제8장 트랜잭션

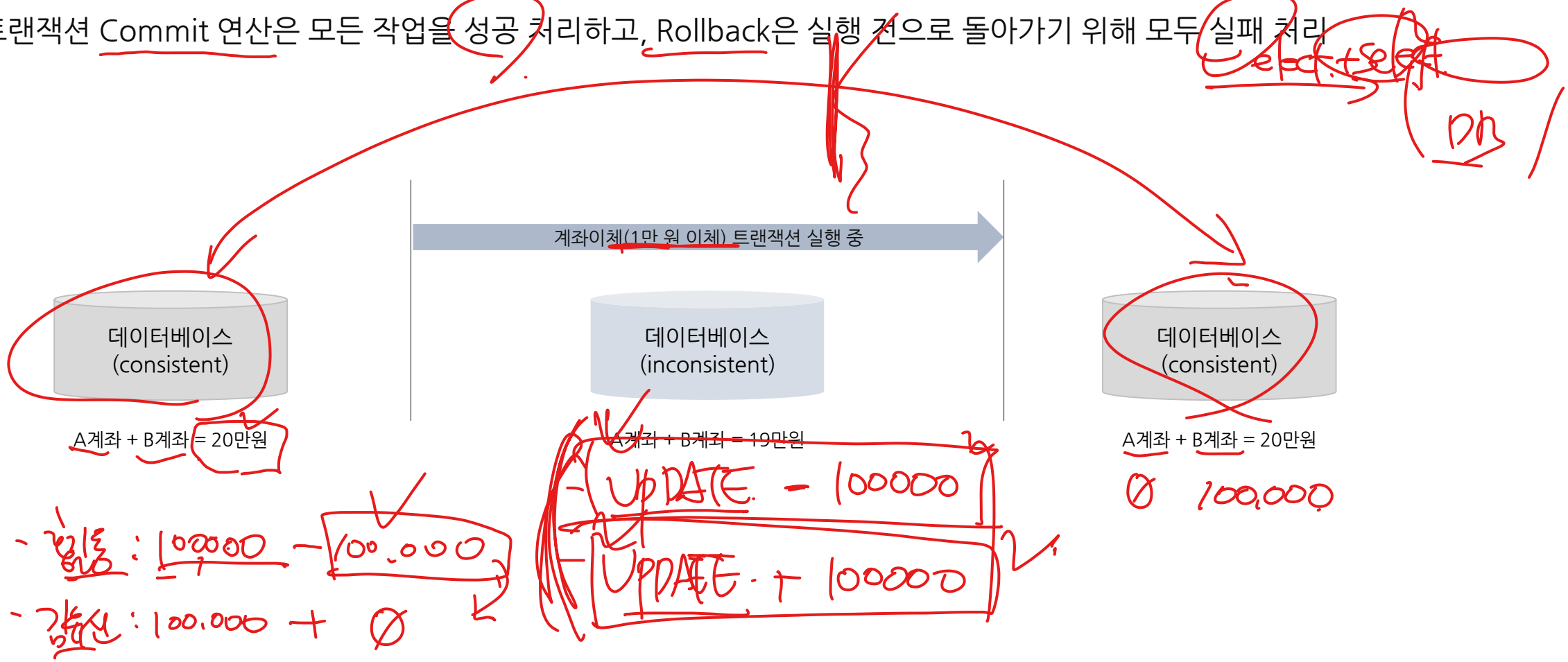
김철학

목차

1. 트랜잭션 개요
2. 트랜잭션 과정
3. 트랜잭션 특성 ACID.
4. 트랜잭션 상태
5. 병행 제어

1. 트랜잭션 개요

- 트랜잭션Transaction 은 하나의 작업을 수행하는데 필요한 데이터베이스 연산들을 모아 놓은 논리적인 작업의 단위
- 데이터베이스의 무결성과 일관성을 보장하기 위해 작업 수행에 필요한 연산들을 하나의 트랜잭션으로 정의하고 관리
- 트랜잭션 Commit 연산은 모든 작업을 성공 처리하고, Rollback은 실행 전으로 돌아가기 위해 모두 실패 처리



2. 트랜잭션 과정

- 트랜잭션은 데이터를 디스크에서 메모리로 가져와 처리한 후 그 결과를 디스크로 저장
- MySQL은 기본적으로 트랜잭션 모드로 처리되지만 다수의 쿼리를 하나의 작업 단위로 처리할 때는 명시적인 트랜잭션을 사용

