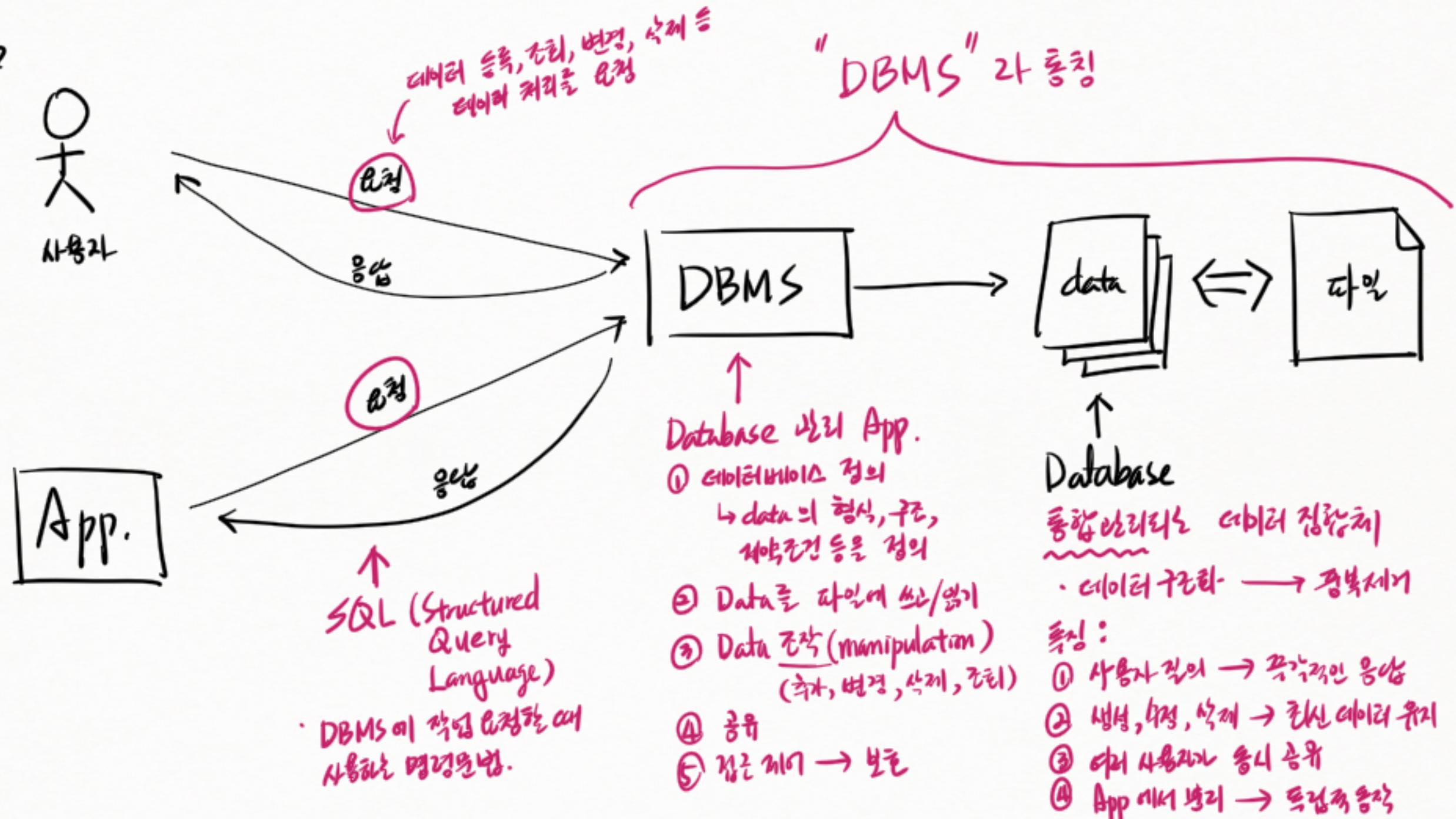


DBMS
Database
Management
System

* Database, DBMS, SQL

★ DBMS 를 사용하는 이유?

