

Transaction : DB

01 Transaction

Transaction

DB의 상태를 변화시키기 위해 수행하는 작업단위
혹은 한꺼번에 모두 수행되어야 할 일련의 연산

상태를 변화시키다 = select, insert, delete, update 와 같은 질의어를 통해 접근하는 것
작업 단위 = 많은 SQL문들을 사람이 정하는 기준에 따라 정하는 것

01 Transaction

특징

원자성(Atomicity)

- 트랜잭션이 DB에 모두 반영되거나 전혀 반영되지 않아야 한다.
- 트랜잭션 내의 모든 명령은 반드시 완벽하게 수행되어야 하며 어떤 것 하나라도 수행이 되지 않는다면 트랜잭션 모두 취소되어야 한다.

주요 기법: commit, rollback

01 Transaction

특징

일관성(Consistency)

- 트랜잭션의 작업 처리 결과는 항상 일관성이 있어야 한다.
- 시스템이 가지고 있는 고정 요소는 트랜잭션 수행 전과 트랜잭션 수행 완료 후의 상태가 같아야 한다.

주요 기법: 무결성 제약 조건, 동시성 제어

01 Transaction

특징

일관성(Consistency)

- 트랜잭션의 작업 처리 결과는 항상 일관성이 있어야 한다.
- 시스템이 가지고 있는 고정 요소는 트랜잭션 수행 전과 트랜잭션 수행 완료 후의 상태가 같아야 한다.

주요 기법: 무결성 제약 조건, 동시성 제어

동시성 제어: DBMS가 다수의 사용자 사이에서 동시에 작용하는 다중 트랜잭션의 상호간섭 작용에서 Database를 보호하는 것을 의미

01 Transaction

특징

독립성(Isolation)

- 둘 이상의 트랜잭션이 동시에 병행 실행되고 있을 때 어떤 트랜잭션도 다른 트랜잭션 연산에 끼어 들 수 없다.

주요 기법: read uncommitted, read committed, repeatable read, serializable

01 Transaction

특징

독립성(Isolation)

- 둘 이상의 트랜잭션이 동시에 병행 실행되고 있을 때 어떤 트랜잭션도 다른 트랜잭션 연산에 끼어 들 수 없다.

주요 기법: read uncommitted, read committed, repeatable read, serializable

트랜잭션 격리 수준

동시에 여러 트랜잭션이 처리 될 때 특정 트랜잭션이 다른 트랜잭션에서 변경하거나 조회하는 데이터를 볼 수 있도록 허용할지 말지 결정하는 것

01 Transaction

특징

지속성(Durability)

- 트랜잭션이 성공적으로 완료 되었으면 결과는 영구적으로 반영되어야 한다.

주요 기법: 회복 기법

01 Transaction

특징

지속성(Durability)

- 트랜잭션이 성공적으로 완료 되었으면 결과는 영구적으로 반영되어야 한다.

주요 기법: 회복 기법

회복

데이터베이스를 장애가 발생했던 이전의 상태로 복구시켜서 일관된 데이터베이스 상태를 만드는 것

01 Transaction

TCL

트랜잭션 제어 언어 (Transaction Control Language)

트랜잭션의 결과를 허용하거나 취소하는 목적으로 사용되는 언어를 지칭한다.

- commit : 트랜잭션 확정, 트랜잭션을 메모리에 영구적으로 저장하는 명령어
- rollback : 트랜잭션 취소, 트랜잭션 내역을 저장 무료화 시키는 명령어
- checkpoint : 저장 시기 설정, rollback을 위한 시점을 저장하는 명령어

