

## \* DBMS 개념

↳ Database Management System

데이터 보관과 조회를 서비스하는 프로그램

## \* Database

- 실시간 접근 가능 : 데이터 처리 요청에 즉시 응답
- 동시에 풍부 : 여러 사용자가 풍부
- 데이터의 독립성 : 애플리케이션에 비중적
- 일관성 : 데이터 처리 작업이 끝난 후 데이터의 값은 유일한 상태를 유지해야 한다.
- 무결성 : 유일하지 않은 데이터의 등록과 변경, 삭제는 제한
- 보안성 : 사용자 인증과 권한 관리 ↳ 예) 기사를 작성할 수 있다. ↳ 예) 기사를 작성할 수 없다.

auth

(authentication) (authorization)

ID/PWD 일치

기능의 이용권한 검사

유일한 사용자

등록 허용?  
변경 허용?

:

## \* DBMS

- 데이터베이스를 관리하는 S/W  
- 예) Oracle, MySQL, PostgreSQL, MS-SQL, Altibase, Tibero, Cubrid 등

↳ 예) 일관한 데이터 관리

↓  
실제적인 환경의 구조 관리

SQL 언어로 명령어 작성



명령어



결과 출력

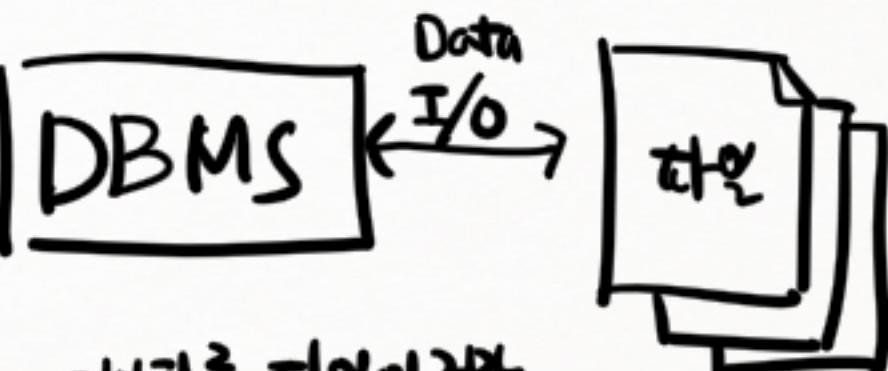
↑  
MariaDB  
client  
App.

