



Section3. 트랜잭션과 무결성

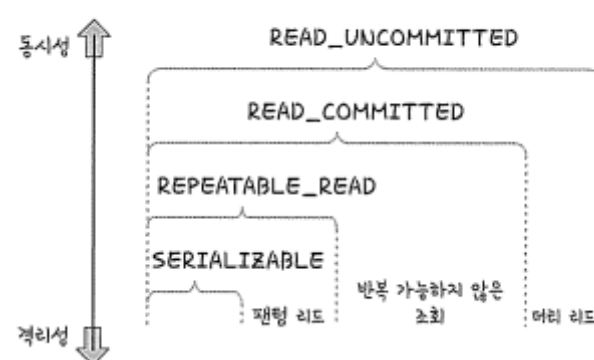
4.3.1 트랜잭션

- 데이터베이스에서 하나의 논리적 기능을 수행하기 위한 작업의 단위.
- 여러개의 쿼리들을 하나로 묶는 단위
- ACID 특징을 가짐

ACID

- 원자성(Atomicity)
 - all or nothing
 - 트랜잭션과 관련된 일이 모두 수행되었거나 되지 않았거나를 보장하는 특징
 - 롤백 발생 시, 어떻게 해야 할것인지에 대한 해결방법이 있어야하고, 트랜잭션 전파를 신경써서 관리해야함.
- 일관성(Consistency)
 - 허용된 방식으로만 데이터를 변경해야 함.
 - 데이터 베이스에 기록된 모든 데이터는 여러가지 조건, 규칙에 따라 유효함을 가져야 함.
- 격리성(isolation)
 - 트랜잭션 수행 시 , 서로 끼어들지 못하는 것.
 - 복수의 병렬 트랜잭션은 서로 격리되어 마치 순차적으로 실행되는 것처럼 작동해야 함.
 - 여러 사용자가 같은 데이터에 접근할 수 있어야 함.
 - 여러 격리 수준으로 나누어 격리성을 보장

▼ 그림 4-28 여러 개의 격리 수준



- SERIALIZABLE, REPEATABLE_READ, READ_COMMITTED, READ_UNCOMMITTED
- 위로 갈수록 동시성이 강해지고 격리성이 약해짐.

- 지속성(Durability)
 - 성공적으로 수행된 트랜잭션은 영원히 반영되어야 하는 것.
 - 시스템 장애가 발생해도 원래 상태로 복구하는 회복기능이 있어야 함.
 - 이를 위해 체크섬,롤백, 저널링 등의 기능 제공

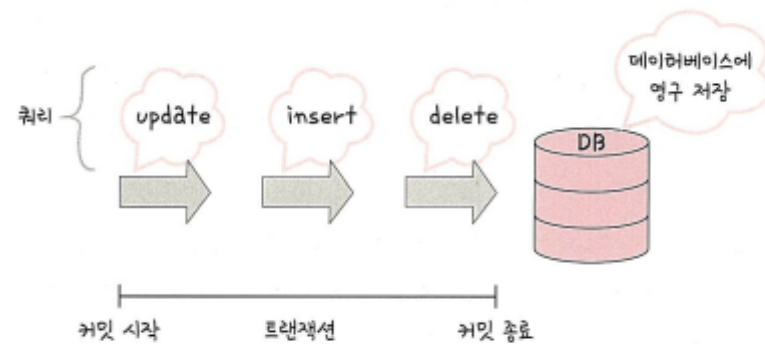
체크섬 : 중복검사의 한 형태, 오류 정정을 통해 송신된 자료의 무결성을 보호하는 단순한 방법

저널링 : 파일 시스템 또는 데이터베이스 시스템에 변경 사항을 반영하기 전에 로깅하는 것. 트랜잭션 등 변경 사항에 대한 로그를 남기는 것.

커밋과 롤백

- 커밋
 - 여러 쿼리가 성공적으로 처리되었다고 확정하는 명령어.
 - 트랜잭션 단위로 수행.
 - 변경된 내용이 모두 영구적으로 저장되는 것을 말함.
 - 커밋이 수행되었다 == 하나의 트랜잭션이 성공적으로 수행되었다.