01 Transaction

상태 변화

활동 상태

: 초기 상태, 트랜잭션이 실행 중일 때 가지는 상태

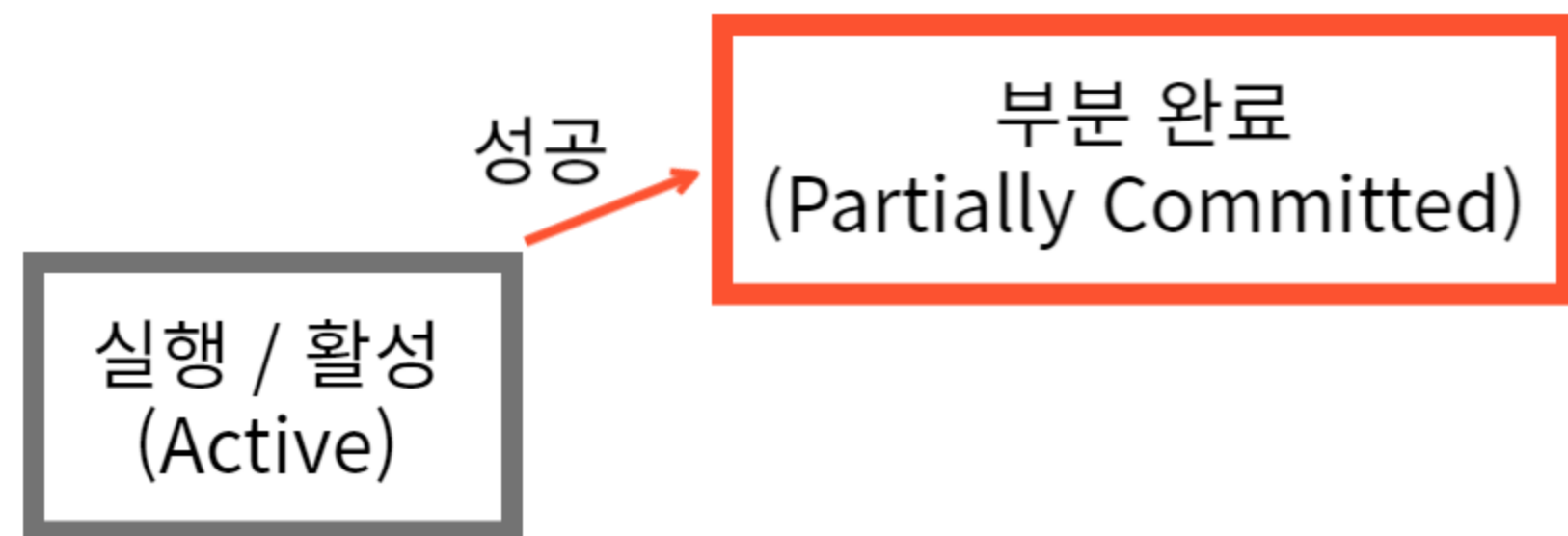
실행 / 활성
(Active)

