

제5장 데이터베이스 회복

1. 트랜잭션(transaction)

- 트랜잭션은 데이터베이스를 대상으로 논리적인 기능을 수행하는 작업 단위이다.
- 트랜잭션 실행 도중에 장애가 발생하면, 데이터를 회복하는 작업 단위가 된다.
- 즉, 트랜잭션은 한꺼번에 수행되어야 할 일련의 데이터베이스 연산 집합이다.

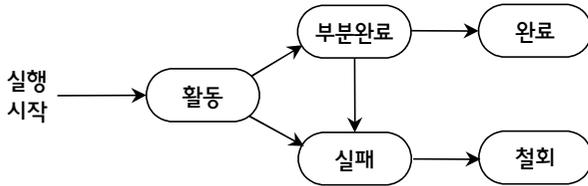
// 계좌 A에서 계좌 B로 50원을 이체하는 트랜잭션

```
-----
Begin_Trans
  Read A
  A = A - 50
  Write A
  Read B
  B = B + 50
  Write B
End_Trans
-----
```

// 트랜잭션 특징(ACID 성질)

원자성 (atomicity)	<ul style="list-style-type: none"> • 트랜잭션은 자신의 연산을 전부 또는 전부 실행한다.(all or nothing) • 트랜잭션은 일부 실행으로 트랜잭션 기능을 갖는 것은 아니다. • 트랜잭션 실행 도중에 실패하게 되면, 부분적인 일부 실행 결과는 시스템 내에서 제거해야 한다. • 원자성은 가장 중요한 트랜잭션 성질이다.(논리적 작업 단위이므로, <u>이석호교수</u>)
일관성 (consistency)	<ul style="list-style-type: none"> • 트랜잭션 실행이 성공적으로 완료되면, 데이터베이스는 일관된 또 다른 상태로 변하게 된다. • 즉, 트랜잭션 실행 결과로 데이터베이스는 모순되지 않는다.
격리성 (isolation)	<ul style="list-style-type: none"> • 트랜잭션 실행 중에 있는 연산 중간 결과는 다른 트랜잭션이 접근할 수 없다. • 격리성은 독립성, 고립성이라고도 한다. • 즉, 트랜잭션 실행은 서로 영향을 끼치지 않는다. • DBMS는 격리성 유지를 위해 동시성제어 알고리즘을 작동시킨다.
영속성 (durability)	<ul style="list-style-type: none"> • 트랜잭션 실행이 성공적으로 끝나면 그 결과는 영속적이다.(지속성) • 즉, 시스템은 완료된 트랜잭션의 결과를 어떤 경우에도 보장해야 한다. • 정상적으로 완료된 트랜잭션에 의한 모든 변경은 향후에 어떤 소프트웨어나 하드웨어 장애가 발생되더라도 보존되어야 한다.

// 트랜잭션의 5가지 상태



활동 active	<ul style="list-style-type: none"> 트랜잭션이 실행을 시작하였거나 실행 중에 있는 상태를 나타낸다. 트랜잭션의 실행은 BEGIN_TRANS로부터 시작된다. 활동 상태에서 직접 전이될 수 있는 상태는 '부분완료나 실패' 상태이다.
부분완료 partially committed	<ul style="list-style-type: none"> 트랜잭션을 구성하는 명령문 중 마지막 명령문을 실행한 직후의 상태이다. 부분완료는 트랜잭션 실행은 이 시점에서 끝났지만 변경된 메모리 버퍼 블록이 아직 디스크의 데이터베이스에 출력되지 않은 상태이다. 부분완료는 메모리 버퍼 블록에 트랜잭션 실행 결과가 보관되어 있을 뿐이다. 디스크로 출력하는 동안 하드웨어 장애가 발생하면 트랜잭션은 실패가 된다. 부분완료의 트랜잭션은 실패하지 않는다는 것을 보장받을 때 완료로 들어간다.
실패 failed	<ul style="list-style-type: none"> 트랜잭션이 정상적으로 더 이상 실행할 수 없는 상태가 되어 중단된 상태이다. 트랜잭션 실행 오류가 발생한 상태이다. 실패 상태에서 전이될 수 있는 다음 상태는 '철회' 상태이다.
철회 aborted	<ul style="list-style-type: none"> 철회 상태는 트랜잭션 실행이 실패로 끝난 상태이다. 철회 상태에서는 복귀(rollback) 연산을 실행한다.(복귀 연산을 실행한 상태)
완료 committed	<ul style="list-style-type: none"> 트랜잭션이 실행을 성공적으로 완료하여 commit 연산이 실행된 상태이다. 완료는 데이터베이스에 적용한 트랜잭션의 연산들이 안전하게 수행된 것이다. 데이터베이스 상태는 일관성 있는 상태로 변환되었다. 취소(undo) 되지 않는다. 갱신된 데이터 값들이 영속성을 가지도록 시스템이 보장하게 된다.