데이터작업의 요청

DBMS 통신규칙

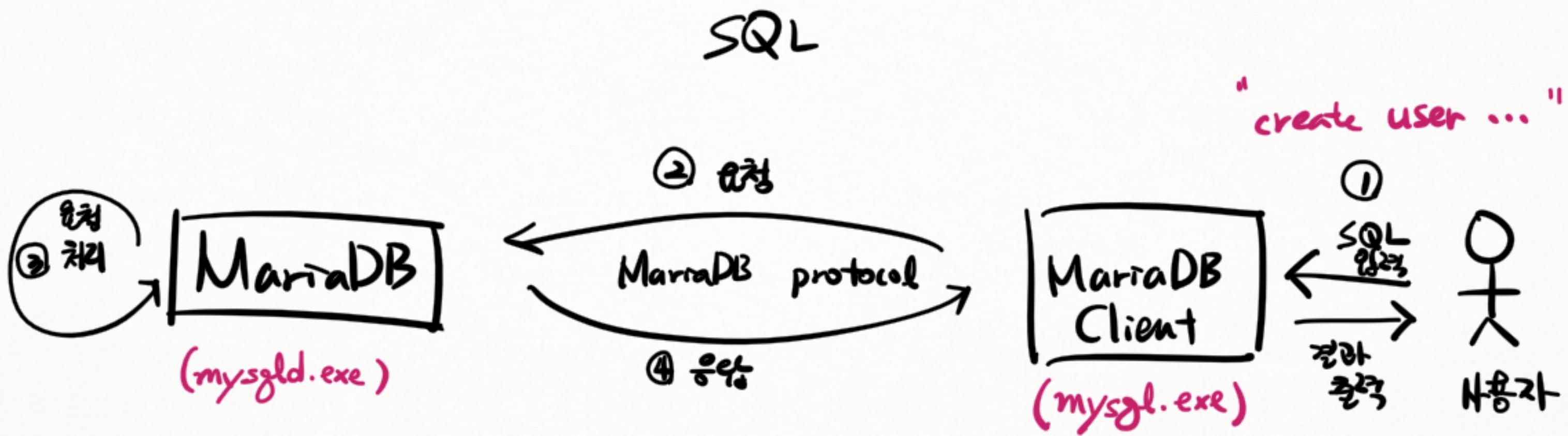
요청 처리 결과

MariaDB

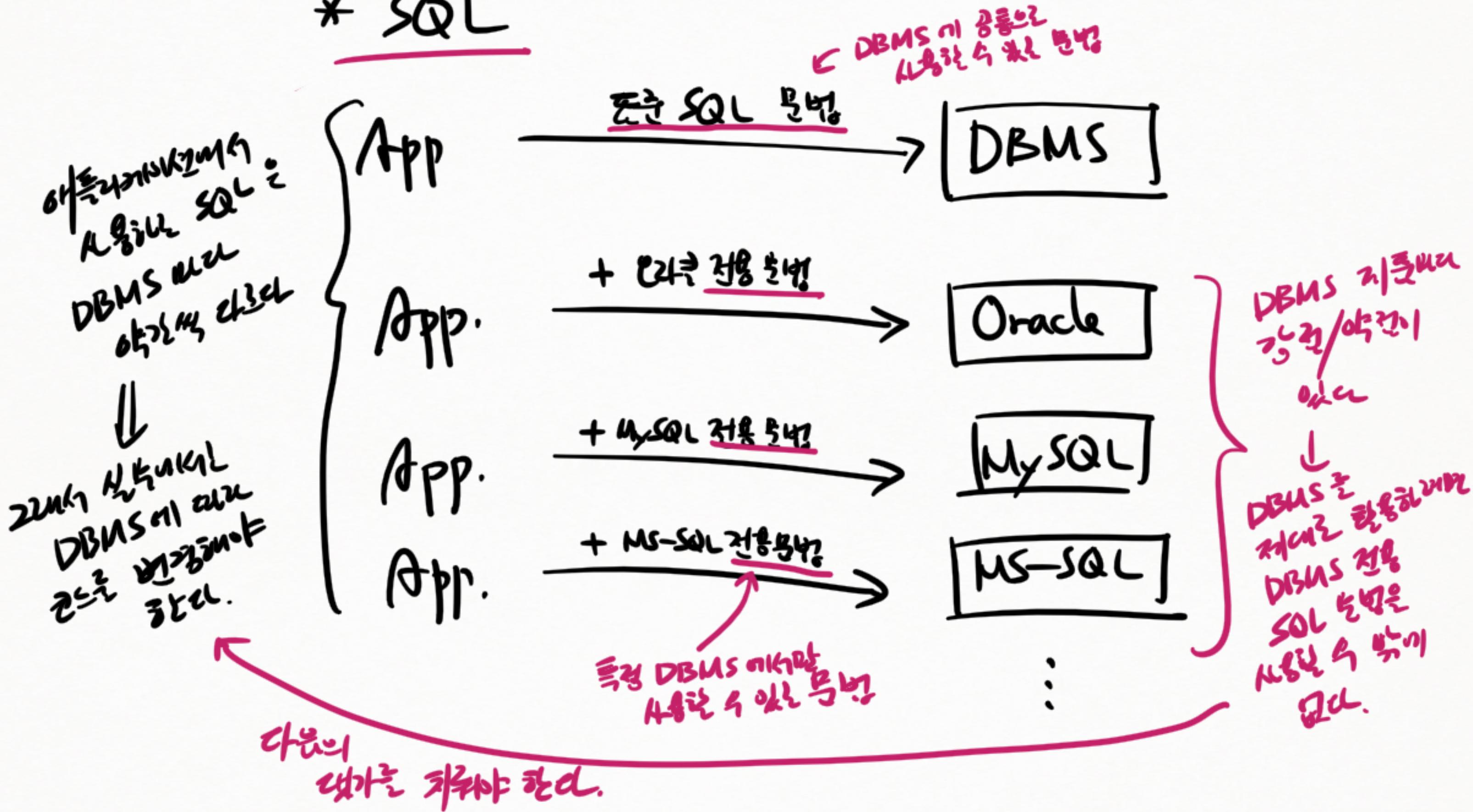


- 데이터를 파일에 저장
- 파일에 저장된 데이터를 조회/변경/삭제
- 데이터베이스의 기능을 제공

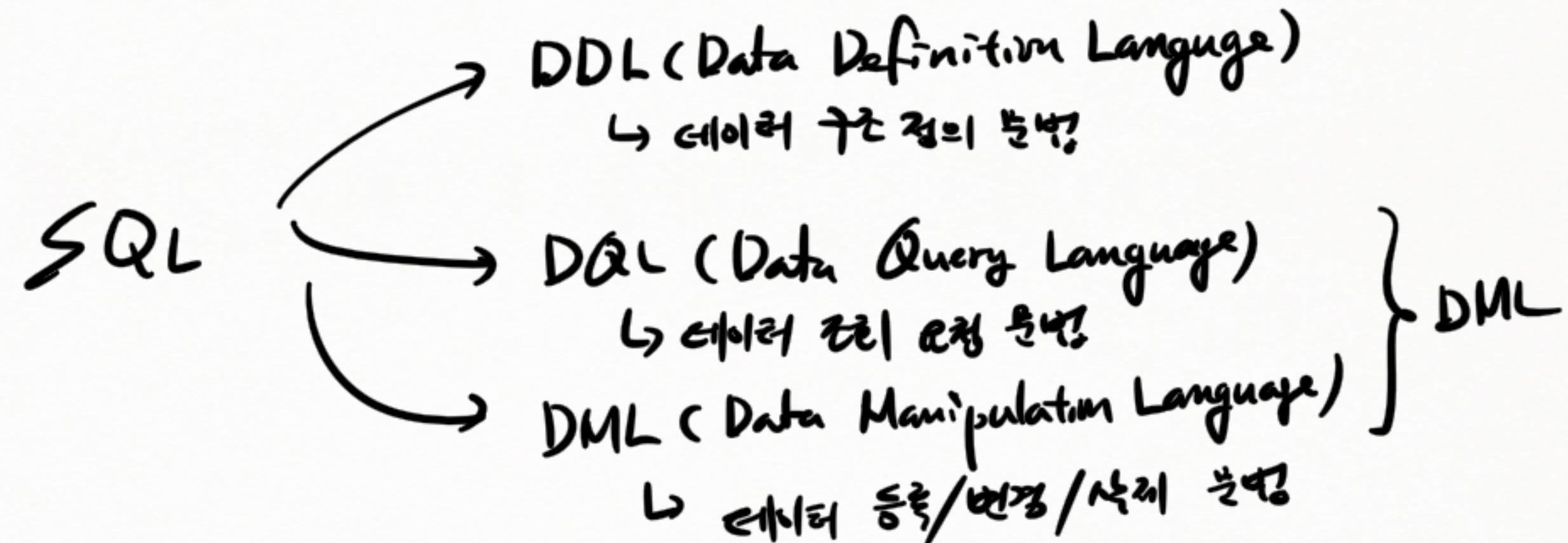
## \* DBMS 와 SQL



## \* SQL



## \* SQL



## \* SQL

① DDL → 데이터를 저장하는 다음 DB 개체를 정의한다

②  $\begin{cases} DQL \\ DML \end{cases}$  → DB 개체의 데이터를 다룬다  $\Rightarrow \begin{cases} \text{Table} \\ \text{View} \\ \text{Procedure} \\ \text{Function} \\ \vdots \end{cases}$

### \* SQL 테스트 준비

## ① 사용자 추가

```
> create user 'study'@'%' identified by '1111'; <
```

## ② 레이터를 저장할 레이터 베이스 생성

③ 데이터베이스를 사용할 사용자의 권한을 지정

> grant all on studydb.\* to 'study'@'localhost' ; ↴  
↑ 권한 범위      ↑ 데이터베이스      ↴ DB 품목      ↴ 사용자 아이디      ↴ 접속 PC 주소

