JDBC

Java Database Connectivity

> ☑️ **자바에서 데이터베이스에 접속할 수 있도록 하는 자바 API**

1. JDBC Driver Loading작업

사용하려는 Database에 맞는 driver class를 등록

- class라는 class가 있다.
- Driver Manager class

2. DBMS 연결 (실제로 데이터 베이스 연결)

성공하면 connection instance가 생성

- 부화가 많이 걸린다.
- 다 사용한 후 반드시 자원을 해제 (close)
- Database의 session이 종료될 수 있도록 처리를 해주어야 한다.

JDBC URL → 문자열 이용

3. Statement 생성

- **일반 Statement**
- **prepared Statement**
- **callable Statement**
  - stored procedure를 호출할때 사용

4. Query 실행

- **Statement를 이용해서 SQL Query를 DBMS에 전달해서 실행**
- `execute()`

    - 다 가능하다!
    - 내가 사용할 SQL문이 뭔지 모른다 OR 동적으로 변한다 하면 이거 사용!
- `execute Query()`
    - `SELECT`
- `execute Update()`
    - `INSERT,UPDATE,DELETE`

5. 결과처리 (Result Set)

- `ResultSet rs = Pstmt.executeQuery();`
- 여기서 `rs`는 컬럼명 pointing하고있다.
- 다음행으로 내려갈려면 `rs.next()` 하면 밑으로 내려간다.
    - **return값은 boolean값이다.** 내려갈 수 있으면 `true` 못내려가면 `false`
    - 기본적인 **Result set**은 한번 내려가면 **올라올 수 없다.**
- **Scrollable cursor는 DATABASE와 JAVA가 연동되어있어서 DATA가 변경되더라도 반영이된다.**
- `rs.getString("name")`
- `re.getString(2)`

6. Connection Pool

- 동시에 많은 사용자에 대해서 Database처리를 제공하려면??
    - Thread를 이용해서 각각의 connection을 하나씩 부여해야한다.
- 각각 하나씩 부여하지말고 하나의 connection을 공유하면 안될까??
    - connection에 Transaction을 걸 수 있다. 한 사람이 작업하고 있는데 다른사람이 rollback시키거나 commit 시킨다면 오류가 생긴다.
- 그래서 사용하는것이 **Connection pooling**기법!!
    - Connection을 많이 만들어 놓고 대여하고 반납하는 형식으로 사용하자!!
    - 속도 향상
    - 자원의 효율성

Layered Architecture

MVC

Moder View Controler

가장 대표적인 모델