

DB

5 트랜잭션 개념

주요 질문

5.

트랜잭션 개념

- ✓ 트랜잭션의 정의에 대해서 설명해 보세요. (트랜잭션이란 무엇일까요?)
- ✓ ACID 에 대해 설명해 보세요.
- ✓ SQL-92 에서 표준으로 만든 Isolation Level 에는 어떤 것들 있을까요?
- ✓ 데이터 동시성 제어를 하지 않을 때 발생하는 문제 – Phantom Read 에 대해서 설명해 보세요.

트랜잭션(Transaction) 개념

5.

트랜잭션 개념

- 하나의 논리적 기능을 수행하기 위한 작업 단위
- 한 번에 수행되어야 할 DB의 일련의 Read와 Write 연산을 수행하는 단위



ACID

5.

트랜잭션 개념

ACID 특징

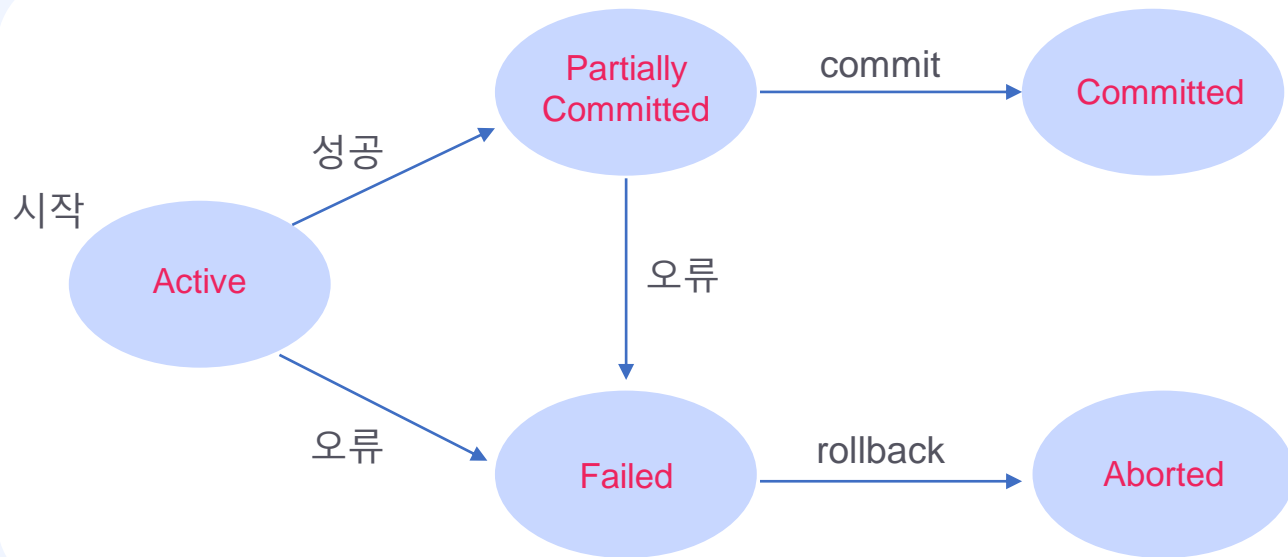
특징	기본 설명	관리주체
A tomicity (원자성)	<ul style="list-style-type: none"> 한 개 이상의 동작이 논리적으로 한 개의 작업 단위 (single unit of work) 연산 전체가 성공적으로 처리되거나 실패할 경우 전체가 취소되도록 보장(All or Nothing) 	트랜잭션 관리자
C onsistency (일관성)	<ul style="list-style-type: none"> 트랜잭션이 실행 성공 후 항상 모순 없이 일관성 있는 DB 상태 보존 	무결성 제어기
I solation (고립성)	<ul style="list-style-type: none"> 트랜잭션 실행 중 생성하는 연산의 중간결과를 다른 트랜잭션 접근 불가 	병행제어 관리자
D urability (영속성)	<ul style="list-style-type: none"> 성공이 완료된 트랜잭션의 결과는 영구(속)적으로 데이터베이스에 저장됨 	연관회복 관리자

트랜잭션 상태도

5.

트랜잭션 개념

트랜잭션 상태도



트랜잭션 상태도

5.

트랜잭션 개념

상태	기본 설명	전이
활동 (Active)	초기 상태, 트랜잭션이 실행을 시작하였거나 실행 중인 상태	Partially Committed, Failed
부분완료 (Partially Committed)	마지막 명령문이 실행된 후 Commit 직전 상태 Commit 명령이 도착한 상태	Failed, Commit
실패 (Failed)	정상적인 실행이 더 이상 진행될 수 없는 상태	Aborted
철회 (Aborted)	트랜잭션 실행이 실패하여 취소되고 트랜잭션 시작전 상태로 환원된 상태 (Rollback)	종료
완료 (Committed)	성공이 완료된 트랜잭션의 결과는 영구(속)적으로 데이터베이스에 저장됨	종료

동시성 제어

- 멀티 프로세스 환경을 지원하는 데이터 베이스 시스템에서 여러 트랜잭션들이 성공적으로 동시에 실행될 수 있도록 지원하는 기능
- DBMS가 다수의 사용자 사이에서 동시에 작용하는 다중 트랜잭션의 상호간섭에서 Database를 보호하는 것을 의미



동시성 제어 하지 않을 때 문제점

5.