\* DDL (Data Definition Language)

## ① 테이블 생성

레이블 쟁쟁

```
create table test01 (
    name varchar(50) not null,  

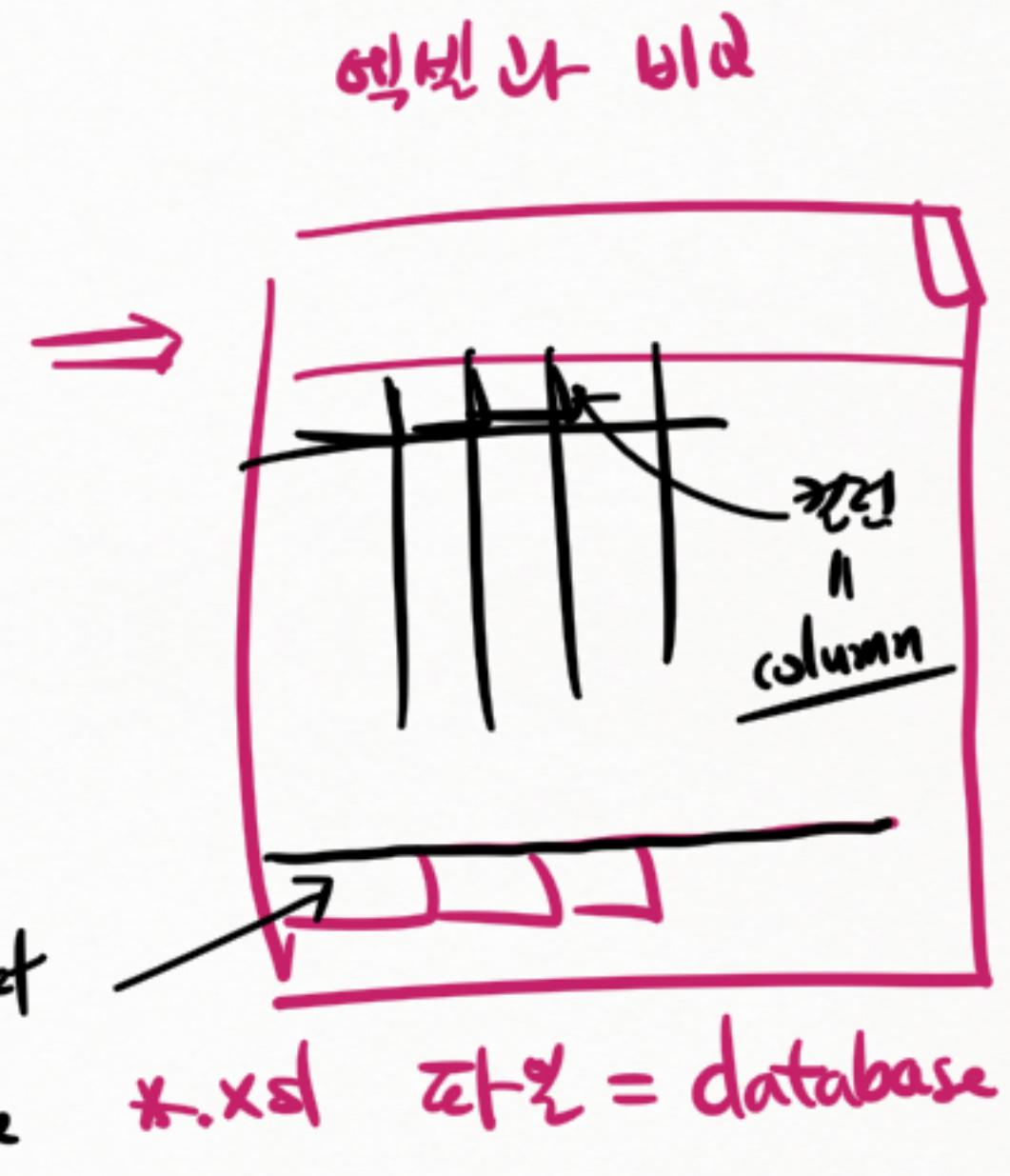
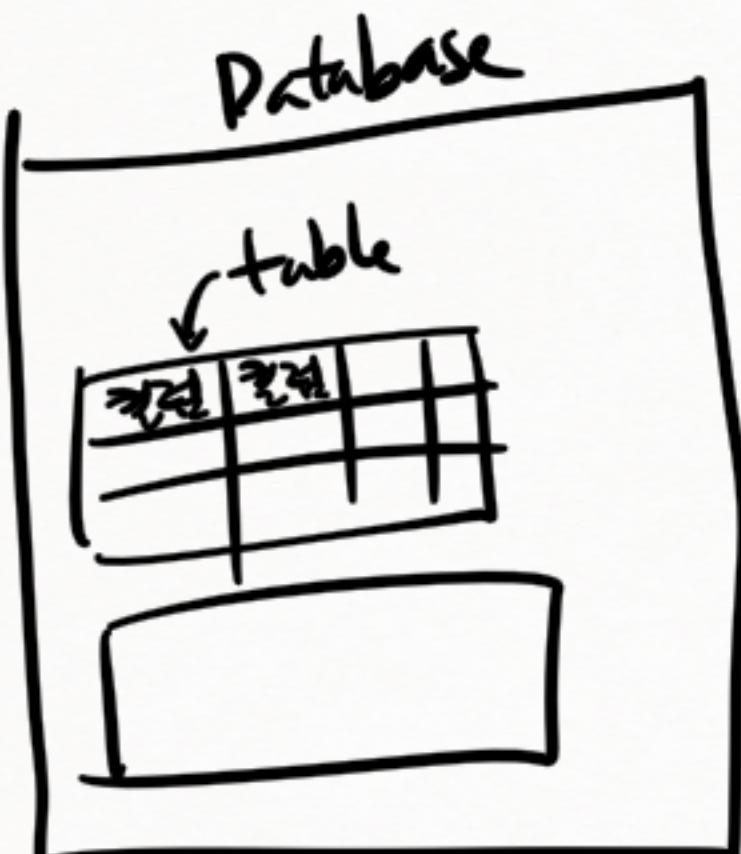
    kor int not null,  

    eng int not null,  

    math int not null,  

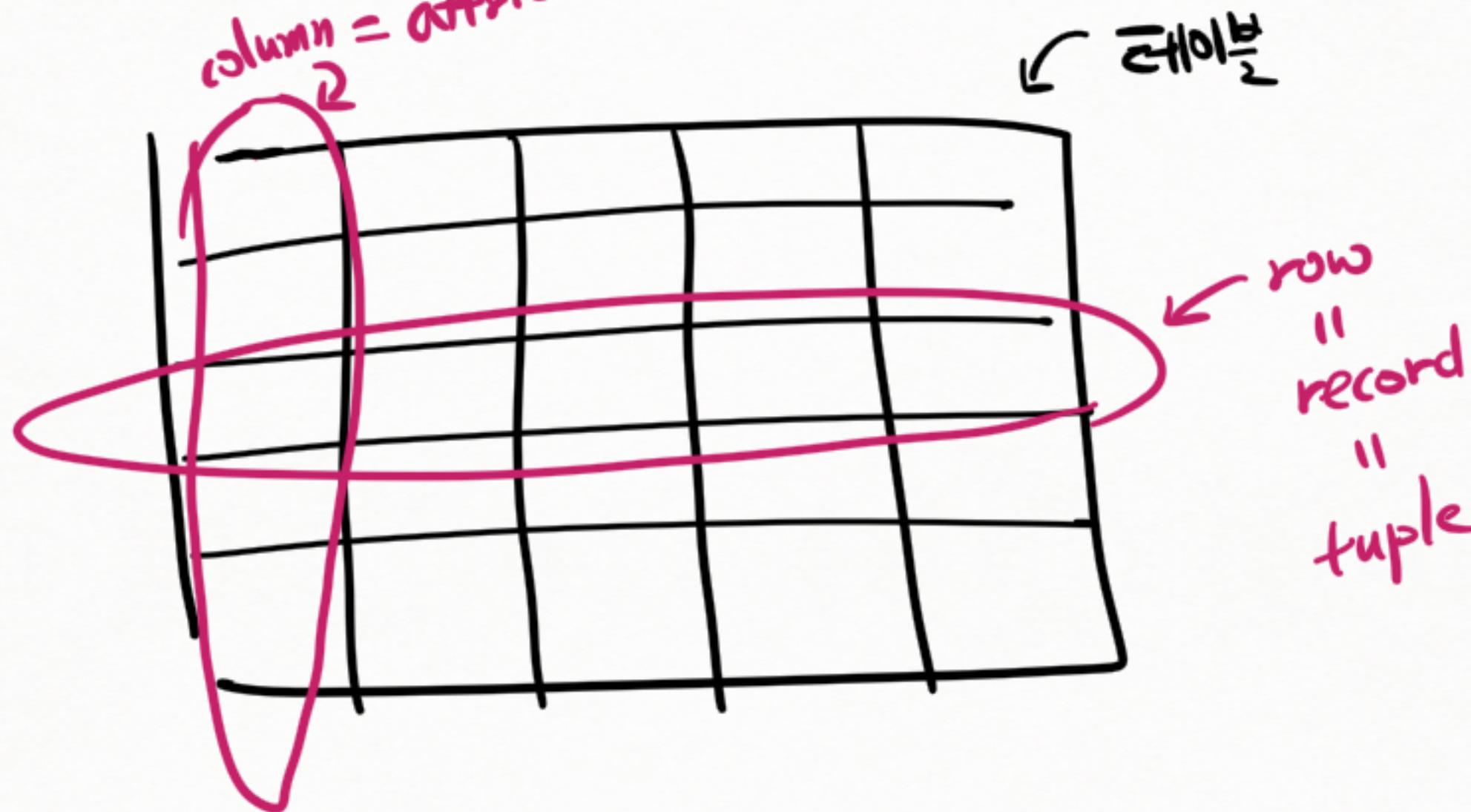
    sum int not null,  

    aver float not null);
```



