

● Transaction을 간단히 설명해라.

▶ transaction는 데이터베이스 내에서 수행되는 작업의 최소 단위로, 데이터베이스의 무결성을 유지하며, (즉, 데이터베이스에 저장된 데이터의 일관성과 정확성을 지키며) 데이터베이스의 상태를 변화시키는 기능을 수행합니다. transaction은 하나 이상의 query를 포함해야 하고, ACID라고 칭해지는 transaction이 안전하게 수행된다는 것을 보장하기 위한 성질 네 가지 - 원자성, 일관성, 고립성, 지속성의 4가지 규칙을 만족해야 합니다.

● Transaction

▶ 데이터베이스에서 Transaction이 필요한 이유는 데이터를 다룰 때 장애가 일어나는 경우 transaction은 장애 발생시 데이터를 복구하는 작업의 단위가 되기 때문입니다. 또한 데이터베이스에서 여러 작업이 동시에 같은 데이터를 다룰 때, transaction을 통해 이 작업을 서로 분리하고, 이를 통해 오류가 발생하지 않게 합니다. DBMS(DataBase Management System)는 transaction이 이러한 규칙을 유지하도록 지원합니다.

▶ 예시: Transaction이란 분리될 수 없는 업무 처리의 최소 단위입니다. 예를 들어 은행 시스템에서 입금과 출금은 하나의 작업 단위, transaction으로 작동해야 합니다. 은행 시스템에서 A가 100만원을 출금해서 B에게 입금하는 상황에서 A의 잔고에서 100만원을 출금하였는데, 이 때 전산오류가 생겨서 B의 계좌에는 100만원이 입금 되지 않는다면 전산시스템의 치명적인 오류입니다. 이렇게 예상치 못하게 오류가 발생하여 하여 데이터의 부정합이 발생하는 경우, 다시 원상복귀 해야합니다. 따라서 모든 입출금은 하나의 묶음 형태로 작동해야 합니다. 출금을 했으면 입금을 마치든지 아니면 아예 없던 일이 되어야 합니다. 이런 식으로 두 행위는 분리될 수 없는 하나의 거래로 처리되어야 하는 단일 업무입니다. 이러한 업무 처리의 최소 단위를 데이터베이스에서는 transaction이라고 합니다.

START TRANSACTION

```
// (1) A 계좌 잔액 가져옴 A = 1000
// (2) B 계좌 잔액 가져옴 B = 1000

// (3) A 출금 A = A - 100
// (4) B 입금 B = B + 100
UPDATE Customer SET balance = balance - 100 WHERE name='A';
UPDATE Customer SET balance = balance + 100 WHERE name='B';

//COMMIT

// (5) A 계좌 잔액 저장 A = 900
// (6) B 계좌 잔액 저장 B = 1100
```

COMMIT

● ACID (Atomicity, Consistency, Isolation, Durability)

▶ 트랜잭션은 데이터베이스의 무결성을 유지하기 위해 ACID(원자성, 일관성, 고립성, 지속성)의 성질을 갖습니다. Atomicity(원자성)이란, all or nothing으로, transaction에 포함된 작업은 전부 수행되거나 아니면 전부 수행되지 말아야 함을 뜻합니다.