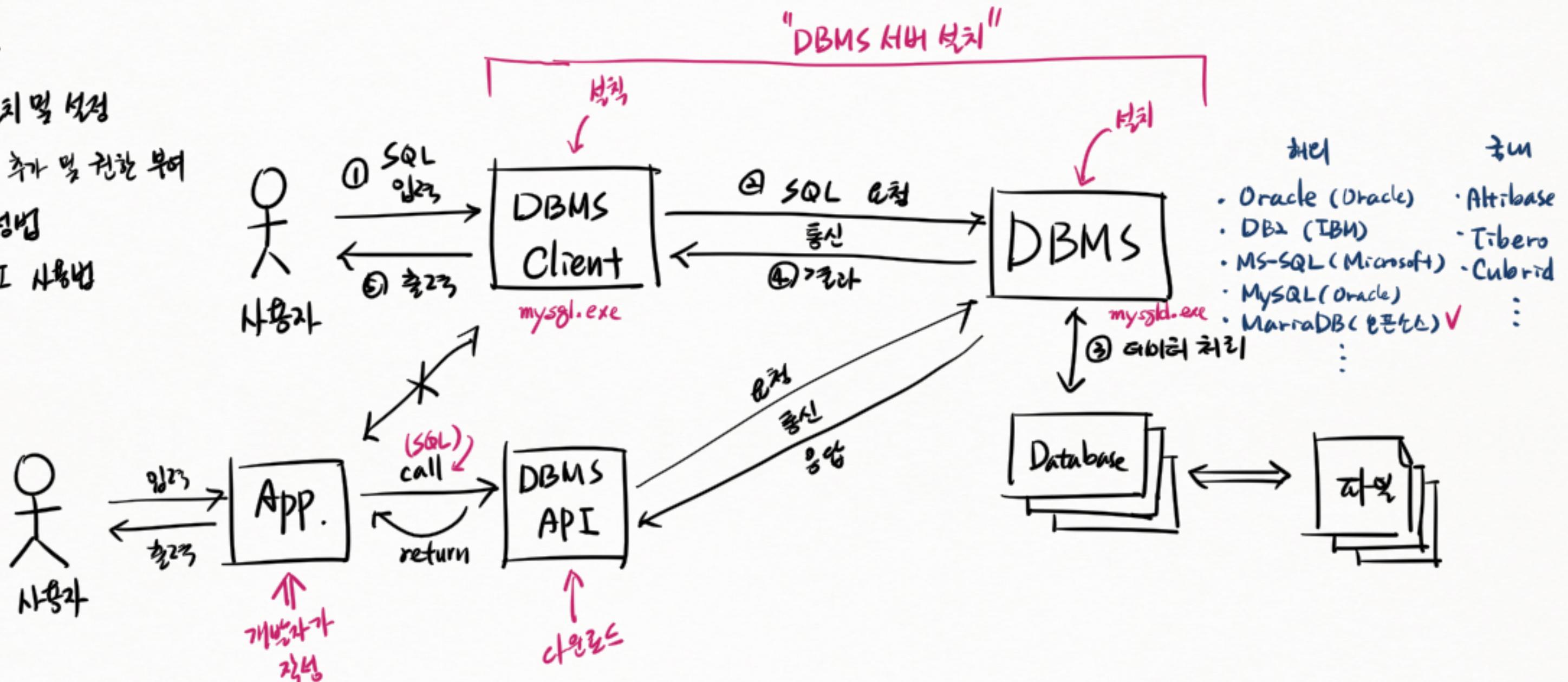
- 직접 파일 I/O 을
프로그래밍 하기
불편한 점 있음.
- 데이터를 차례대로
관리하는게 번거롭다
- 프로그래밍 언어에
생각보다
밀접한(종속된) 명령으로
작동할 수 있음.



* DBMS 教材

* 336 118

- ✓ ① DBMS 설치 및 설정
 - ✓ ② Database 추가 및 권한 부여
 - ✓ ③ SQL 작성법
 - ④ DBMS API 사용법



SQL Structured
 Query
 Language

* SQL

↳ DBMS에 데이터 처리 명령을 네트 채 사용하는
명령을 표준 언어

* 흔한 SQL

= SQL 표준 언어 + DBMS 특수 언어
⇒ DBMS 외 SQL 표준 언어 사용 가능
⇒ App 작동할 때
DBMS에 맞춰
SQL은 사용할 수 있다.

SQL

DDL (Data Definition Language)

↳ 데이터, 뷰, 프로시저, 트리거 등을
정의하는 언어, DDL

DML (Data Manipulation Language)

↳ 데이터를 데이터를 삽입, 수정, 삭제 등 데이터 조작

DQL (Data Query Language)