\* 레이블과 row, column

column = attribute = field



\* 테이블

✓ create table 테이블명 ( 컬럼, ...) ← 테이블 정의

✓ drop table 테이블명 ← 테이블 삭제

✓ alter table 테이블명 변경사항 ← 테이블 변경

✓ describe 테이블명 ← 테이블 정보 조회  
    "desc"

## \* insert

insert into test1(no, name) values(1, 'aaa')

↑  
레코드명

↑  
컬럼명

↑  
컬럼에 들어갈 값

컬럼을 나열할 순서다 일치해야 한다.

\* select

select no, name from test1

          ↑↑  
          컬럼명

          \*  
          ↑  
          모든컬럼

\* key, candidate key, primary key / alternate key

키/C→I

artificial key

데이터를 구현할 때 사용한 건전진\*

key

[이메일]

[아이디]

[주민번호]

[이름, 전화번호]

[아이디, 전화번호]

[이메일, 주민번호]

[이메일, 이름]

[주민번호, 이름]

[이름, 아이디, 전화번호]

[이름, 주민번호]

\* key, candidate key, primary key / alternate key

키/C→I

artificial key

데이터를 구현할 때 사용한 키는 \*

key

candidate key

↳ 키/C

\* [이디]

\* [아이디]

\* [주민번호]

\* [이름, 전화번호]

[이디, 이름]

[아이디, 전화번호]

[주민번호, 이름]

[이름, 주민번호]

[이디, 주민번호]

[이름, 아이디, 전화번호]

\* key, candidate key, primary key / alternate key

키/C

candidate key

데이터를 구분할 때 사용한 키

Alternate  
key

→ 키로 설정되지 않은  
내부 키로 다른 키를  
'대안키'가 된다

key

candidate key  
↳ 키/C

[아이디]

\* [주민번호]

\* [이름, 전화번호]

↑  
DB 관리자가  
주 키로  
설정한 키

"Primary key"

## \* Artificial key

제시글 : 제목, 내용, 작성일, 작성자, 조회수, 좋아요수

제시글 번호

Primary key?

