

프론트 엔드 개발자가 알아야 하는 컴퓨터 공학 지식

데이터베이스

데이터베이스 | 프론트 엔드 개발자가 알아야 하는 CS 지식

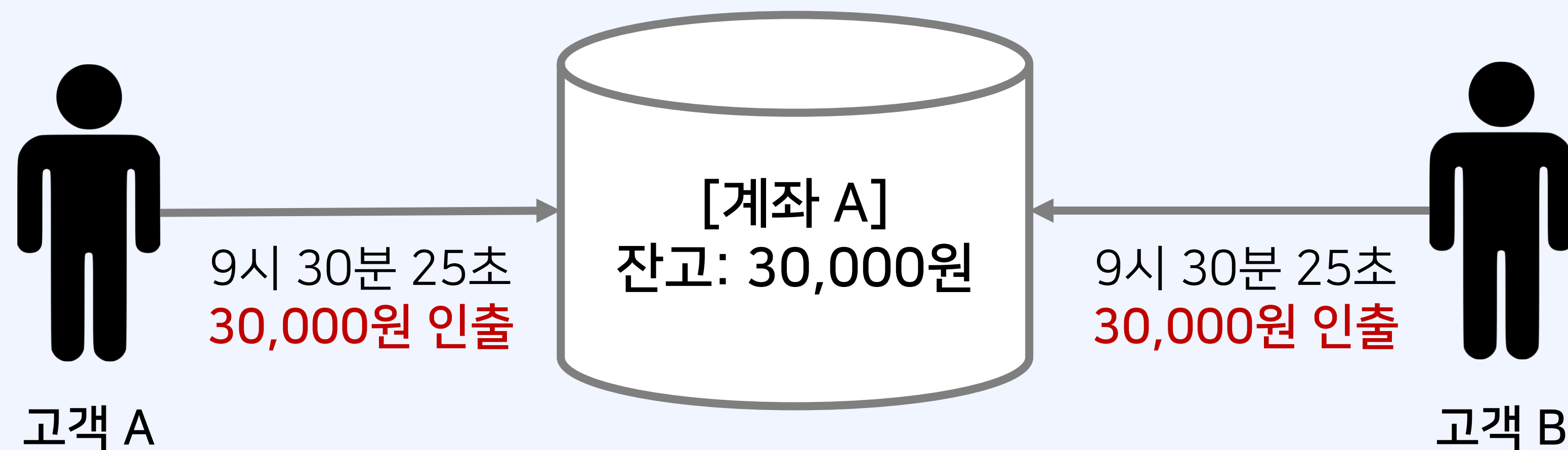
강사 나동빈

프론트 엔드 개발자가 알아야 하는 컴퓨터 공학 지식

데이터베이스

트랜잭션(Transaction)이란?

- **트랜잭션(transaction)**은, DBMS에서 하나의 작업 단위를 의미한다.
- 하나의 계좌에 대해 서로 다른 **두 곳**에서 동시에 돈을 인출한다면 어떻게 될까?
- 두 트랜잭션을 모두 거부하거나, 더 빠르게 호출된 요청을 처리하는 것이 합당할 것이다.



트랜잭션(Transaction)이란?

- 따라서 DBMS의 트랜잭션(transaction)은 네 가지 특성을 만족해야 한다.
 - 1) 원자성(Atomicity): 트랜잭션에 포함된 연산들을 모두 처리하거나, 모두 처리하지 않아야 한다.
 - 2) 일관성(Consistency): 트랜잭션이 수행된 이후에도 DB 내의 데이터는 **일관된 상태를 유지한다.**
 - 3) 격리성(Isolation): 하나의 트랜잭션이 끝날 때까지 다른 트랜잭션이 간섭할 수 없다. (독립적)
 - 4) 지속성(Durability): 트랜잭션이 성공적으로 완료된 이후에 데이터들이 영구적으로 보존된다.
- 이를 ACID 특성이라고 부르기도 한다.