• 완료(commit)와 복귀(rollback)는 트랜잭션의 원자성을 위한 연산이다.

// 완료(commit) 연산 - 트랜잭션 실행이 성공적으로 종료되었음을 선언

- 트랜잭션 실행이 성공적으로 종료된 것이다.(데이터베이스는 일관된 상태 유지)
- 트랜잭션 실행이 완료된 것을 트랜잭션 관리자에게 알리는 연산이다.
- 트랜잭션의 연산 결과가 메모리 버퍼 블록에서 디스크의 데이터베이스에 저장된다.(취소 불가)

// 복귀(rollback) 연산 - 트랜잭션 실행이 실패되었음을 선언

- 복귀 연산은 트랜잭션 실행 도중에 장애가 발생한 경우에 수행된다.
- 장애가 발생하면, 데이터베이스는 모순된 상태가 될 수 있다.(트랜잭션 일부만 반영되어)
- 트랜잭션이 작업한 모든 갱신 연산을 복귀 또는 취소하도록 DBMS에게 알린다.
- DBMS는 해당 트랜잭션이 실행되기 이전의 상태로 데이터베이스를 복원한다.
- 트랜잭션 실행을 모두 취소하거나 트랜잭션을 다시 시작해야 한다.



탐구

트랜잭션 시작과 끝

먼저, SQL에서 트랜잭션의 명시적인 시작문은 없다.(생략 가능하다는 것!)

즉, SQL에서 명시적인 Begin Transaction문이 없다.

단지, SQL 문장을 만났을 때 트랜잭션이 묵시적으로 시작한다.

하지만, 모든 트랜잭션은 명시적으로 종료문을 가져야 한다.

트랜잭션의 명시적인 종료는 Commit 또는 Rollback문을 사용한다.

Commit : 트랜잭션에서 변경하려는 내용을 데이터베이스에 완전하게 반영되는 것

SQL 구문 : Commit Work

Commit Transaction과 Commit Work는 성공적인 트랜잭션의 실행 종료를 의미한다.

Commit Transaction과 Commit Work는 같은 기능을 가진다.

Commit Work에서 Work는 생략할 수 있다.

Abort : 트랜잭션이 갱신한 것을 트랜잭션 수행되기 전의 상태로 되돌리는 것

SQL 구문 : Rollback Work

Rollback Transaction과 Rollback Work는 비정상적인 트랜잭션의 종료를 의미한다.

Rollback Transaction과 Rollback Work는 같은 기능을 가진다.

[예] 트랜잭션 시작과 끝

Update 수강

Set 점수 = 점수 + 5

Where 학번 = 100 And 과목 = '디비';

Commit;

- 위의 예는 트랜잭션 시작은 명시적으로 나타내지 않고 있다.
- 일반적으로, 트랜잭션 시작은 기술하지 않는 것이 관례이다.
- 위의 예는 하나의 Update 문으로 이루어진 트랜잭션을 나타낸다.
- 마지막에 Commit을 사용하여 트랜잭션의 끝을 표시하고 있다.