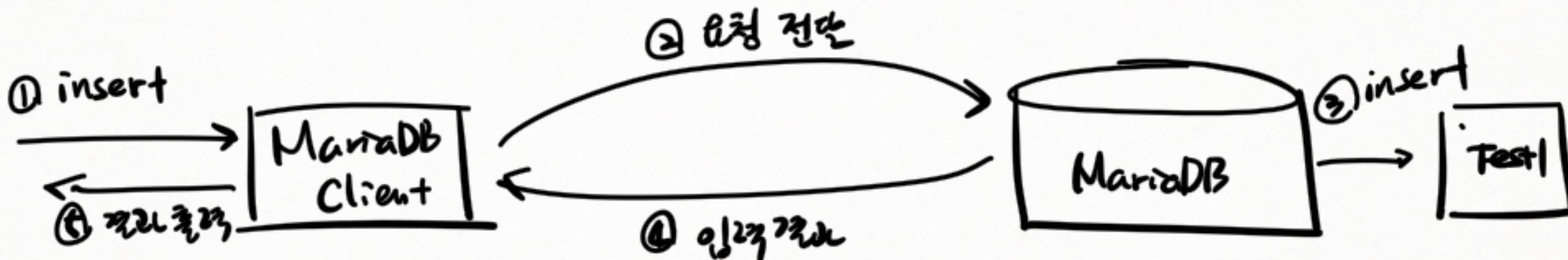
보통히 사용할 key를 없을 경우

Artificial (인공적)  
key

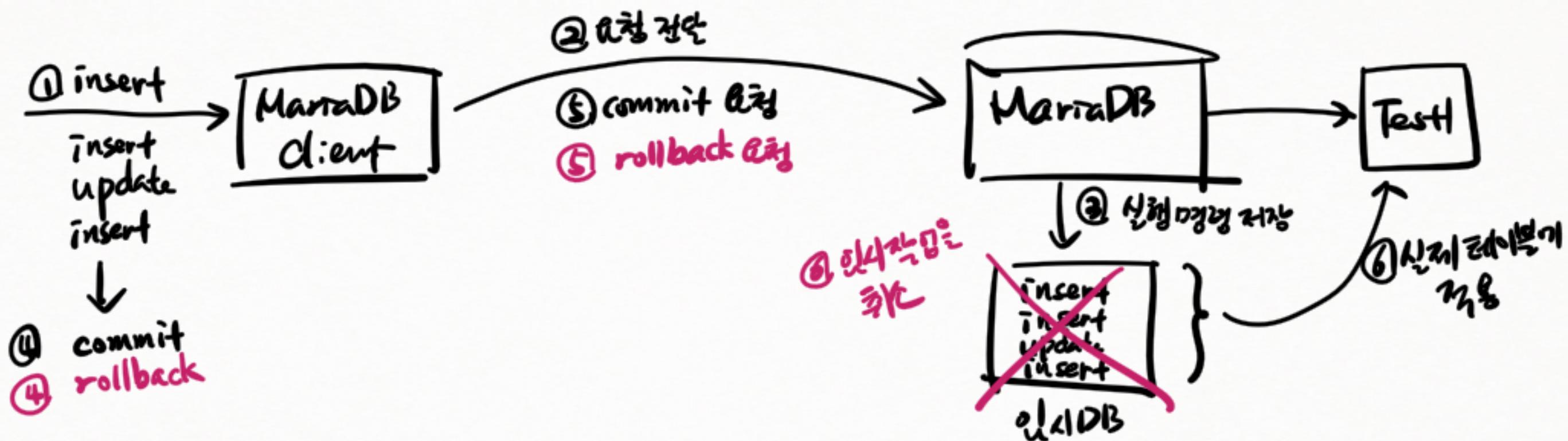
= 인공적 번호를 만들어  
PK로 사용한다  
↳ ex) 일련번호

## \* Commit

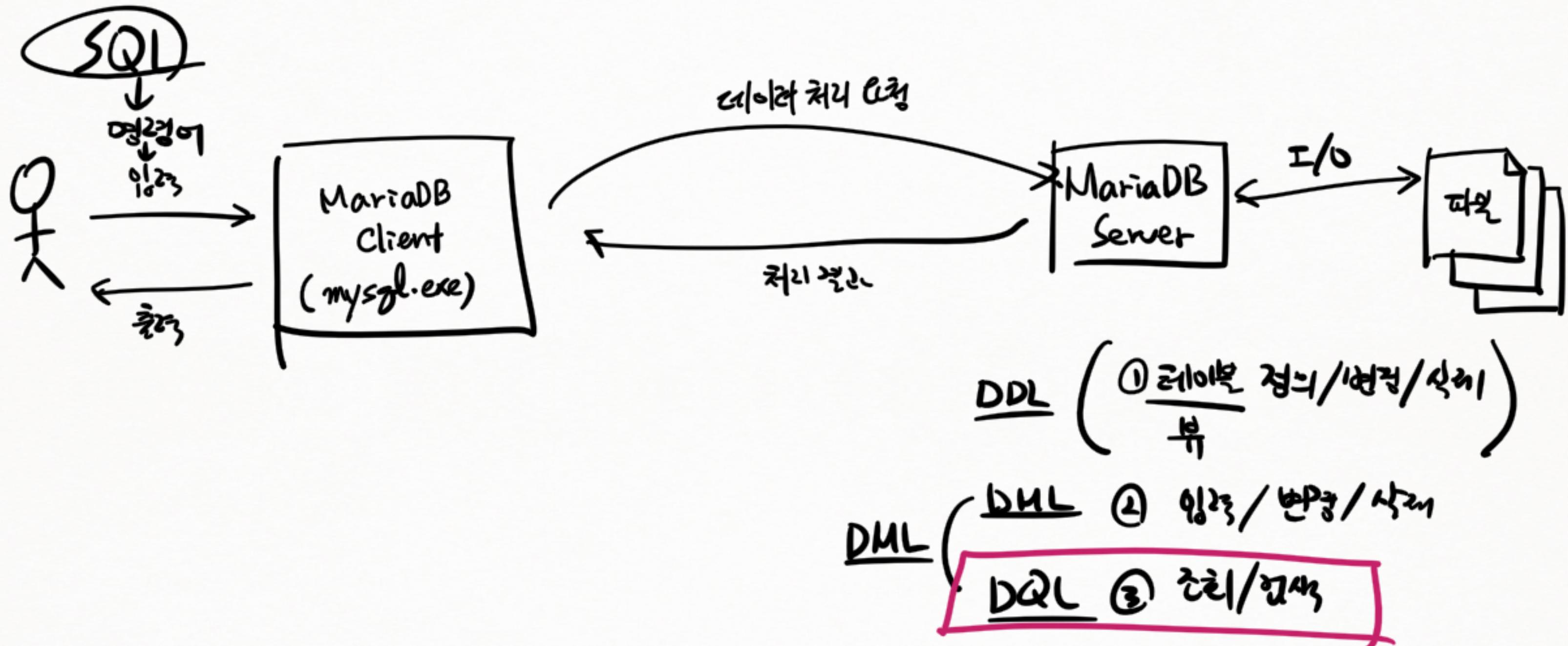
① autocommit = true



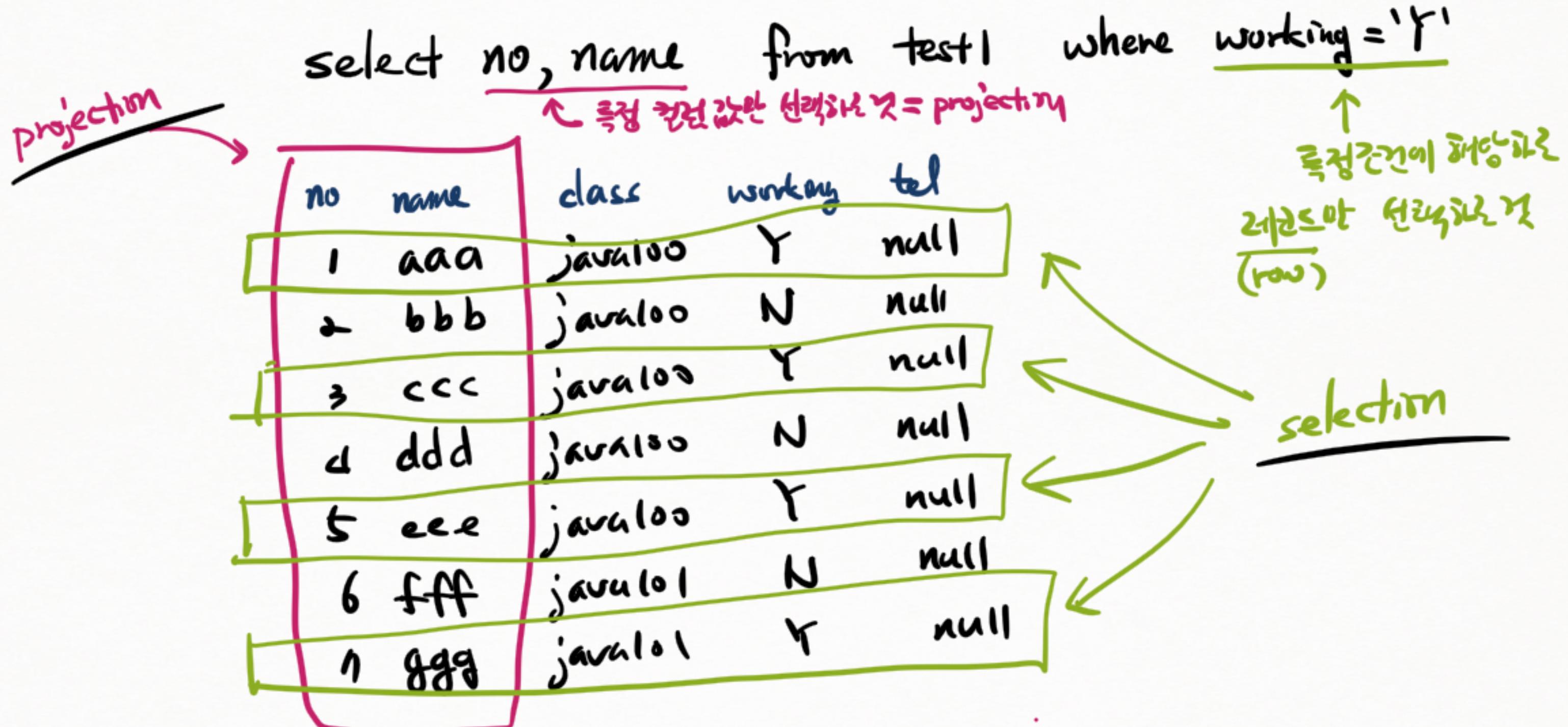
② autocommit = false



\* select

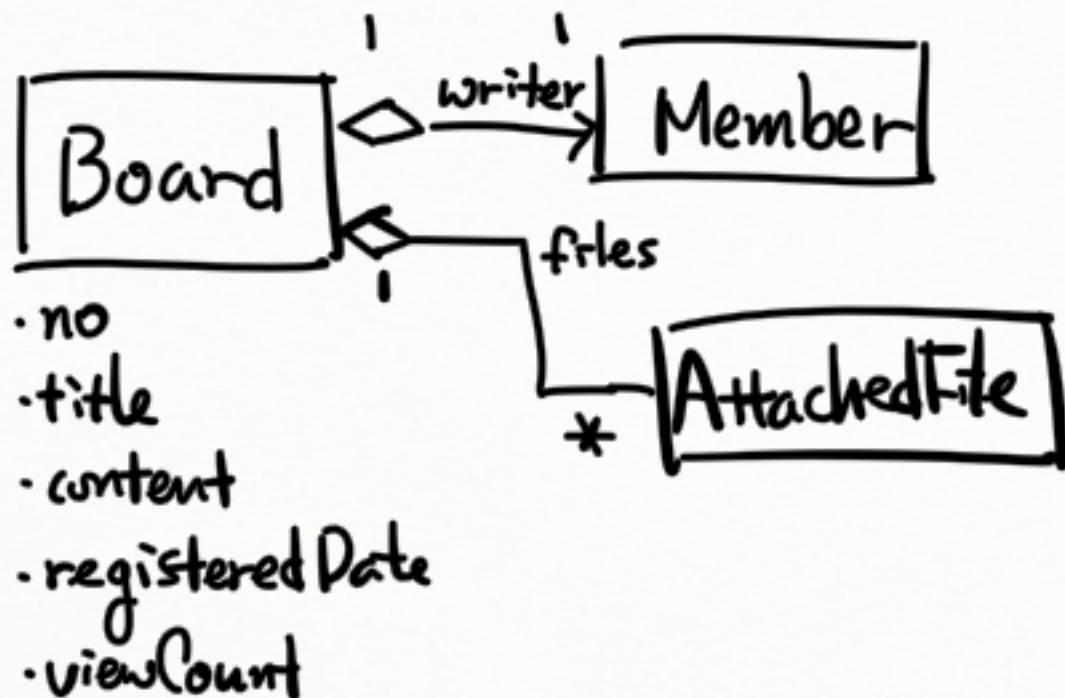


## \* selectin / projection



## \* Foreign Key

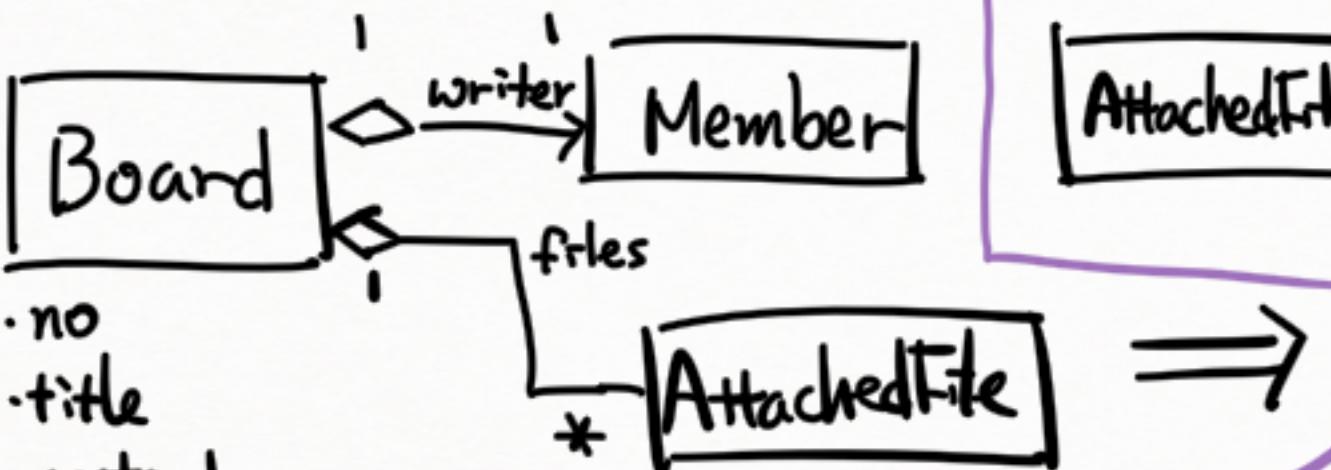
① 자식 가족



```
class Board {  
    int no;  