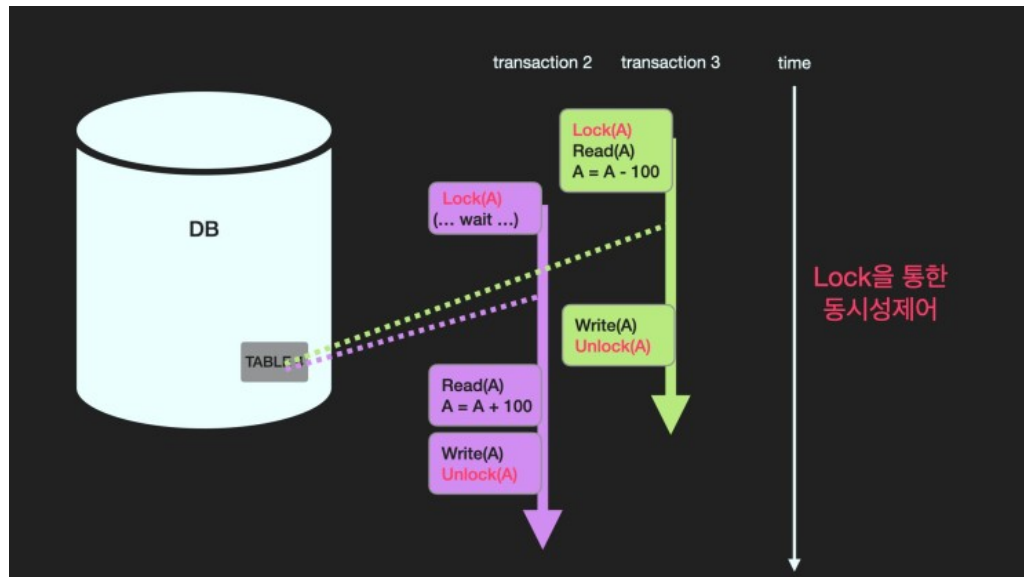
Consistency(일관성)이란, transaction이 실행을 성공적으로 완료하면 언제나 일관성 있는 데이터베이스 상태로 유지하는 것을 의미합니다. 예를 들면, 송금 전후 모두 잔액의 data type은 integer이어야 합니다.

Isolation(고립성)이란, 여러 Transaction이 동시에 수행될 때 각 transaction이 다른 transaction의 연산 작업이 끼어들지 못하도록 보장하여 독립적으로 작업을 수행함을 뜻합니다. 동시에 수행되는 transaction이 동일한 data를 동시에 갱신(update)할 때 어느 한 transaction의 갱신이 무효화되는 갱신손실이 발생하지 않도록 데이터 lock을 통해 제어해줘야 하고, 이런 DBMS의 기능을 동시성 제어(concurrency control)라고 합니다.

마지막으로 Durability(지속성)이란, 성공적으로 수행된 transaction은 데이터베이스에 영원히 반영되어야 함을 의미합니다. transaction이 완료되어 저장된 데이터베이스는 저장 후에 생기는 정전, 장애, 오류 등에 영향을 받지 않아야 합니다.

- 동시성 제어(concurrency control)

▶ (여러 개의 transaction이 한 개의 데이터를 동시에 갱신(update)할 때 어느 한 transaction의 갱신이 무효화 될 수 있는데 이를 갱신손실이라고 합니다. 동시성제어를 통해 갱신손실을 미리 막을 수 있습니다. 즉,) transaction이 동시에 수행될 때 일관성을 해치지 않도록 transaction의 데이터 접근을 제어하는 DBMS의 기능을 동시성 제어라고 합니다. 갱신손실 문제를 해결하기 위한 방법중에 하나로 데이터를 수정 중에 있는 transaction은 해당 데이터를 Lock으로 잠금장치를 하여 다른 Transaction이 접근하지 못하게 하는 방법이 있습니다. Lock이 걸린 데이터는 Unlock이 될 때까지 다른 Transaction들은 접근하지 못하고 기다려야 합니다.



- COMMIT과 ROLLBACK에 대해 설명해라

▶ 데이터베이스는 COMMIT과 ROLLBACK 명령어를 통해 데이터 무결성을 보장합니다. COMMIT이란 transaction 작업을 완료했다고 확정하는 명령어입니다. transaction 작업 내용을 실제 DB에 저장하고, DB가 변경됩니다. ROLLBACK은 작업 중 문제가 발생했을 때, transaction 처리 과정에서 발생한 변경 사항을 취소하고, 이전 COMMIT의 상태로 되돌립니다.