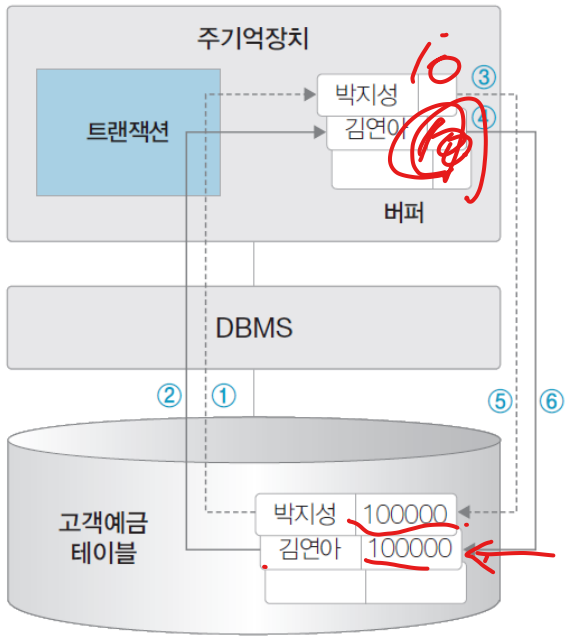
```
START TRANSACTION
① /* 박지성 계좌를 읽어온다 */
② /* 김연아 계좌를 읽어온다 */
   /* 잔고 확인 */

③ /* 예금인출 박지성 */
UPDATE Customer
SET    balance=balance-10000
WHERE  name='박지성';

④ /* 예금입금 김연아 */
UPDATE Customer
SET    balance=balance+10000
WHERE  name='김연아';

COMMIT /* 부분완료 */
⑤ /* 박지성 계좌를 기록한다 */
⑥ /* 김연아 계좌를 기록한다 */
```

(a) 계좌이체 트랜잭션

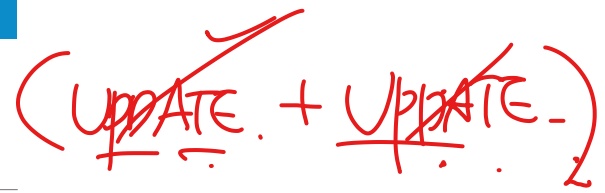


(b) 트랜잭션 수행 과정

3. 트랜잭션 특성

- 트랜잭션은 데이터베이스의 일관성과 무결성을 유지하는 중요한 메커니즘
- 트랜잭션의 ACID 특성을 통해 신뢰성 있는 데이터 처리 보장

특징	설명
원자성Atomicity	• 트랜잭션에 포함된 작업은 전부 수행되거나 아니면 전부 수행되지 않아야 함 • 은행 계좌 이체에서 돈을 보내는 작업과 받는 작업 중 하나라도 실패하면 전체 트랜잭션 실패
일관성Consistency	• 트랜잭션을 수행하기 전이나 수행한 후나 데이터베이스는 항상 일관된 상태를 유지 • 외래 키 제약 조건, 유니크 제약 조건 등이 트랜잭션 실행 전후에 모두 만족
고립성Isolation	• 수행 중인 트랜잭션에 다른 트랜잭션이 끼어들어 변경 중인 데이터 값을 훼손하는 일이 없어야 함 • 동시에 실행되는 트랜잭션들이 서로 간섭하지 않음
지속성Durability	• 수행을 성공적으로 완료한 트랜잭션은 변경한 데이터를 영구히 저장해야 함 • 데이터베이스가 영구적으로 데이터 변경 내용을 유지



4. 트랜잭션 상태

- 부분 완료 상태는 트랜잭션의 마지막 연산이 실행을 끝낸 직후의 상태
- 완료 상태는 트랜잭션이 성공적으로 완료되어 Commit을 수행한 상태로 작업 내용을 데이터베이스에 반영
- 취소 상태는 트랜잭션이 오류로 인해 완료되지 않고 Rollback을 수행한 상태로 모든 변경 사항이 원래 상태로 되돌려진 상태

