

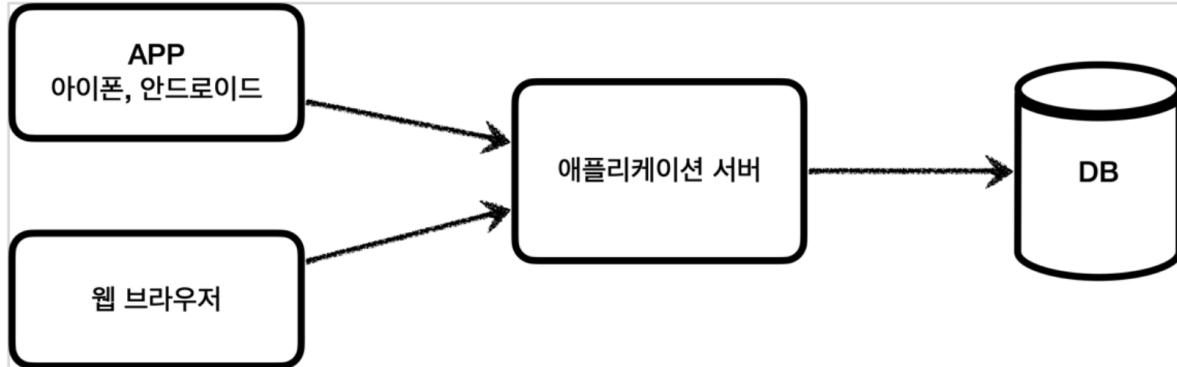
01. JDBC 이해

1. JDBC 등장 이유

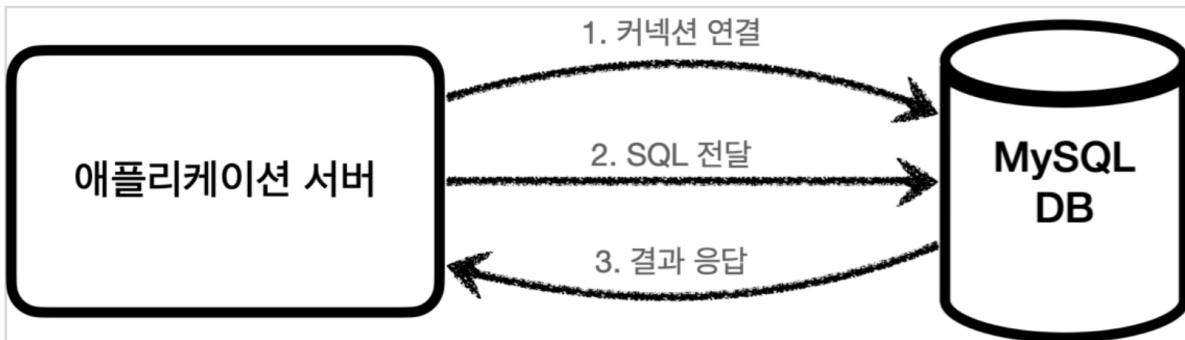
애플리케이션을 개발할 때 중요한 데이터는 대부분 데이터베이스에 보관함

일반적으로 DB 사용 메커니즘

클라이언트, 애플리케이션 서버, DB



애플리케이션 서버와 DB - 일반적인 사용법

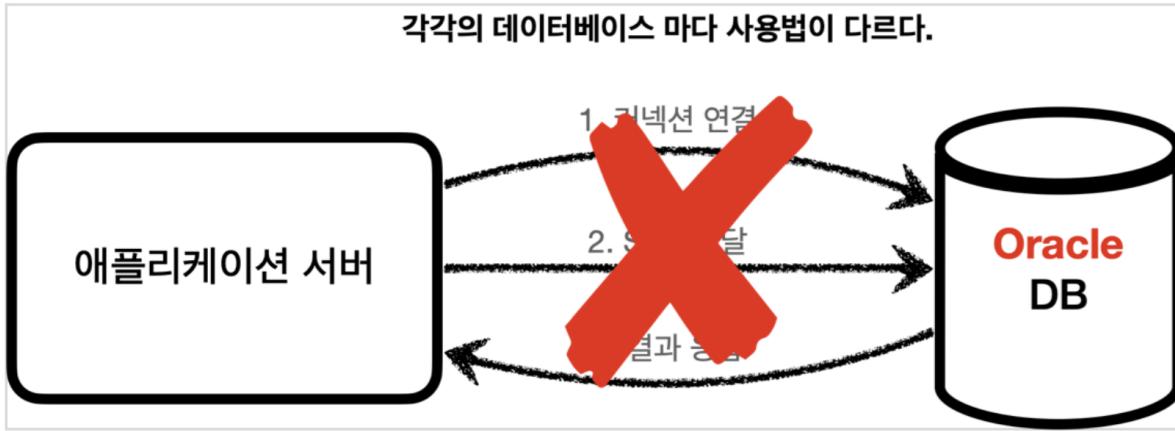


보통 TCP/IP를 통해서 커넥션을 연결하고

애플리케이션 서버는 DB가 이해할 수 있는 SQL을 연결된 커넥션을 통해 DB에 전달

DB는 전달된 SQL을 수행 → 결과를 응답 → 애플리케이션 서버는 응답 결과를 활용

옛날 방식의 문제점



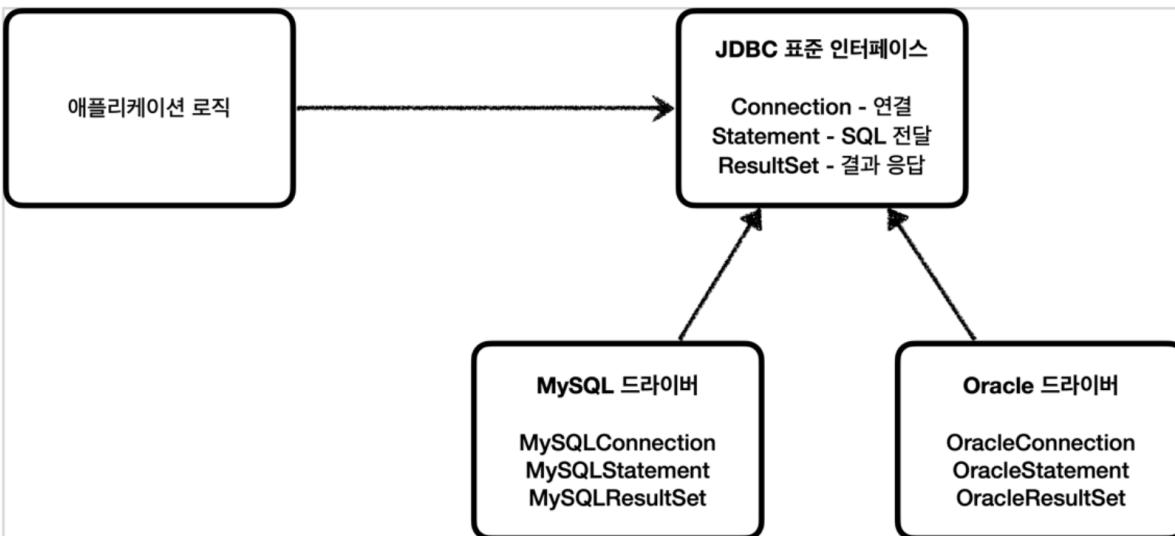
- 각각의 데이터베이스마다 커넥션을 연결하는 방법, SQL을 전달하는 방법, 결과를 응답 받는 방법이 모두 다름
- 데이터베이스를 다른 종류의 데이터베이스로 변경하면 애플리케이션 서버에 개발된 데이터베이스 사용 코드도 함께 변경해야함
- 개발자가 각각의 데이터베이스마다 커넥션 연결, SQL 전달, 그리고 그 결과를 응답 받는 방법을 새로 학습해야함

해결법?

이런 문제를 해결하기 위해 JDBC라는 자바 표준이 등장함

2. JDBC 표준 인터페이스

- JDBC(Java Database Connectivity)는 자바에서 데이터베이스에 접속할 수 있도록 하는 자바 API
