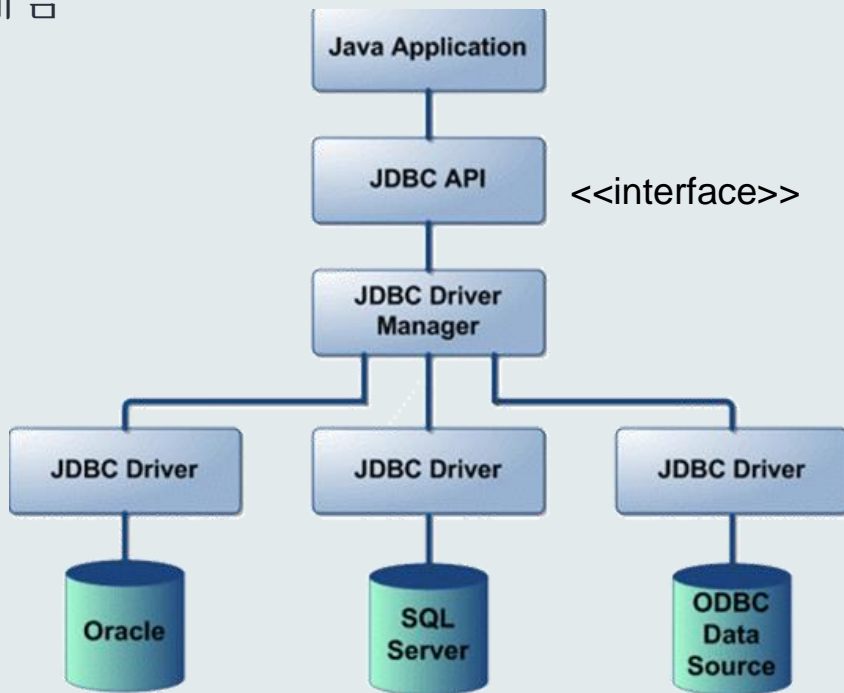
× ORM

- ◆ iBATIS, MyBATIS등의 데이터 매퍼와 달리 데이터베이스 연동에 필요한 java코드, 실질적인 sql구문까지 제공



* JPA개요

- + JPA는 다형성을 기반으로 제공되는 JDBC API와 동일한 개념



[그림1 JDBC구조]