↳ 데이터의 내용을 조회

} DML이나
DQL

* $\text{entity} \approx \text{table}$
column = attribute

name	kor	eng	math	sum	aver

row = record = tuple

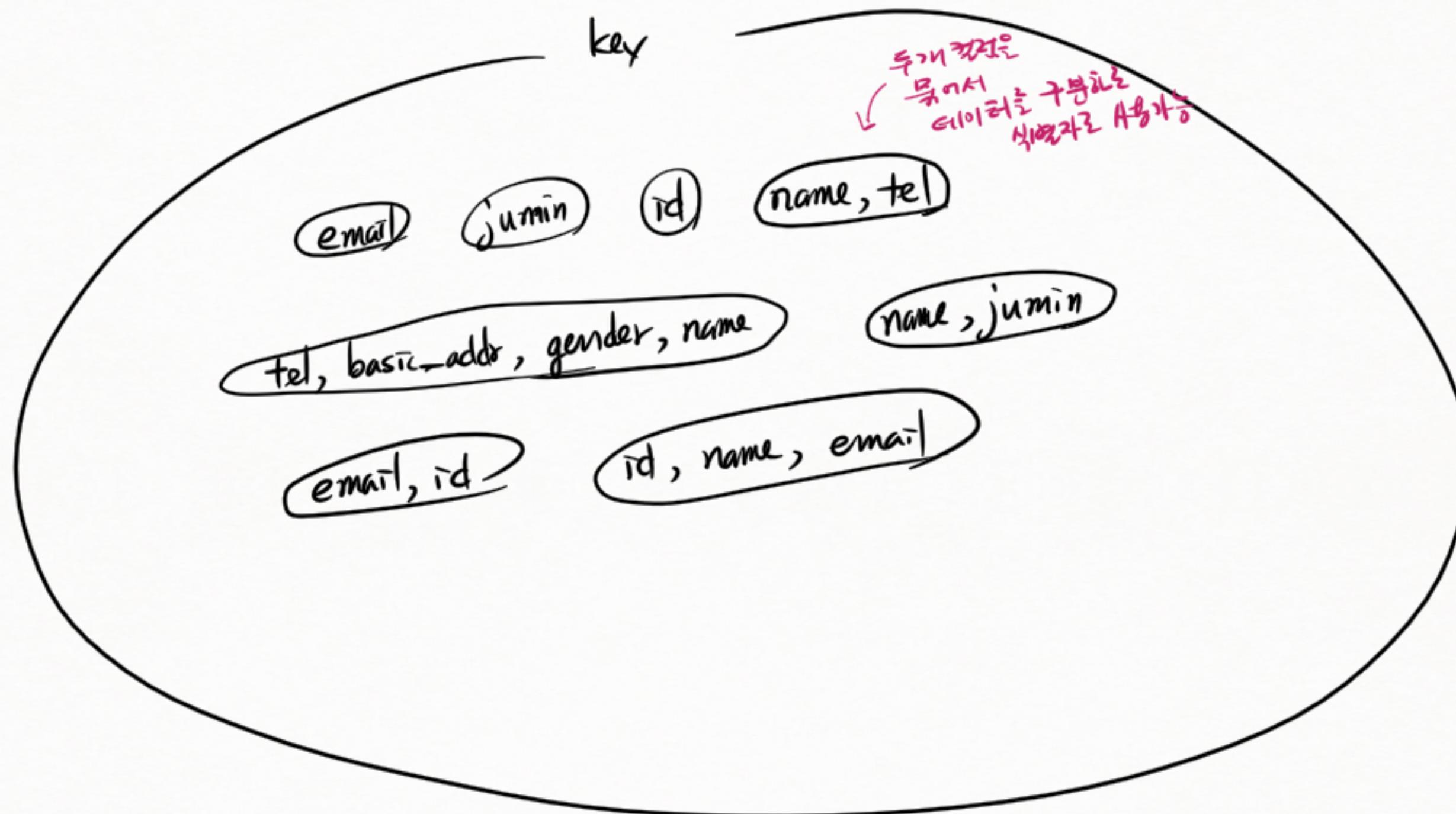
* 진수와 decimal

(Java)