// 트랜잭션 종류

<p>명시적 트랜잭션</p>	<p>Begin Tran //트랜잭션 시작 Update 학생 Set 주소 = '서울' Where 학번 = 1; Update 학생 Set 주소 = '평양' Where 학번 = 2; Commit 또는 Rollback Tran //트랜잭션 끝</p> <hr/> <ul style="list-style-type: none"> • 사용자가 트랜잭션 시작과 끝을 직접 명시하는 트랜잭션이다. • Begin Transaction 문으로 시작되어 Commit 또는 Rollback 문으로 끝난다. • 명시적 트랜잭션은 '사용자 트랜잭션' 또는 '수동 트랜잭션'이라 한다.
<p>자동완료 트랜잭션</p>	<p>Update 학생 Set 주소 = '서울' Where 학번 = 1; Update 학생 Set 주소 = '평양' Where 학번 = 2; Update 학생 Set 주소 = '제주' Where 학번 = 3; ↓ 다음과 같다.</p> <p>Begin Tran //트랜잭션 시작 Update 학생 Set 주소 = '서울' Where 학번 = 1; Commit 또는 Rollback Tran //트랜잭션 끝</p> <p>Begin Tran //트랜잭션 시작 Update 학생 Set 주소 = '평양' Where 학번 = 2; Commit 또는 Rollback Tran //트랜잭션 끝</p> <p>Begin Tran //트랜잭션 시작 Update 학생 Set 주소 = '제주' Where 학번 = 3; Commit 또는 Rollback Tran //트랜잭션 끝</p> <hr/> <ul style="list-style-type: none"> • 트랜잭션 시작과 끝을 시스템이 자동으로 붙이는 트랜잭션이다. • 각 쿼리마다 자동으로 트랜잭션 시작과 끝을 붙인다. • 보통 DBMS의 설정을 on-off할 수 있다. • Autocommit을 1로 설정하면 자동완료 트랜잭션이 된다.(default) • Autocommit을 0로 설정하면 수동완료 트랜잭션이 된다.
<p>수동완료 트랜잭션</p>	<p>Update 학생 Set 주소 = '서울' Where 학번 = 1; Update 학생 Set 주소 = '평양' Where 학번 = 2; Update 학생 Set 주소 = '제주' Where 학번 = 3; Commit 또는 Rollback Tran //트랜잭션 끝</p> <hr/> <ul style="list-style-type: none"> • 수동완료 트랜잭션은 시작점은 지정하지 않고 끝나는 시점만 지정한다. • 수동완료 트랜잭션은 암시적(implicit) 트랜잭션이라고도 한다.

5. 트랜잭션(transaction)의 동작에 대한 설명으로 옳지 않은 것은? [2014년 국가 7급]

- ① 트랜잭션이 완료(committed)되었다는 사실을 DBMS로부터 응용프로그램이 확인한 경우라도 하드디스크에 위치한 데이터베이스 파일에는 해당 트랜잭션에 의해 변경된 내용이 반영되어 있지 않을 수도 있다.
- ② 트랜잭션이 마지막 명령문을 수행하여 부분완료(partially committed)된 상태에서도 철회(aborted) 상태로 옮겨갈 수 있다.
- ③ DBMS는 아직 완료되지 않은 트랜잭션이 변경한 내용을 데이터베이스 버퍼 공간의 제약으로 인하여 하드디스크의 데이터베이스 파일에 기록할 수 있다.
- ④ 시스템 고장으로 트랜잭션이 철회된 경우, 회복관리자에 의해 복구가 수행된 경우라도 하드디스크에 위치한 데이터베이스 파일에는 해당 트랜잭션에 의해 변경된 내용이 반영될 수 있다.

☞ 트랜잭션 동작

-
- 트랜잭션이 철회된 경우는 해당 트랜잭션에 의해 변경된 내용은 취소되어야 한다.(원자성)
-

정답 : ④

6. 트랜잭션의 특성에 대한 설명으로 옳지 않은 것은? [2020년 국가 7급]

- ① S은행 계좌에서 K은행 계좌로 이체하는 도중에 오류가 발생하면 수행된 트랜잭션을 취소하여 원래의 상태로 만들어야 한다.
- ② S은행 계좌와 K은행 계좌 금액의 합은 S계좌에서 K계좌로 이체한 후에도 그 금액의 합이 같아야 한다.
- ③ S은행 계좌에서 K은행 계좌로의 이체가 성공적으로 완료된 후에는 항상 이체가 완료된 상태를 유지해야 한다.
- ④ S은행 계좌에서 K은행 계좌로 이체하는 트랜잭션을 수행하는 동안 다른 트랜잭션이 관여할 수 없으나 참조할 수는 있다.