트랜잭션 개념

구분	상세 설명
갱신 손실 (Lost Update)	<ul style="list-style-type: none"> 트랜잭션들이 동일 데이터를 동시에 갱신 할 경우 발생 이전 트랜잭션이 데이터를 갱신한 후 트랜잭션을 종료하기 전에 나중 트랜잭션이 갱신 값을 덮어쓰는 경우 발생
현황파악오류 (Dirty Read)	<ul style="list-style-type: none"> 트랜잭션의 중간 수행결과를 다른 트랜잭션이 참조함으로써 발생하는 오류
모순성 (Inconsistency)	<ul style="list-style-type: none"> 두 트랜잭션이 동시에 실행할 때 DB가 일관성이 없는 상태로 남는 문제
연쇄복귀 (Cascading Rollback)	<ul style="list-style-type: none"> 복수의 트랜잭션이 데이터 공유 시 특정 트랜잭션이 처리를 취소할 경우 다른 트랜잭션이 처리한 부분에 대해 취소 불가능
반복할 수 없는 읽기 (Unrepeatable Read)	<ul style="list-style-type: none"> 한 트랜잭션 내에서 같은 질의를 2번 수행했을 때, 그 사이 다른 트랜잭션의 간섭으로 결과가 다르게 나오는 오류

동시성 제어 목적과 기법

5.

트랜잭션 개념

동시성 제어 목적

- 트랜잭션의 직렬성 보장, 동시 수행 트랜잭션 처리량 최대화
- 데이터의 무결성 및 일관성 보장
- 공유도 최대, 응답 시간 최소, 시스템 활동의 최대 보장

동시성 제어 기법들

- 2-Phase Locking 기법
- Time Stamp Ordering 기법
- 낙관전 검증(Validation) 기법
- 다중버전 병행제어 기법- MVCC(Multi-Version Concurrency Control)

Isolation Level

- 트랜잭션 실행 중 중간 연산 결과가 다른 트랜잭션으로 접근 불가하도록 하는 고립성을 유지하기 위한 데이터를 허용하는 수준
- SQL-92표준에서 4단계로 표준화, DBMS 벤더들이 준수

레벨	상태	기본 설명	전이
0	Read Uncommitted	트랜잭션에서 처리중인, 아직 COMMIT 되지 않은 데이터를 다른 트랜잭션이 읽는 것을 허용	<ul style="list-style-type: none"> Dirty Read Nonrepeatable Read Phantom Read
1	Read Committed	트랜잭션이 COMMIT 되어 확정된 데이터만 읽는 것을 허용 (해당 데이터만 shared lock)	<ul style="list-style-type: none"> Nonrepeatable Read Phantom Read
2	Repeatable Read	선행 트랜잭션이 읽은 데이터는 트랜잭션이 종료될 때까지 후행 트랜잭션이 갱신, 삭제하는 것 불허	<ul style="list-style-type: none"> Phantom Read
3	Serializable	선행 트랜잭션이 읽은 데이터를 후행 트랜잭션이 갱신하거나 삭제하지 못할 뿐 아니라, 중간에 새로운 레코드를 삽입도 금지	모두 발생 불가 Select 테이블 전체 Lock

Isolation Level 문제점

5.

트랜잭션 개념

Isolation Level 따른 트랜잭션 수행 중에 문제점이 발생 가능

현상	설명
Dirty Read	A 트랜잭션 : 101번 사원 나이를 27 -> 28 update (커밋전) B 트랜잭션: 101번 사원 select (28) A 트랜잭션: 문제로 Rollback
Non-repeatable Read	B 트랜잭션: 101번 계좌 잔고 select (결과 : 3000원) A 트랜잭션 : 101번 계좌 이체 나이를 3000 -> 0 update (커밋 완료) B 트랜잭션 : 101번 계좌 select (결과 : 0)
Phantom Read	A 트랜잭션 : 사원 전체 count(*) -> 1200명 B 트랜잭션 : 신입사원 insert A 트랜잭션 : 사원 전체 cout(*) -> 1201명

Isolation Level 요약

5.

트랜잭션 개념

Isolation Level	Dirty Read	Non-Repeatable Read	Phantom Read
Read Uncommitted	O	O	O
Read committed	-	O	O
Repeatable Read	-	-	O
Serializable	-	-	-

Isolation Level 명령어

5.

트랜잭션 개념

show variables like 'tx_isolation'

```
MariaDB [bootex]> show variables like 'tx_isolation';
+-----+-----+
| Variable_name | Value               |
+-----+-----+
| tx_isolation  | READ-UNCOMMITTED   |
+-----+-----+
1 row in set (0.001 sec)
```

set tx_isolation = 'READ-COMMITTED';