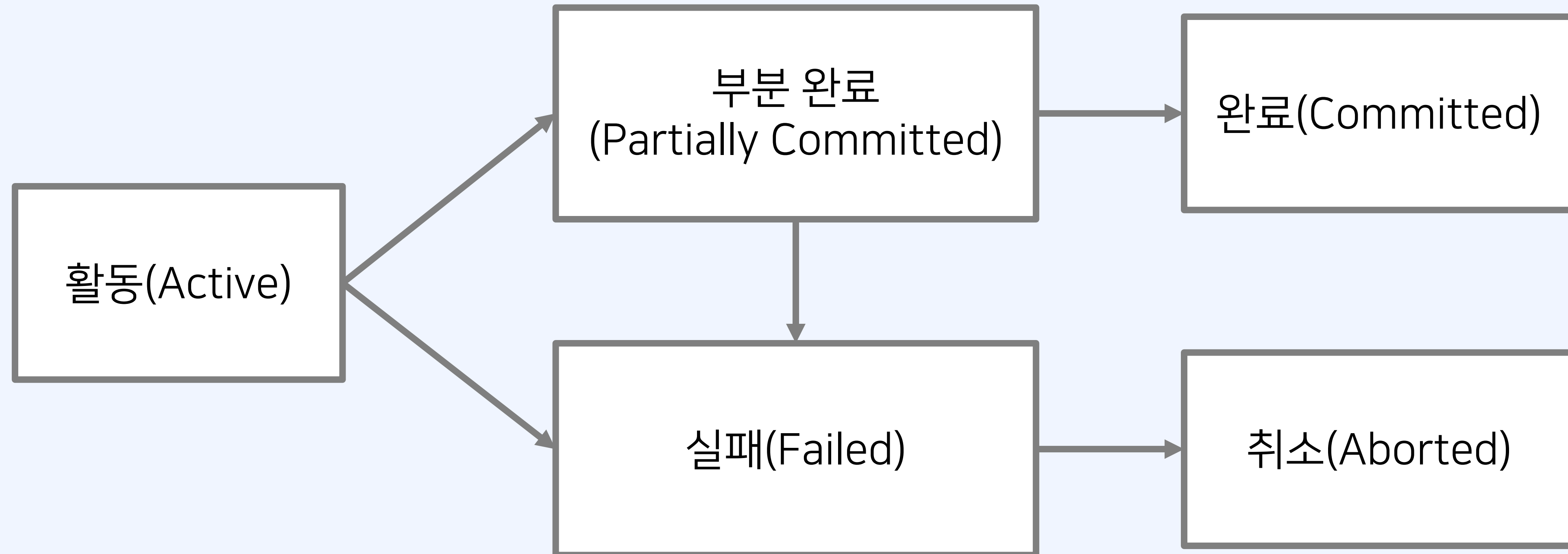
트랜잭션의 Commit과 Rollback

- **Commit:** 하나의 트랜잭션이 성공적으로 수행되었음을 명시하기 위해 사용한다.
 - 하나의 트랜잭션이 성공적으로 끝났다면, 데이터베이스가 일관성 있는 상태를 유지하게 된다.
- **Rollback:** 트랜잭션이 비정상적으로 종료되어, DB의 일관성을 유지하기 어려울 때 사용한다.
 - 원자성을 유지하기 위하여 해당 트랜잭션에 포함된 모든 연산을 취소한다.
 - 해당 트랜잭션의 일부가 정상적으로 처리되었더라도 모든 연산을 이전으로 되돌린다.

트랜잭션(Transaction)의 상태 전이

- **활동(active)**: 트랜잭션이 실행 중이며, 동작 중인 상태이다.
- **부분 완료(partially committed)**: 트랜잭션이 마지막 명령을 실행한 직후의 상태이다.
→ commit 요청이 들어오면, 부분 완료 상태가 된다. 문제가 없는 경우 완료 상태로 전환된다.
- **완료(committed)**: 트랜잭션이 정상적으로 완료되어 종료된 상태이다.
- **실패(failed)**: 트랜잭션이 더 이상 정상적으로 진행될 수 없어, 실패한 상태이다.
- **취소(aborted)**: 트랜잭션이 취소되어 트랜잭션 실행 이전 데이터로 돌아간 상태이다. (Rollback)

트랜잭션(Transaction)의 상태 전이



트랜잭션 사용의 유의사항

- 따라서, 필요한 작은 단위의 코드 위주로 트랜잭션을 사용하는 것을 권장한다.
 - 기본적으로 DBMS 입장에서 사용자 연결(connection)의 수가 많으면 서버 부담이 커질 수 있다.
 - 트랜잭션을 처리하기 위한 커넥션 유지 기간이 길어지면, 전반적으로 각 프로그램의 대기 시간이 과도하게 길어질 수 있다.