☞ 트랜잭션의 특성

-
- S은행 계좌에서 K은행 계좌로 이체하는 트랜잭션을 수행하는 동안 다른 트랜잭션이 관여할 수 없으나 참조할 수는 있다.(×)
→ 트랜잭션 실행 중에 있는 연산 중간 결과는 다른 트랜잭션이 접근할 수 없다.(격리성)
-

정답 : ④

7. 다음 중 트랜잭션 종류에 대한 설명으로 가장 옳은 것은? [2022년 군무원 7급]

- ① 명시적 트랜잭션은 트랜잭션의 시작과 끝을 사용자가 직접 명시적으로 지정하는 트랜잭션으로 수동 트랜잭션이라고 한다.
- ② 자동완료 트랜잭션은 SQL문의 실행 결과에 따라 자동으로 Commit 또는 Rollback하는 트랜잭션으로 암시적 트랜잭션이라고 한다.
- ③ 수동완료 트랜잭션은 트랜잭션의 끝만 사용자가 직접 명시적으로 지정하는 트랜잭션으로, 트랜잭션의 시작은 자동으로 실행되며 시스템 트랜잭션이라고 한다.
- ④ Autocommit 문을 사용하여 수동완료 모드를 설정하는 경우, Autocommit을 1로 설정한다.

☞ 트랜잭션 종류

- ① **명시적 트랜잭션**은 트랜잭션의 시작과 끝을 사용자가 직접 명시적으로 지정하는 트랜잭션으로 수동 트랜잭션이라고 한다.(○)

명시적 트랜잭션	<p>Begin Tran //트랜잭션 시작</p> <p>Update 학생 Set 주소 = '서울' Where 학번 = 1;</p> <p>Update 학생 Set 주소 = '평양' Where 학번 = 2;</p> <p>Commit 또는 Rollback Tran //트랜잭션 끝</p> <hr/> <ul style="list-style-type: none"> · 사용자가 트랜잭션 시작과 끝을 직접 명시하는 트랜잭션이다. · 이를 '사용자 트랜잭션' 또는 '수동 트랜잭션'이라 한다.
자동완료 트랜잭션	<p>Update 학생 Set 주소 = '서울' Where 학번 = 1;</p> <p>Update 학생 Set 주소 = '평양' Where 학번 = 2;</p> <hr/> <ul style="list-style-type: none"> · 트랜잭션 시작과 끝을 시스템이 자동으로 붙이는 트랜잭션이다. · 각 쿼리마다 자동으로 트랜잭션 시작과 끝을 붙인다. · 보통 DBMS의 설정을 on-off할 수 있다. · Autocommit을 1로 설정하면 자동완료 트랜잭션이 된다.(default) · Autocommit을 0로 설정하면 수동완료 트랜잭션이 된다.
수동완료 트랜잭션	<p>Update 학생 Set 주소 = '서울' Where 학번 = 1;</p> <p>Update 학생 Set 주소 = '평양' Where 학번 = 2;</p> <p>Update 학생 Set 주소 = '제주' Where 학번 = 3;</p> <p>Commit 또는 Rollback Tran //트랜잭션 끝</p> <hr/> <ul style="list-style-type: none"> · 수동완료 트랜잭션은 시작점은 지정하지 않고 끝나는 시점만 지정한다. · 수동완료 트랜잭션은 암시적(implicit) 트랜잭션이라고도 한다.

8. 트랜잭션의 ACID 특성 중 DBMS의 회복(recovery) 기능과 가장 밀접한 관련이 있는 특성을 옳게 짝지은 것은? [2021년 서울 7급]

- ① 원자성(atomicity), 일관성(consistency)
- ② 일관성(consistency), 격리성(isolation)
- ③ 격리성(isolation), 영속성(durability)
- ④ 영속성(durability), 원자성(atomicity)

♣ 트랜잭션의 ACID 특성

- DBMS는 트랜잭션의 원자성 유지를 위해 회복관리자를 작동시킨다.
 - DBMS는 트랜잭션의 일관성 유지를 위해 동시성제어와 무결성 제약조건을 이용한다.
 - DBMS는 트랜잭션의 격리성 유지를 위해 동시성제어 알고리즘을 작동시킨다.
 - DBMS는 트랜잭션의 영속성 유지를 위해 회복관리자를 작동시킨다.
-

정답 : ④