ORM

자세히 알고 맛깔나게 쓰자 (feat.SpringBoot 

우아한 테크코스 3기

검프🏃

객체 지향 패러다임

시스템을 구성하는 객체들에게 적절한 책임을 할당하는 것

다형성, 추상화, 캡슐화, 상속의 특징을 가진다

객체 지향 패러다임

객체는 자유롭게 객체 그래프를 탐색할 수 있어야 한다

```
class Member {
    Long id;
    String name;
    Team team;
}

class Team {
    Long id;
    String name;
    List<Member>member members;
}
```

SQL을 직접 다루면 생기는 문제점

반복작업

- 새로운 필드가 추가되면 관련된 SQL을 전부 수정해야 함

```
class Member {
    Long id;
    String name;
    String age;
}
INSERT INTO member(`id`, `name`, ‘team_id’ ‘age’) VALUES ...
SELECT `id`, `name`, ‘team_id’, ‘age’ FROM member
UPDATE member SET …
```

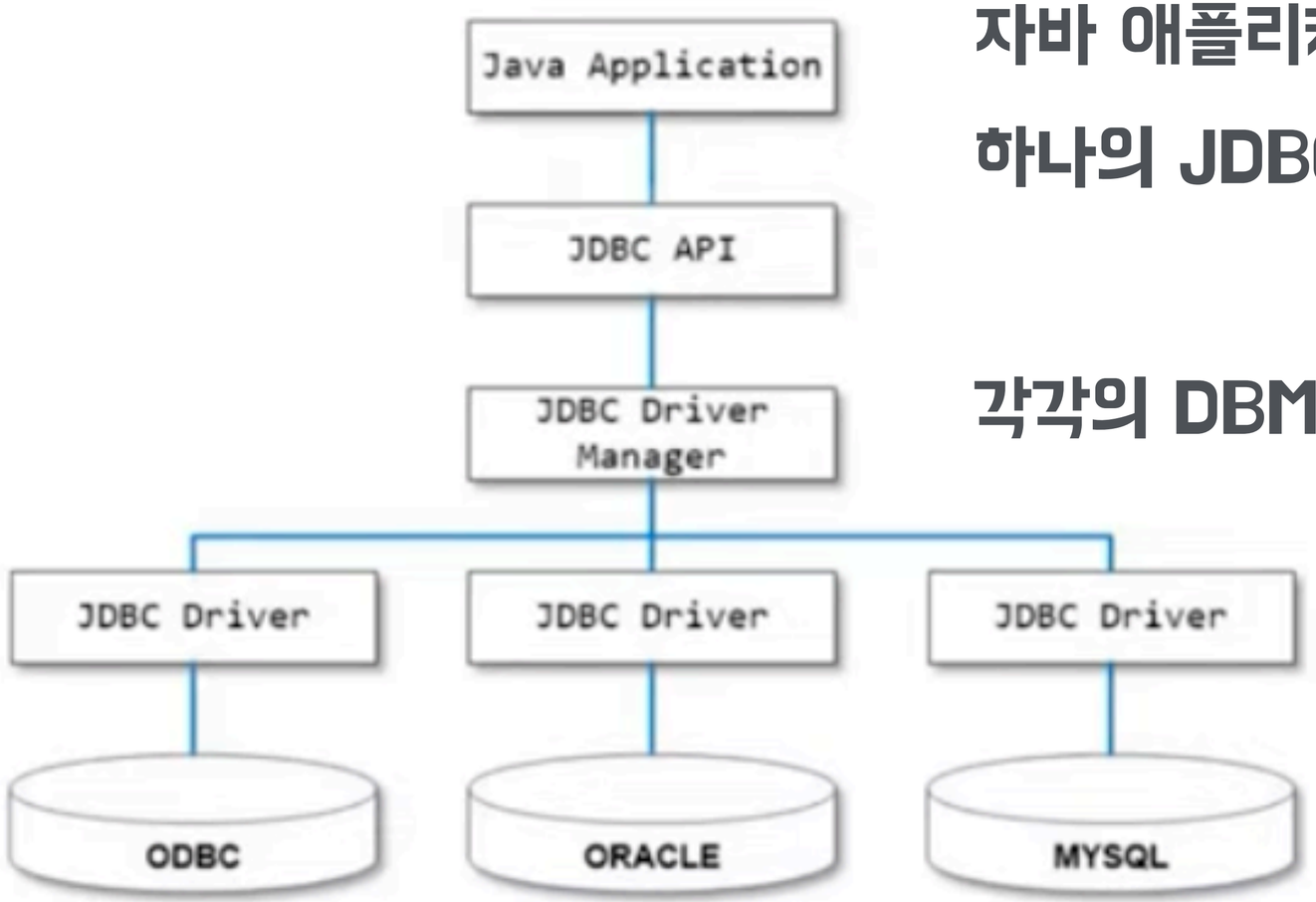
SQL을 직접 다루면 생기는 문제점