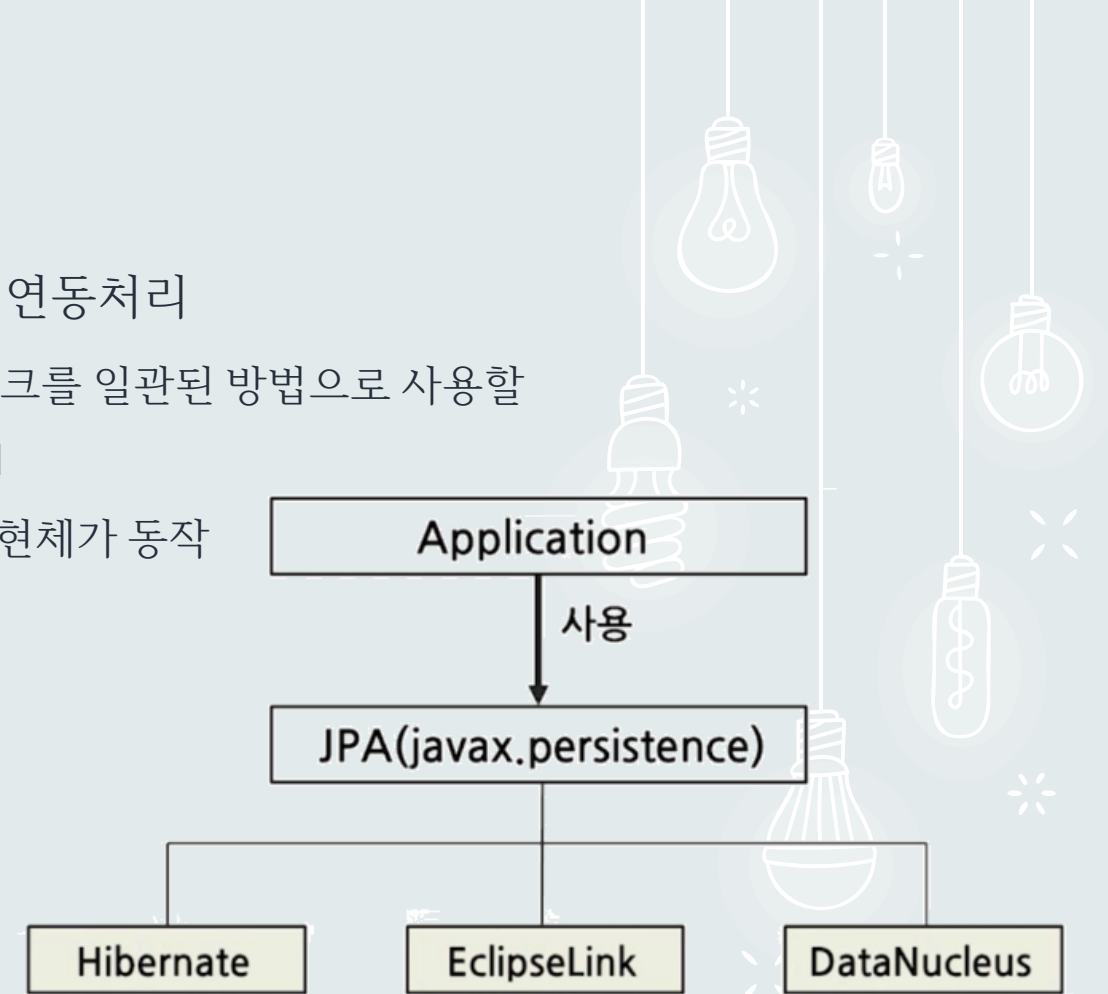
* JPA개요

+ JPA를 이용한 데이터베이스 연동처리

- ✕ JPA는 모든 ORM 프레임워크를 일관된 방법으로 사용할 수 있도록 제공된 확장 API
- ✕ 실제로는 JPA를 구현한 구현체가 동작

+ JPA구현체

- ✕ Hibernate (기본 구현체)
- ✕ EclipseLink
- ✕ DataNucleus 등 다양



* JPA 동작원리

+ JPA와 JDBC

- ✕ JPA는 내부적으로 JDBC API를 이용하여 데이터베이스를 연동 처리

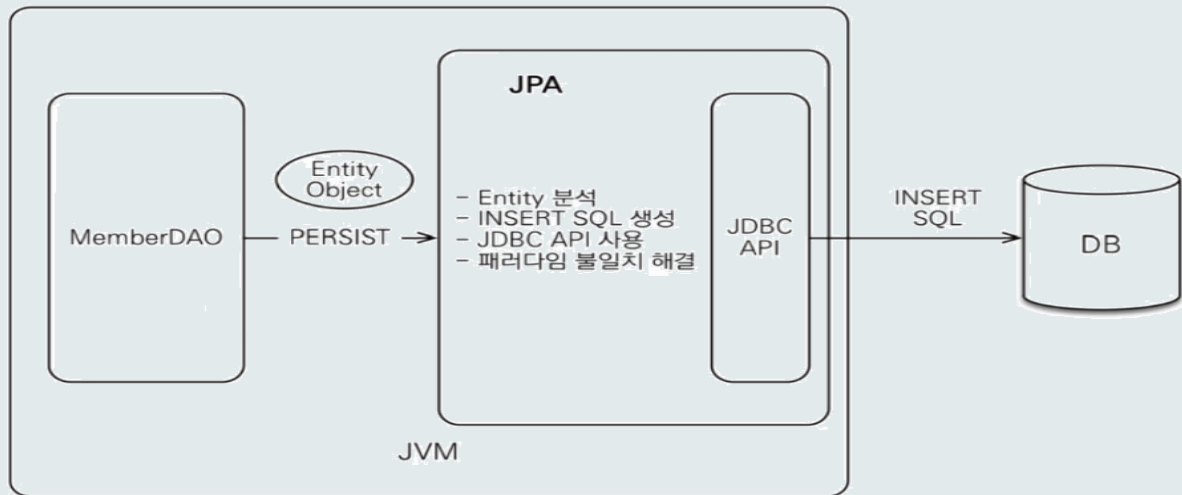


+ JPA와 SQL

- ✕ 애플리케이션에 생성한 JAVA객체(Value Object)를 관계형 DB 테이블과 자동 매핑

JAVA 클래스이름 = 테이블
멤버변수 = 컬럼명

* JPA 동작원리와 장점

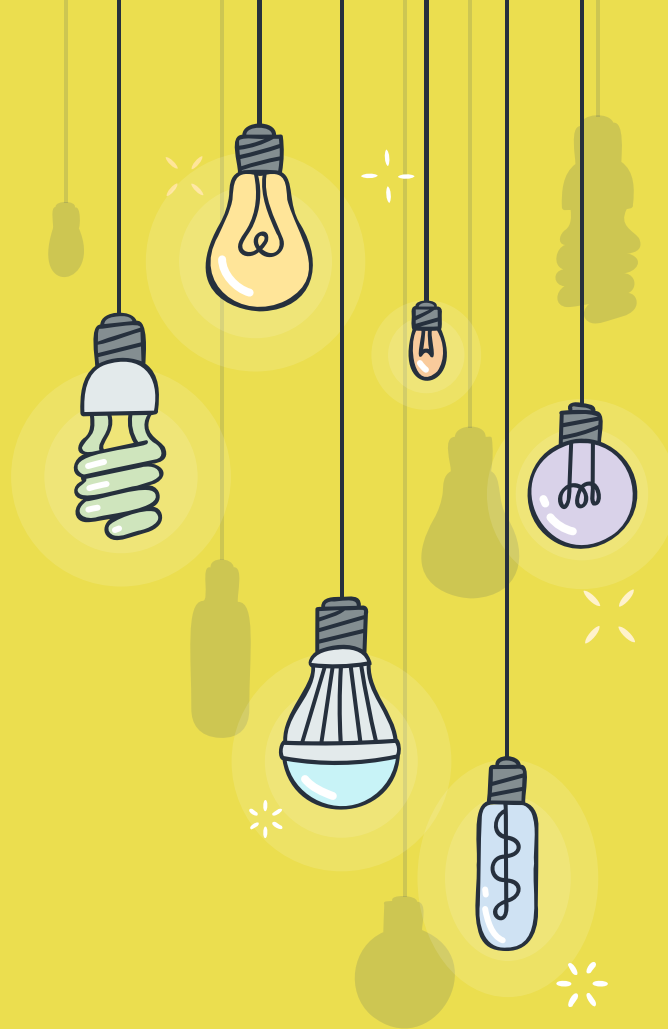


+ 장점

- ✕ 유지보수시 JPA구현체를 쉽게 변경할 수 있음
- ✕ JDBC의 복잡한 절차를 대신 처리

2

연관매핑



* 연관관계 매핑

+ 연관 매핑

- × 테이블의 연관관계와 엔티티의 연관관계를 어노테이션 같은 메타데이터를 통해 매핑하는 것
- × 데이터베이스 테이블은 FK를 기반으로 관계를 표현하지만 객체는 참조 변수를 사용함
 - ◆ 연관 매핑을 설정하여 데이터베이스 조인을 처리
 - ◆ 연관된 엔티티들에 대해 단방향/양방향 매핑을 설정

* 연관매핑 기준