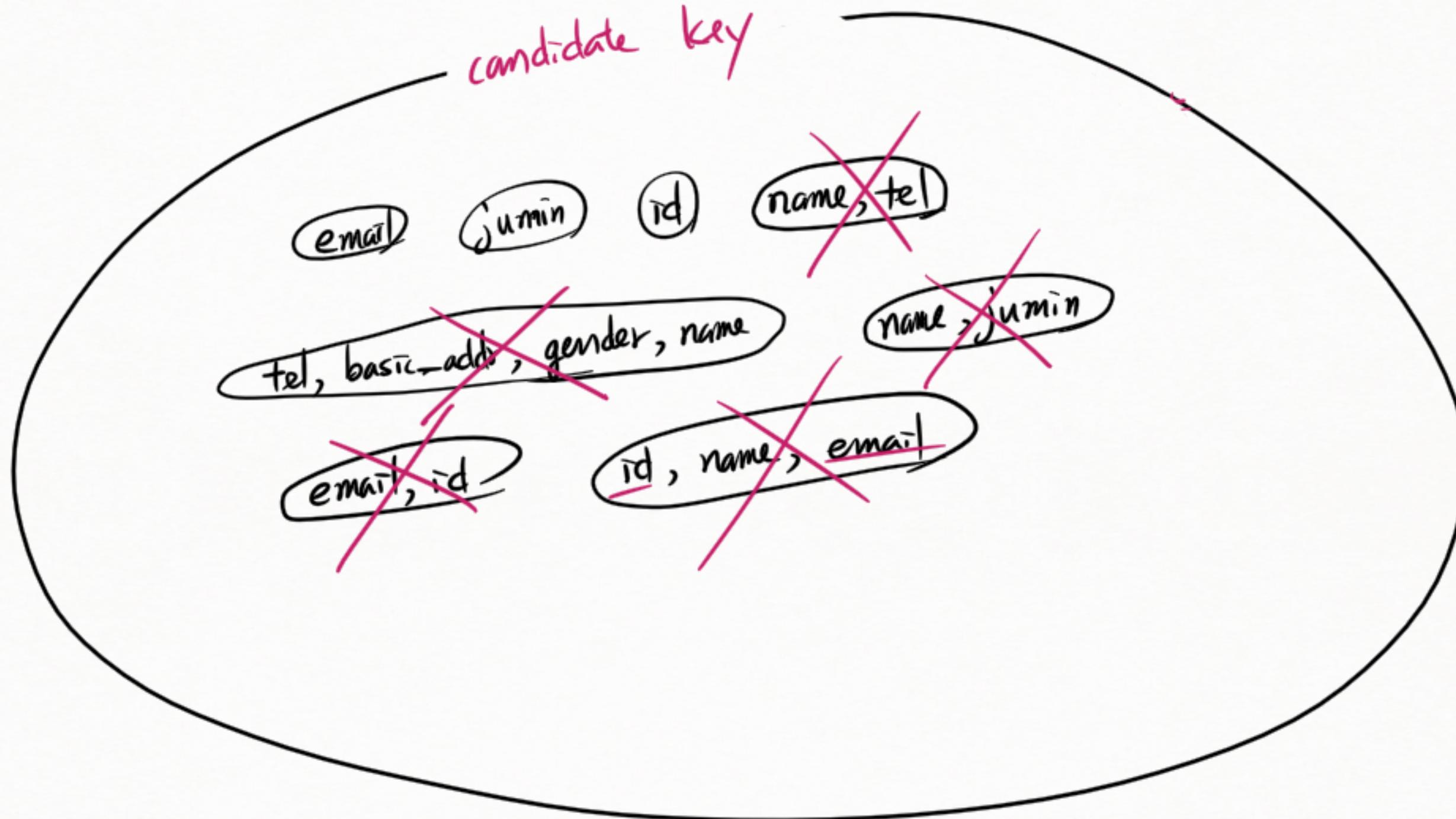
	진수	10진수
8진수	017	15
10진수	17	17
16진수	0x17	23
2진수	00010111	23