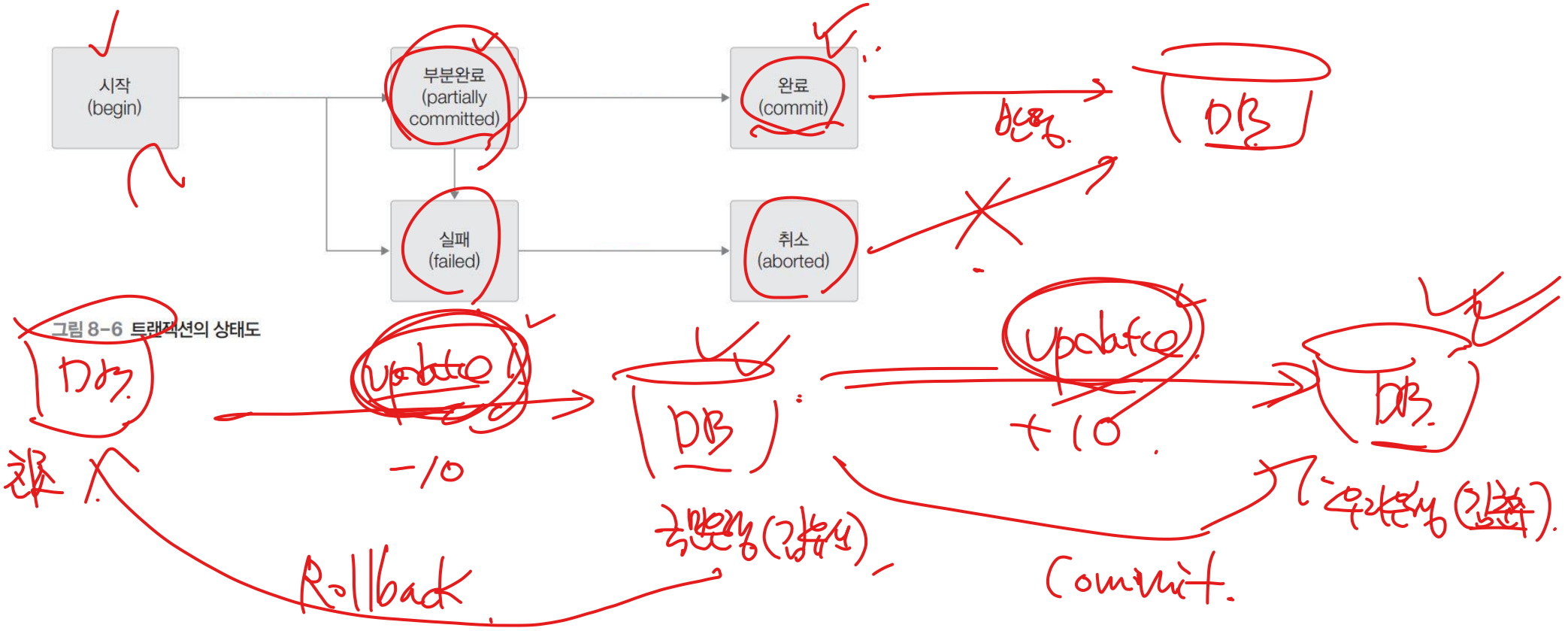
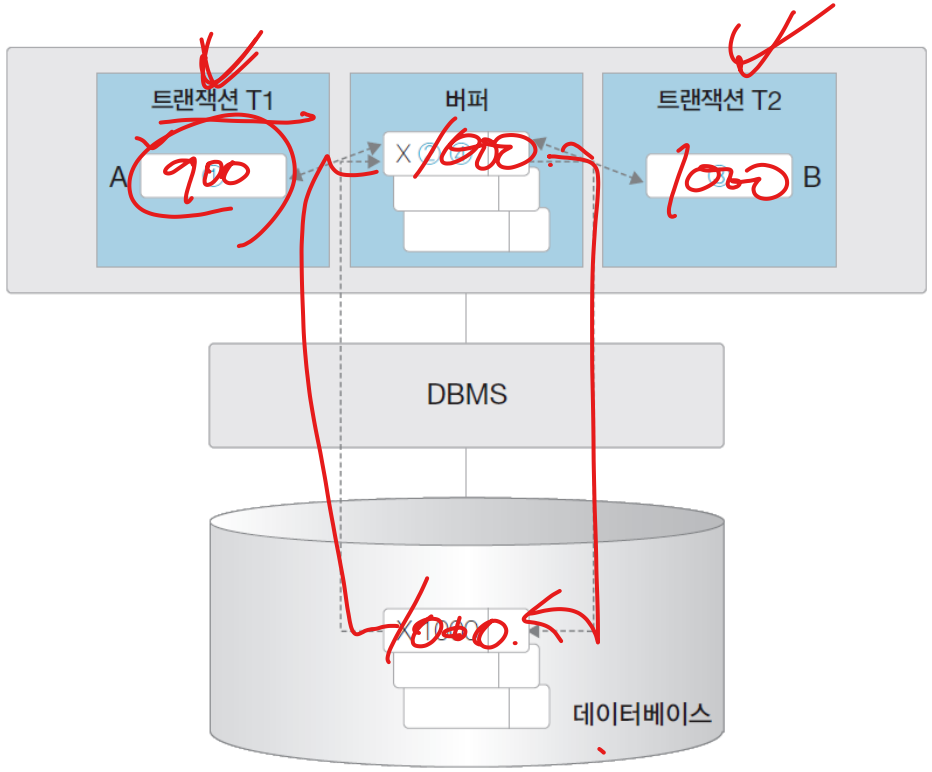


그림 8-6 트랜잭션의 상태도

5. 병행 제어

- 병행 수행은 여러 사용자가 데이터베이스를 동시 공유할 수 있도록 여러 개의 트랜잭션을 동시에 수행하는 것을 의미
- 병행 제어는 병행 수행되는 트랜잭션들이 같은 데이터에 동시에 접근하지 못하도록 Lock과 Unlock을 통해 제어



트랜잭션 T1	트랜잭션 T2	버퍼의 데이터 값
LOCK(X) A=read_item(X); ① A=A-100;		X=1000
	LOCK(X) (wait... 대기)	X=1000
write_item(A→X); ② UNLOCK(X);		X=900
	B=read_item(X); ③ B=B+100; write_item(B→X); ④ UNLOCK(X)	X=1000