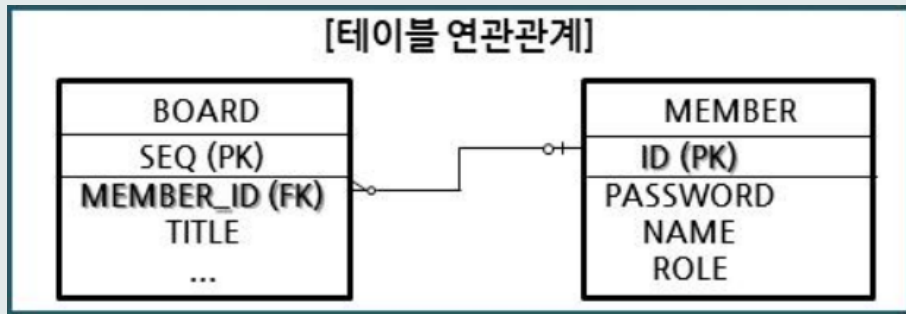
+ 연관 매핑의 종류를 구분하는 기준

용어	설명
방향(Direction)	<ul style="list-style-type: none">• 단방향과 양방향이 있다• 게시판(Board) 객체가 참조 변수를 통해 회원(Member) 객체를 참조하면 단방향• 회원 객체도 게시판 객체를 참조한다면 양방향이 됨• 방향은 객체에만 존재하고 테이블은 항상 양방향임
다중성(Multiplicity)	<ul style="list-style-type: none">• 다대일(N:1), 일대다(1:N), 일대일(1:1), 다대다(N:M)• 회원이 여러 개의 게시글을 소유한다면 회원과 게시판은 일대다(1:N) 관계• 반대로 게시판(Board) 입장에서 보면 게시판과 회원은 다대일(N:1) 관계

* 단방향 연관 매핑

+ 다대일(N:1) 관계

- × 데이터 모델링에서 가장 많이 사용하는관계



- × 참조되는 엔티티 작성
 - ◆ Member는 별다른 설정이 필요 없음

* 단방향 연관 매핑

```
@Data
@Entity(name="BOARDAM")
public class Board {
    @Id
    @GeneratedValue
    private Long bno;
    private String title;
    private String content;

    @ManyToOne
    @JoinColumn(name="WRITER")
    private Member member;
}
```

THANKS!

+ Any questions?