12 ← 정수
 3.14 ← 부동소수점 } 10진수
 decimal

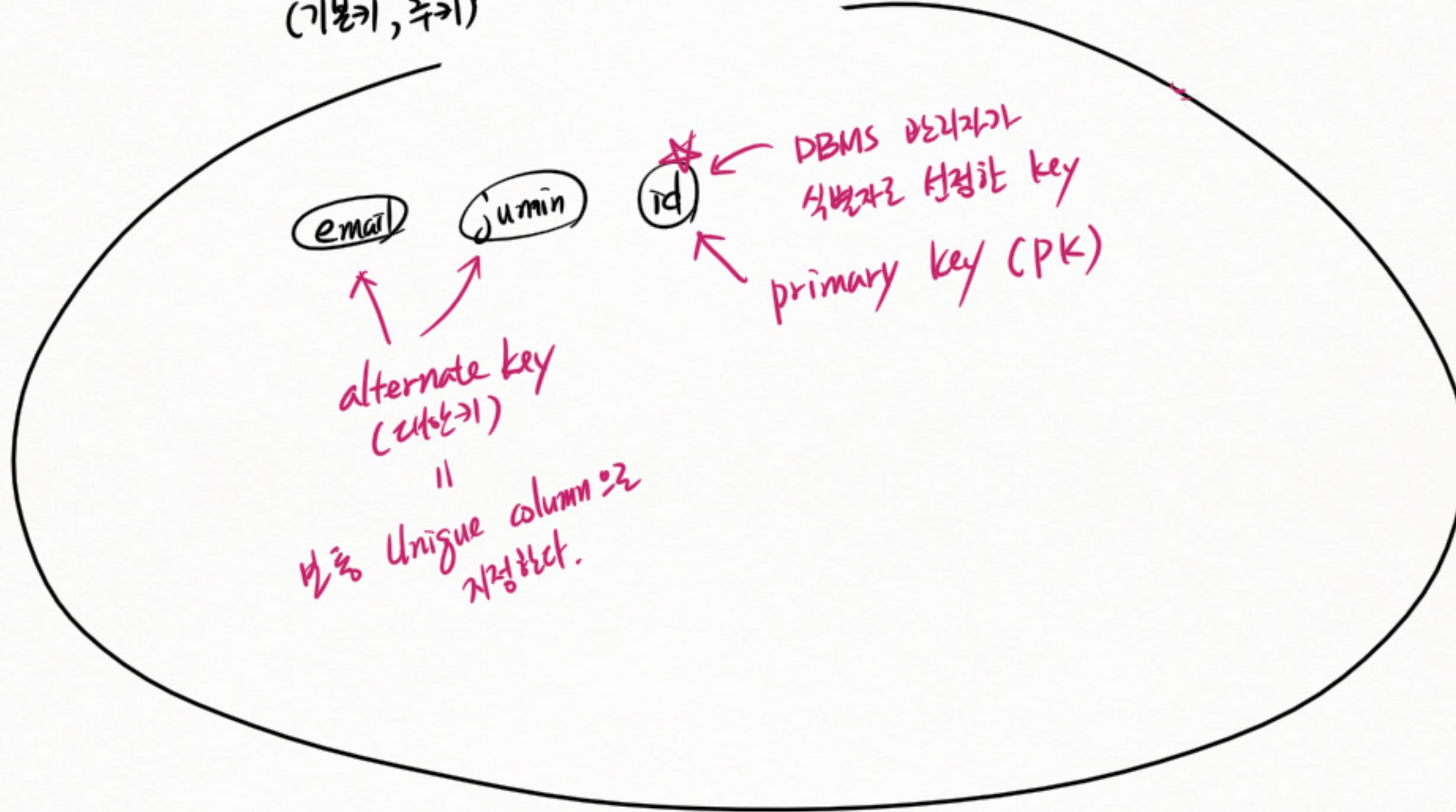
* key : 데이터를 구분할 때 사용하는 ключ값



* candidate key (후보키, 주어지기): 키가 유일성을 지닌 key

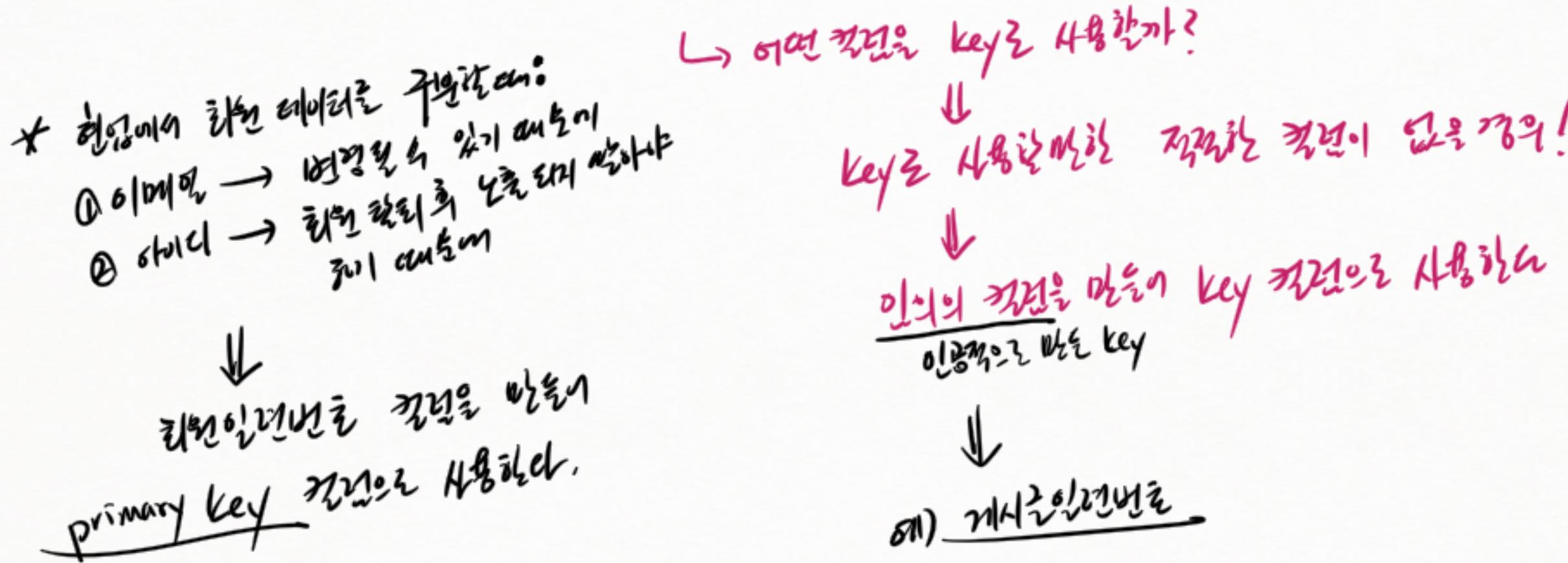


* Primary Key: 주보기 중에서 고유값이 설정되는
(기본키, 주기)



* artificial key (인공키)

Board 테이블: 세종, 세종, 애산, 조희숙, 장성호



기사

PK
no

인공키
번호
PK 기본키
설정

name, age

alternate key

PK는 아님
PK로 충분하지
않다
unique 컬럼으로
지정해야 한다.

기금 암호 중에는 여러 가지 작업을 같은 단계로 다해야 하는 경우가 있다
예) [초안 → 결재] \Rightarrow 주는 암호 commit

Transaction

↳ 여러 모의 데이터 변경 작업을 한 번에 끝내는 것

```

graph TD
    A[작업] --> B[준비]
    A --> C[반복]
    B --> D[rollback]
    C --> E[rollback]