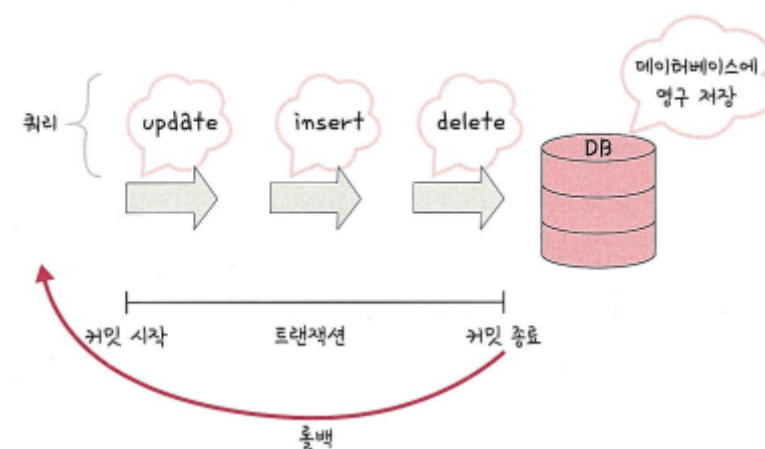
▼ 그림 4-26 커밋



앞의 그림처럼 update, insert, delete의 쿼리가 하나의 트랜잭션 단위로 수행되고 이후에 데이터베이스에 영구 저장됩니다.

- 롤백
 - 예러나 여러 이슈 때문에 트랜잭션 전으로 돌려야 할 때 사용.
 - 트랜잭션으로 처리한 하나의 묶음 과정을 일어나기 전으로 돌리는 일(취소)

▼ 그림 4-27 롤백



트랜잭션 전파

- 트랜잭션 수행 시, 여러 트랜잭션 관련 메서드를 하나의 트랜잭션에 묶이도록 하는 것.

격리 수준

- **SERIALIZABLE**
 - 트랜잭션을 순차적으로 진행시키는 것.
 - 트랜잭션이 동시에 같은 행에 접근 불가
 - 매우 엄격한 수준으로 해당 행에 대해 격리 시킴
 - 가장 성능이 떨어지는 격리 수준
- **REPEATABLE_READ**
 - 수정한 행을 다른 트랜잭션이 수정할 수 없도록 막아줌.
 - 새로운 행을 추가하는 것은 막지 않음.

- 이후에 추가된 행이 발견될 수도 있음.
- **READ_COMMITTED**
 - 가장 많이 사용되는 격리 수준.
 - 대부분의 DBMS에서 기본값으로 설정됨.
 - 커밋 완료된 데이터에 대해서만 조회를 허용
 - 트랜잭션 A가 수정한 행을 트랜잭션 B가 수정할 수 있음.
 - 따라서, 트랜잭션 A가 같은 행을 다시 읽을 때 다른 내용 발견가능성 있음.
- **READ_UNCOMMITTED**
 - 가장 낮은 격리수준.
 - 하나의 트랜잭션이 커밋되기 이전에 다른 트랜잭션에 노출되는 문제가 있지만 가장 빠름.
 - 데이터 무결성을 위해 사용하지 않는 것이 이상적.

격리 수준에 따라 발생하는 현상

- 팬텀 리드
 - 한 트랜잭션 내에서 동일한 쿼리 보냈을 때, 해당 조회 결과가 다른 경우.
- 반복 가능하지 않은 조회
 - 한 트랜잭션 내의 같은 행에 두 번 이상의 조회가 발생했는데, 그 값이 다른 경우.
- 더티리드
 - 반복가능하지 않은 조회와 유사.
 - 한 트랜잭션이 실행중일 때, 다른 트랜잭션에 의해 수정되었지만 아직 커밋되지 않은 행의 데이터를 읽을 수 있을 때 발생.

4.3.2 무결성

- 데이터의 정확성, 일관성, 유효성을 유지하는 것.
- 무결성이 유지되어야 데이터베이스에 저장된 데이터 값과 그 값에 해당하는 현실 세계의 실제 값이 일치하는지에 대한 신뢰가 생김.

• 무결성의 종류

- 개체 무결성 : 기본키로 선택된 필드는 빈값을 허용하지 않음.
- 참조 무결성 : 서로 참조 관계에 있는 두 테이블의 데이터는 항상 일관된 값을 유지해야 함.
- 고유 무결성 : 특정 속성에 대해 고유한 값을 가지도록 조건이 주어진 경우, 그 속성의 값은 모두 고유한 값을 가짐
- NULL 무결성 : 특정 속성 값에 NULL이 올 수 없다는 조건이 주어진 경우, 그 속성 값은 NULL이 될 수 없다는 제약 조건