    String title;  
    String Content;  
    Date registeredDate;  
    int viewCount;  
    Member writer;  
    List<AttachedFile> files;  
    ...  
}
```

## \* Foreign Key

① 자식 키체



- no
- title
- content
- registeredDate
- viewCount

② 테이블 간의 관계 (ER-Diagram; ERD)



- 1번 게시글
  - 2번 게시글
  - 3번 게시글
- 100번 조회수
  - 101번 댓글수
  - 102번 유저수

ERD 특징:

① Information Engineering Notation

공식

↓  
정준으로 일정 사실을  
학문으로 체계화·나누어서  
연구하는 것

↓  
해당 분야에 종사하고 사는들이  
사용하는 언어로 드러낸다

\* 객체 간의 관계와

테이블 간의 관계는 다르다!

↓  
아마히 일부는 관계화시키는 데로  
관계를 설정.

프로그래밍 언어  
위한 목적으로 객체간의  
관계를 설정.

\* Foreign Key (외부키) - 다른 테이블의 pk 참조



(PK) 번호	파일명	제작자(FK)	(PK) 번호	제작자	제작자(FK)	(PK) 번호	이름
11	aaa.gif	101	101	aaaa	1	1	홍길동
12	bbb.gif	102	102	bbbb	1	2	김꺽정
13	ccc.gif	102	103	cccc	3	3	유재석
14	ddd.gif	103	104	dddd	4	4	안우진
15	eee.gif	103					

## \* 커먼 쿵복

Board							
no	title	content	rdt	f1	f2	f3	f4
1	aaa	—	—	a.gif	b.gif		
2	bbb	—	—	x.gif	y.gif	z.gif	t.gif
3	ccc	—	—	o	o	o	o
4	ddd	—	—	m1.gif	m2.gif	m3.gif	m4.gif m5.gif