```

데이터 변경 작업을 한 번에 끝내는 것

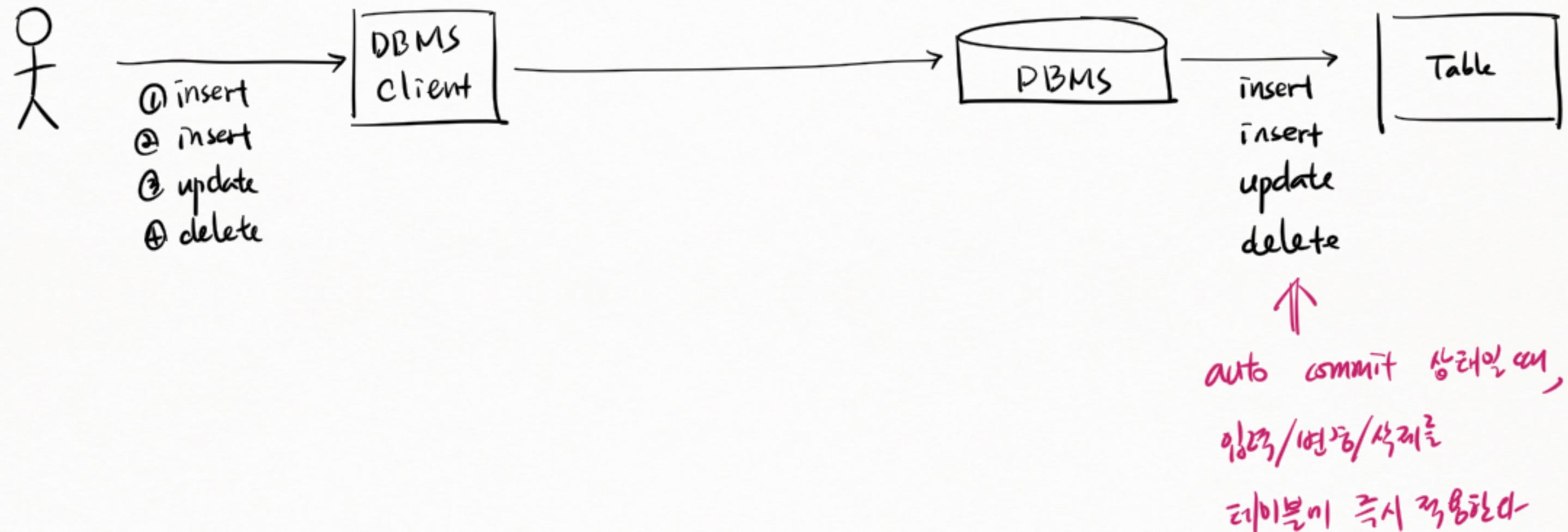
작업 { insert, update, delete }

준비 commit

반복 rollback

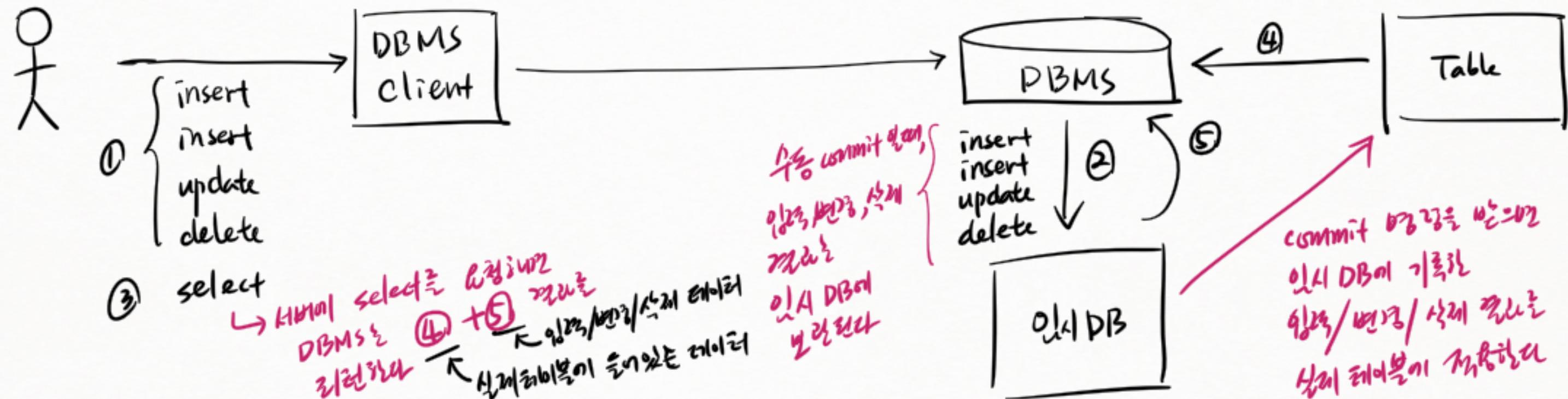