- JDBC는 데이터베이스에서 자료를 쿼리하거나 업데이트하는 방법을 제공함



인터페이스의 기능

대표적으로 3가지 기능을 표준 인터페이스로 정의해서 제공한다.

- java.sql.Connection : 연결
- java.sql.Statement : SQL을 담은 내용
- java.sql.ResultSet : SQL 요청 응답

자바는 이렇게 표준 인터페이스를 정의해둠.

개발자는 이 표준 인터페이스만 사용해서 개발하면 됨
그런데 인터페이스만 있다고해서 기능이 동작하지는 않음.

이 JDBC 인터페이스를 각각의 DB 벤더 (회사)에서 자신의 DB에 맞도록 구현해서 라이브러리로 제공하는데, 이것을 JDBC 드라이버라 함
예를 들어서 MySQL DB에 접근할 수 있는 것은 MySQL JDBC 드라이버라 하고, Oracle DB에 접근할 수 있는 것은 Oracle JDBC 드라이버라 함
요놈을 함께 써줘야한다

....

ORM 기술

ORM 기술

ORM은 객체를 관계형 데이터베이스 테이블과 매핑해주는 기술

이 기술 덕분에 개발자는 반복적인 SQL을 직접 작성하지 않고, ORM 기술이 개발자 대신에 SQL을 동적으로 만들어 실행해줌

추가로 각각의 데이터베이스마다 다른 SQL을 사용하는 문제도 중간에서 해결해줌

메커니즘

- 개발자가 sql query를 직접 전달하는게 아니라 (회원 등) 객체를 JPA에 전달을 함
 - 마치 자바 컬렉션에 객체를 넣을때처럼 JPA에 넣어주면 됨
- 전달된 객체에는 맵핑 정보같은게 있는데 그걸 보고 요청한 query를 직접 만들어냄
- 그리고 JDBC를 통해서 DB로 전달함