이렇게 같은 종류의 파일은 여러 개 저장하기 위해 커먼 쿵복으로 선언된 경우

첨부파일이 없음에도 5개의 커먼이 존재하기 때문에 메모리 낭비

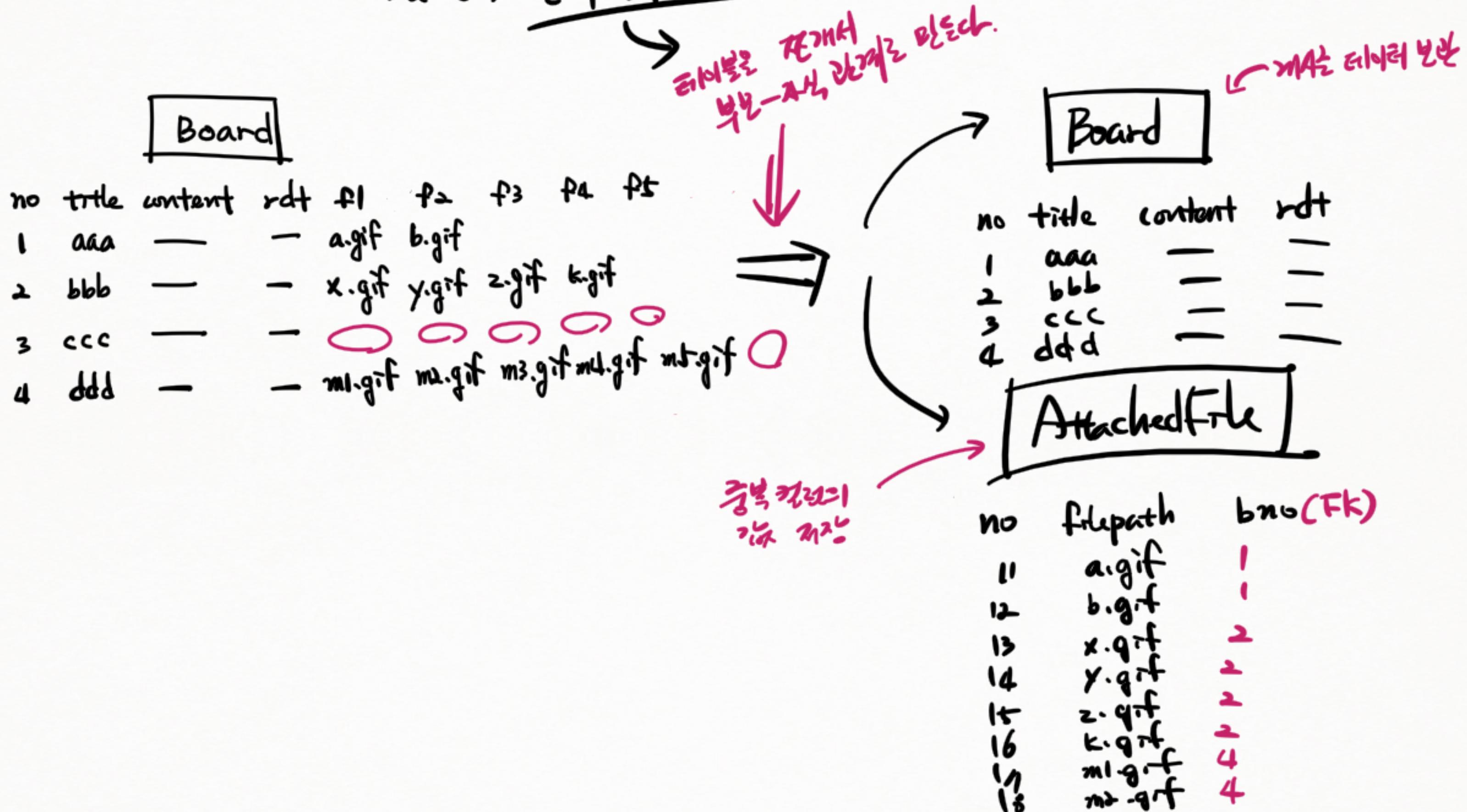
커먼이 부족하여 첨부파일을 더 저장할 수 있다

다음의 위치 발생

이전 커먼 쿵복이나 레이어 쿵복 등에 허용

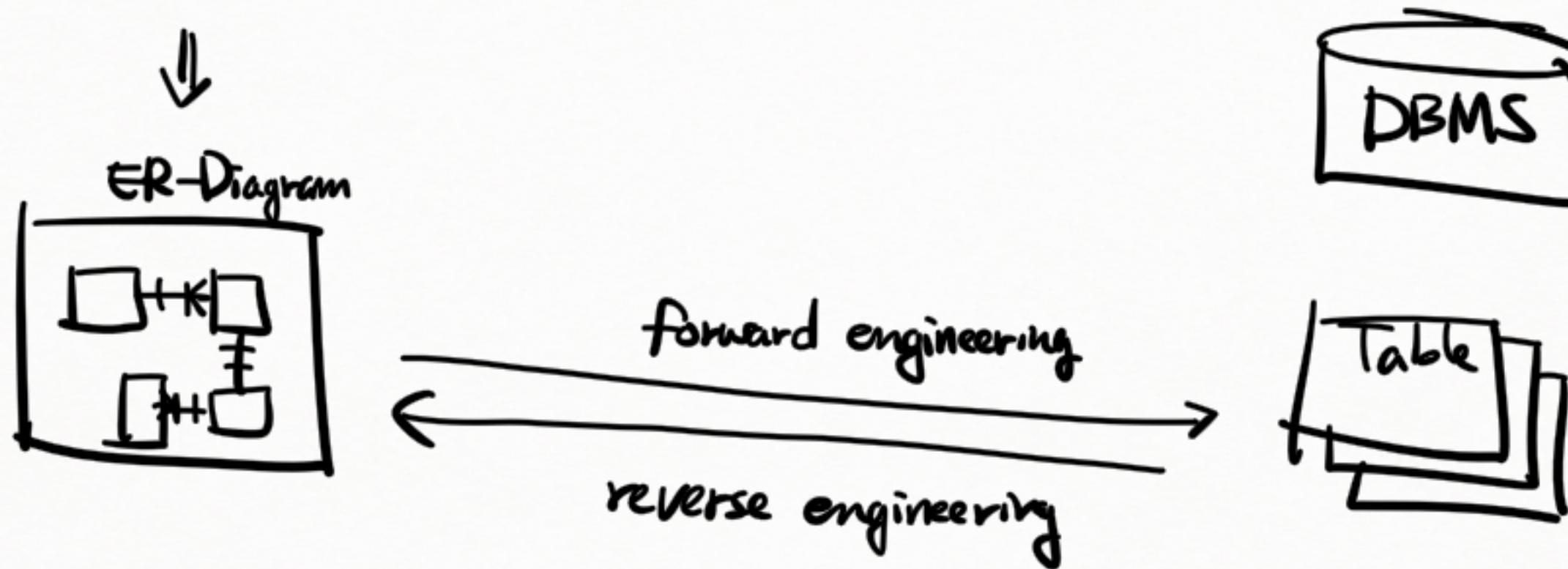
장기화 수행!

## \* 첨부 중복 문제 해결



\* ERD : forward / reverse engineering

제작ツ : ER-Win, eXerd, ...



\* SQL 조인 테스트를 위한 데이터의 ERI

예) 교육센터 관리 시스템