* commit / rollback

① auto commit 상태



* commit / rollback

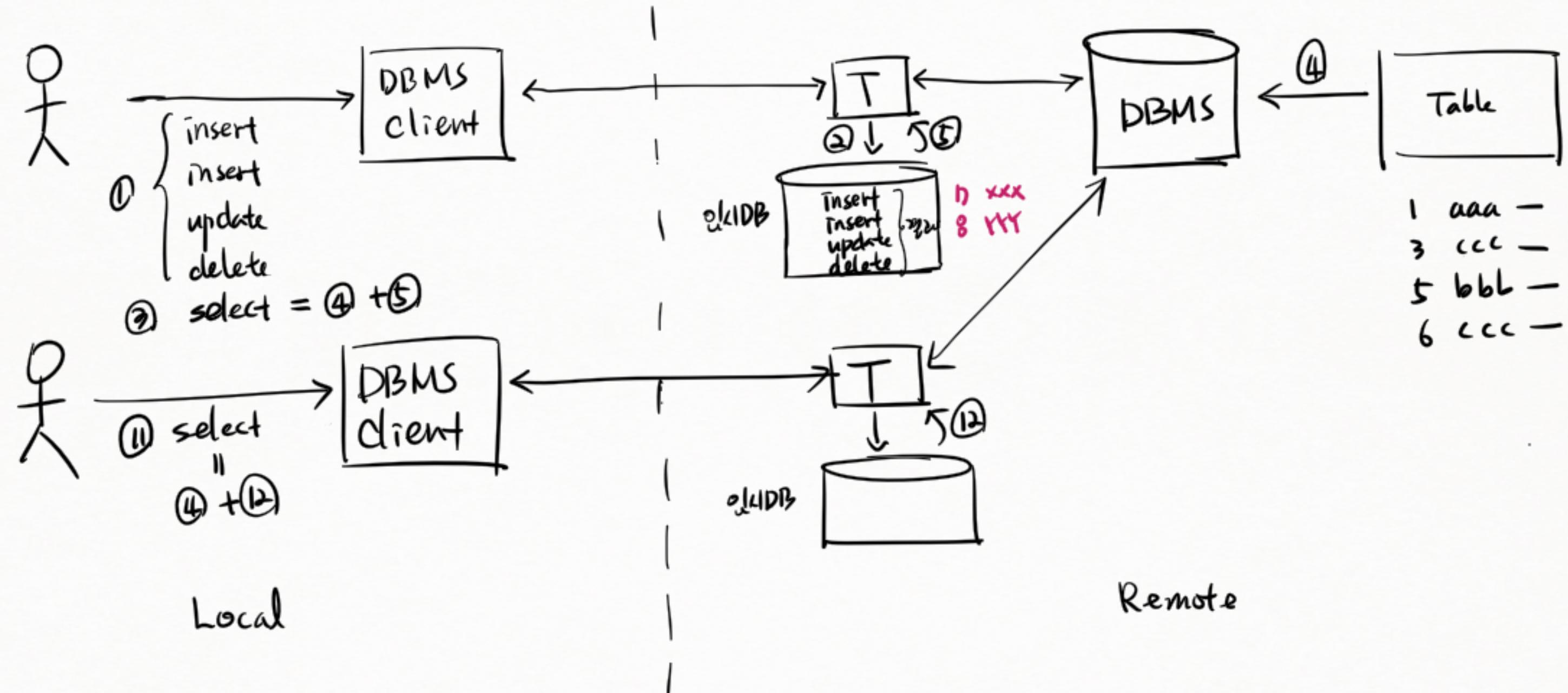
② 수행 commit \Rightarrow set autocommit = false;



