

# JSP 프로그래밍

10차시

– JDBC 프로그래밍



## □ 명령어 종류

- 데이터 정의 언어(DDL: Data Definition Language)
  - 관계형 데이터베이스의 구조를 정의
- 데이터 조작 언어(DML: Data Manipulation Language)
  - 관계형 데이터베이스의 검색 및 업데이트 등 데이터 조작
- 데이터 제어 언어(DCL: Data Control Language)
  - 관계형 데이터베이스의 데이터에 대한 액세스를 제어

# 지난시간 복습



## ☐ Insert

- 데이터 삽입, 저장

## ☐ Select

- 데이터 조회

## ☐ Update

- 데이터 수정

## ☐ Delete

- 데이터 삭제



# 옛날의 상황을 가정해봅시다.



## □ DB에 접속해봅시다.

### ○ Oracle

- userId, userPass로 접속하세요.

### ○ MSSql

- i, p로 접속하세요

### ○ MySql(MariaDB)

- id, pass로 접속하세요

# 옛날의 상황을 가정해봅시다.



## □ 데이터 입력해주세요.

### ○ Oracle

- Java에서 insert를 사용해서 입력하세요.

### ○ MySql(MariaDB)

- Java에서 save를 사용해서 입력하세요.

### ○ MS-Sql

- Java에서 nice를 사용해서 입력하세요.

# 옛날의 상황을 가정해봅시다.



- 명령어 자체는 어디까지나 가정의 상황이었지만 예전에는 DBMS의 벤더가 각각 만든 Library(혹은 API)로 개발을 해야했다.
- 당연히 개발자는 너무 힘들었다.
- 회사에서 Oracle을 사용하다가 MariaDB로 옮긴다고 하면??      ➔ 퇴사하라고??

# 옛날의 상황을 가정해봅시다.



- 커넥션을 만들어서 사용하자. 서로 좋잖아?
- 접속은 getConnection(url, user, pass)로 하자
- 데이터 저장은 execute(sql)로 하자
- 데이터 조회는 ResultSet이라는 객체로 하자

**표준, 미리 만든 형태로 각각 구현하도록 강제하자.**

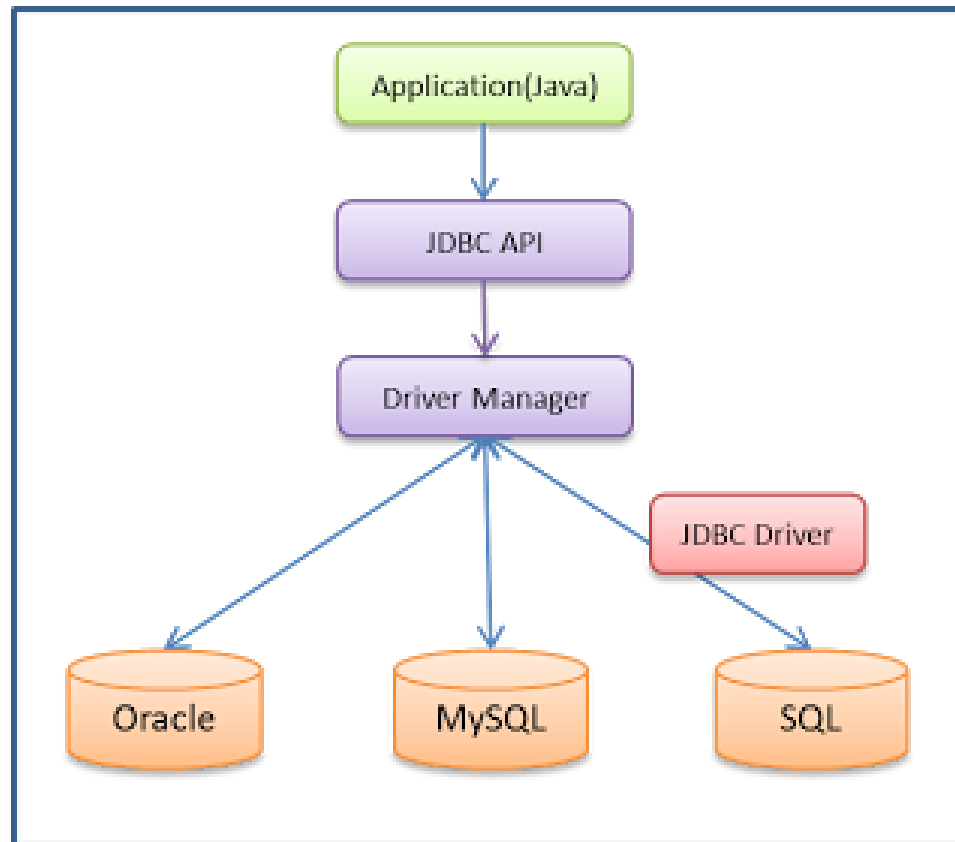


- JDBC(Java Database Connectivity)
- Java에서 데이터베이스에 접속할 수 있도록 하는  
Java API
- JDBC에는 구현 클래스가 거의 없고, 대부분이 바로  
**인터페이스**이다. 그래서 표준이란 표현을 쓴다.
- 인터페이스를 가져다 각 벤더가 구현해서 class를 만  
들어야 한다.





## □ JDBC 구조





## □ JDBC 역할

- 데이터베이스 접속 지원
- 명령문(SQL문) 만들고, 실행하는 것을 지원
- 쿼리의 결과에 대한 처리



## □ JDBC 로직

- JDBC 드라이버 로딩
- 데이터베이스와 연결(Connection)
- Query 실행을 위한 Statement 객체 생성
- Query문 실행
- Statement 객체 해제
- Connection 해제