신뢰성

- SQL에 종속적이기 때문에 엔티티를 신뢰할 수 없다.

```
class Member {
    Long id;
    String name;
    String age;
}
INSERT INTO member(`id`, `name`, ‘team_id’ ‘age’) VALUES ...
SELECT `id`, `name`,‘age’ FROM member —-> ???
UPDATE member SET …
```

JDBC



자바 애플리케이션에서 DBMS의 종류에 상관없이

하나의 JDBC API를 이용해 DB 작업을 처리

각각의 DBMS는 이를 구현한 JDBC 드라이버를 제공

JDBC - 문제점

DB Connection 정의

DB Connection 오픈

SQL 지정

파라미터 선언 및 파라미터 값 제공

Statement 준비와 실행

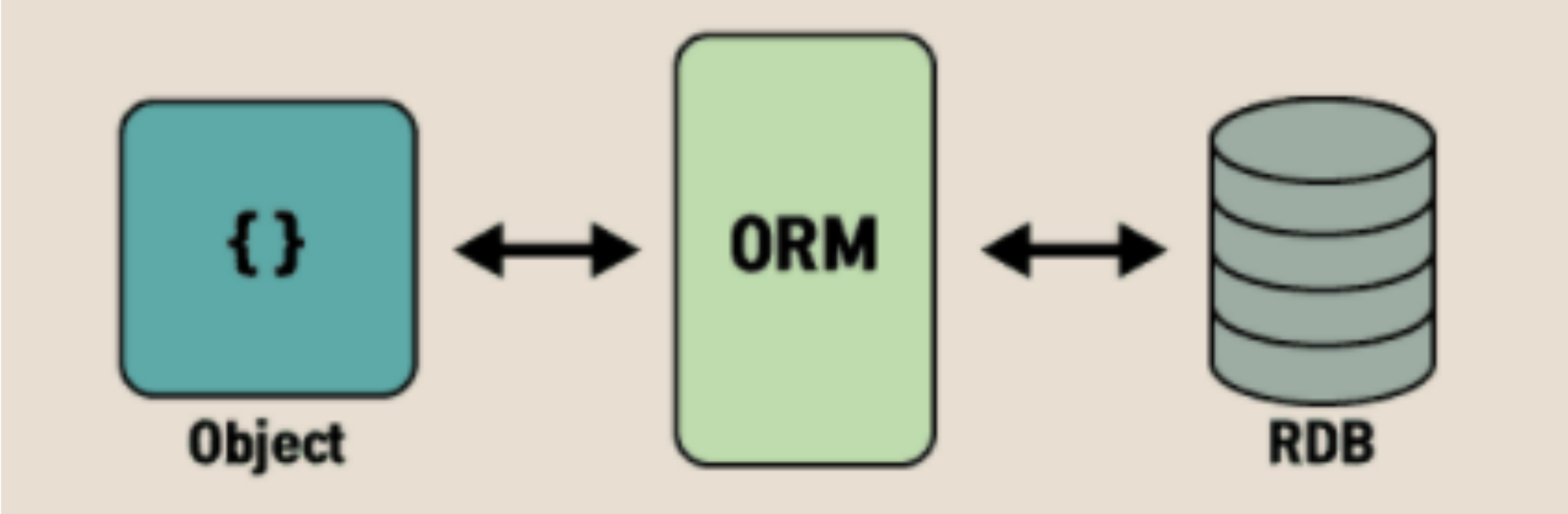
모든 예외 처리

연결, Statement, result 클로즈

SQL Mapper

Mybatis 등

ORM



요구사항

직원 관리 시스템을 만들어보자

멤버는 이름과 나이, 소속팀을 가진다

팀은 이름, 멤버를 가진다

멤버를 삭제, 저장, 조회 하는 기능이 있어야하고

팀에 멤버를 추가, 삭제 할 수 있어야 한다.