예) 1 aaa —
3 ccc —
5 bbb —
6 ccc —

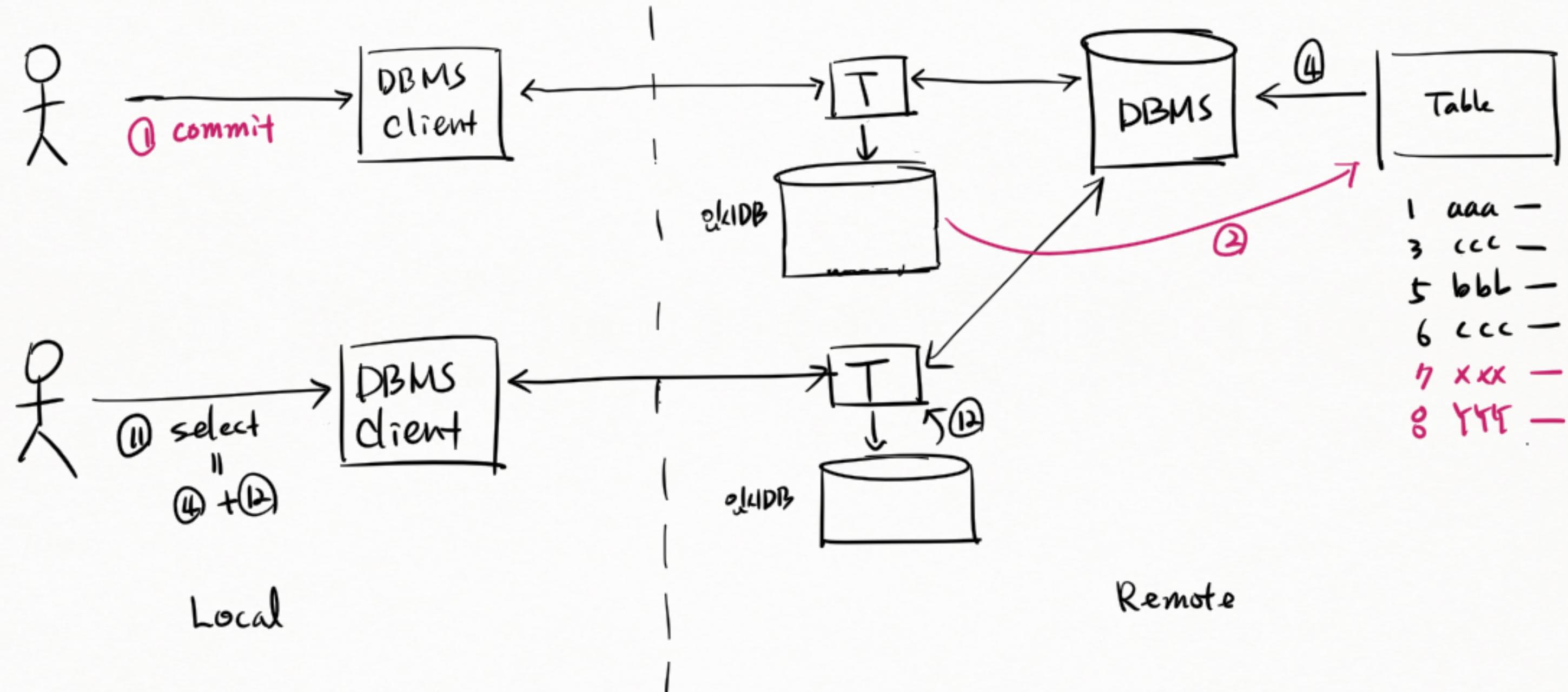
* commit / rollback \Rightarrow 더 자세히!

② 사용 commit \Rightarrow set autocommit = false;



* commit / rollback => 더 자세히!

③ commit → 임시DB에 기록된 결과를 실제 데이터베이스에 적용해줌.



* commit / rollback

④ rollback → olcIDB ni yuxi hig tzu'i xilika.