01 Transaction

상태 변화

부분 완료

: DML 등 트랜잭션의 명령을 실행한 후의 상태

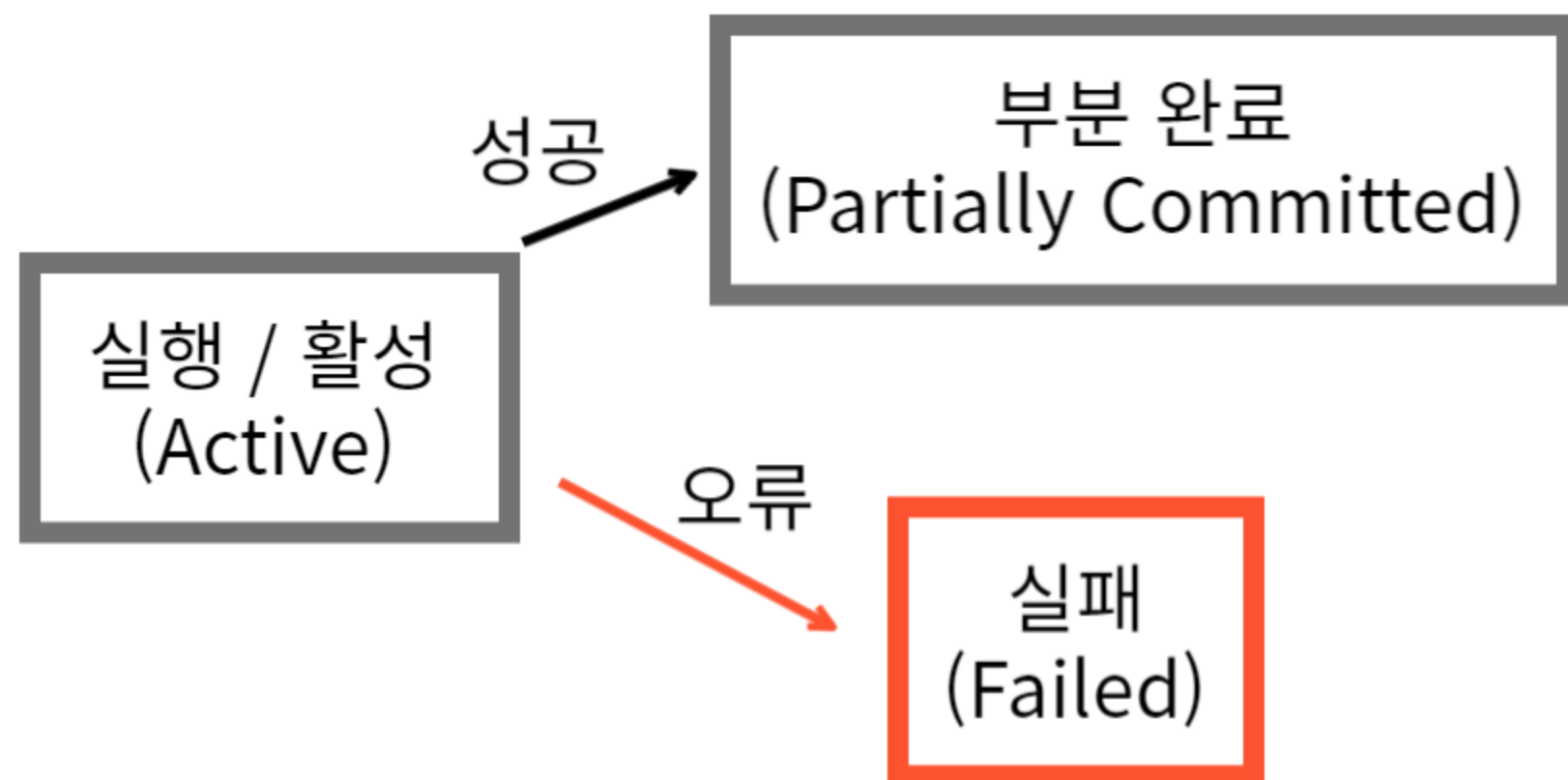


01 Transaction

상태 변화

실패

: 더이상 정상적으로 실행 될 수 없음을 발견



01 Transaction

상태 변화

완료

: 트랜잭션이 성공적으로 완료

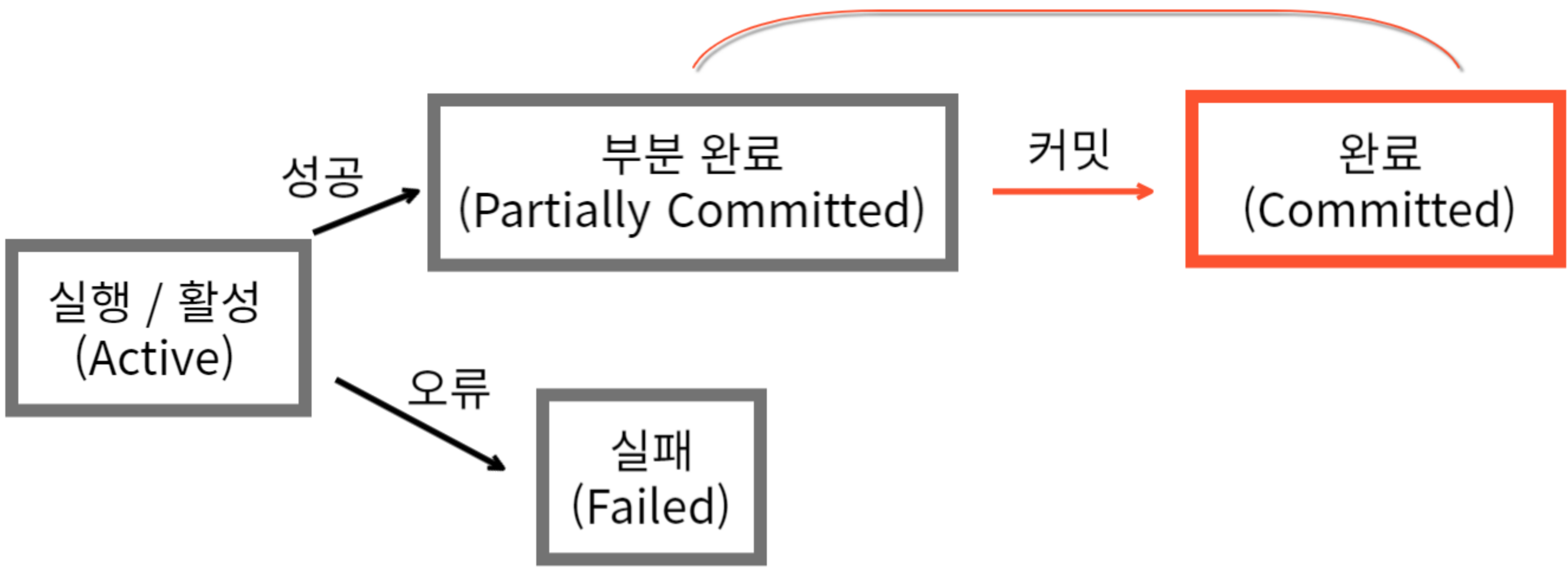


01 Transaction

상태 변화

완료
: 트랜잭션이 성공적으로 완료

부분완료와 완료의 차이점
커밋 요청이 들어오면 상태는 부분완료
이후 커밋을 문제없이 수행할 수 있다면 완전 완료

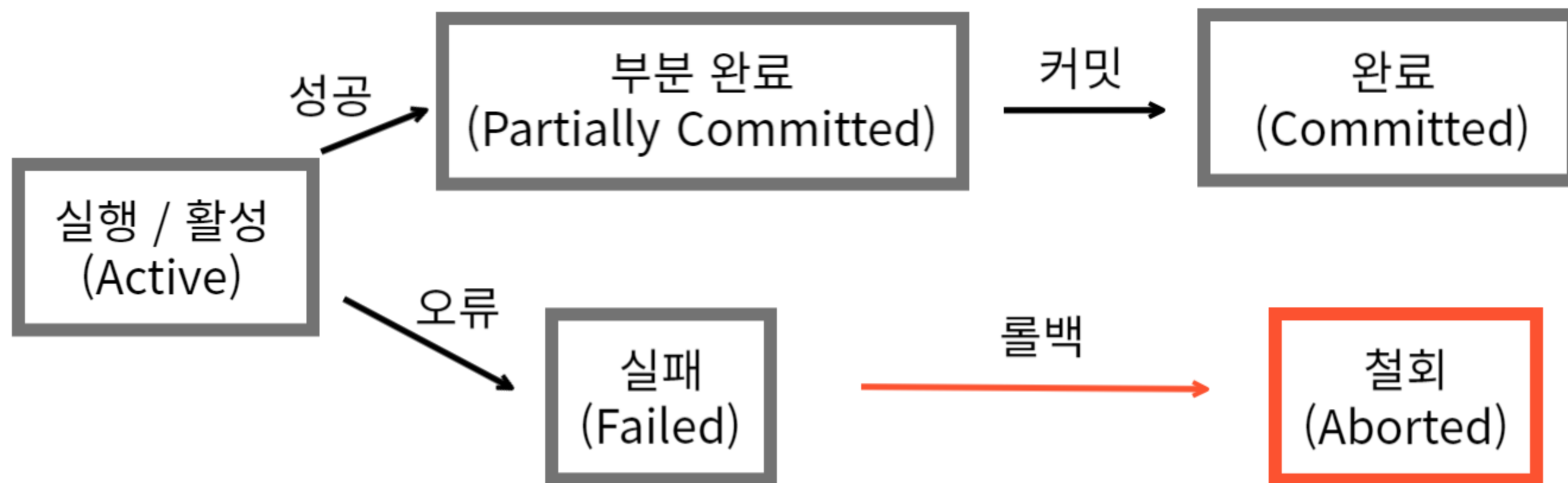


01 Transaction

상태 변화

철회

: 트랜잭션이 복원되어 트랜잭션이 수행 이전 상태로 돌아감



01 Transaction

병행제어 기법

병행제어

: 다수의 사용자 환경에서 여러 트랜잭션을 수행할 때
데이터베이스 일관성 유지를 위해 상호작용을 제어하는 기법

목적

- 데이터베이스 공유 최대화
- 시스템 활용도 최대화
- 데이터베이스 일관성 유지
- 사용자에게 대한 응답시간 최소화

01 Transaction

병행제어 기법

미보장시 문제

- 갱신 손실: 먼저 실행된 트랜잭션의 결과를 나중에 실행된 트랜잭션이 덮어쓸 때 발생하는 오류
- 현황 파악오류: 트랜잭션의 중간 수행 결과를 다른 트랜잭션이 참조하여 발생하는 오류
- 모순성: 두 트랜잭션이 동시에 수행되어 데이터베이스의 일관성이 결여되는 오류
- 연쇄복귀: 복수의 트랜잭션이 데이터 공유 시 특정 트랜잭션이 처리를 취소할 경우 트랜잭션이 처리한 곳의 부분을 취소하지 못하는 오류

02 출처

<https://gyoogle.dev/blog/computer-science/data-base/Transaction.html>
<https://coding-factory.tistory.com/226>
<https://velog.io/@jmlee9707/DB-DB-%ED%8A%B8%EB%9E%9C%EC%9E%AD%EC%85%98transaction>
<https://code-lab1.tistory.com/51>
<https://thebook.io/006977/0277/>