

트랜잭션 격리 수준

SSAFY 9기 구미 6반

김도연

동시성 제어

- ◆ 모든 트랜잭션들이 순서대로만 실행된다면?-> 문제 발생하지 않음..
- ◆ 하지만 트랜잭션들은 동시에 여러개가 실행된다
- ◆ 동시성 제어는 동시에 트랜잭션이 실행되어도 결과가 순차적으로 실행된것과 동일하도록 보장한다.
- ◆ 동시성 제어가 없을때 나타나는 문제들
 - Lost Update
 - Dirty Read
 - Unrepeatable Read
 - Phantom Read

Lost Update

- ◆ 갱신 없어짐
- ◆ 아래 예시에서 트랜잭션1, 2 가 순차적으로 일어나면 A에는 300이 더해져야함
- ◆ 하지만 아래 경우에는 트랜잭션1의 +100은 없어지고 트랜잭션2의 +200만 반영됨
- ◆ 표준 SQL에서 이것이 적용되는 부분은 없음..

트랜잭션 1	트랜잭션 2
read(A)	
	read(A)
A=A+100; write(A)	
	A=A+200 write(A)

Dirty Read

◆ 잘못된 판독

◆ ex) 중고 물품 거래

- 트랜잭션 1에서 통장 잔고에 100을 더함 -> 트랜잭션 2에서 통장 잔고를 보여줌
- -> 잔고를 본 사람이 입금 완료되었다고 생각해서 중고물품을 배송함
- -> 트랜잭션 1에서 롤백함-> 중고물품을 판매한 사람이 손해보게됨

트랜잭션 1	트랜잭션 2
<pre>read(A) A=A+100; write(A)</pre>	<pre>read(A) display(A) commit;</pre>
<pre>rollback;</pre>	

Non Repeatable Read

◆ 반복해서 입을때 다른 결과가 나옴

- 아래 예시에서 트랜잭션2 에서 처음 조회한 A값과 나중에 조회한 A값이 다름
- 트랜잭션1, 2 가 순차적으로 일어나면 트랜잭션2에서는 처음 조회한 A와 나중에 조회한 A가 동일하지만 아래에서는 그렇지 않다

트랜잭션 1	트랜잭션 2
read(A) A=A+100 write(A) commit	read(A)
	read(A)

Phantom Read

- ◆ 없던것이 갑자기 생김
- ◆ Non Repeatable과의 차이: Non Repeatable은 값이 수정되는것이고 Phantom Read는 새로운 값이 추가되는것
- ◆ ex) student테이블 조회
 - 트랜잭션2에서 student 테이블을 조회함->트랜잭션1에서 새로운 학생을 추가
 - ->트랜잭션2에서 다시 student테이블을 조회하면 없던 값이 생겨있음
 - 순차적으로 트랜잭션을 실행할때와는 다른 결과를 보여줌

트랜잭션 1	트랜잭션 2
<pre>insert into student values(...); commit;</pre>	<pre>select * from student; select * from student;</pre>

트랜잭션 격리 수준

- ◆ 동시성 제어 문제를 해결하기 위해 도입됨
- ◆ 동시에 여러 트랜잭션이 처리될 때, 트랜잭션간 얼마나 서로 고립되어 있는지 나타내는것
- ◆ 고립 정도가 높아질수록 성능이 떨어짐

격리 수준

SQL 표준에서는 아래의 네가지 격리 레벨을 정의함

- ◆ Read Uncommitted
- ◆ Read Committed
- ◆ Repeatable Read
- ◆ Serializable

◆ 오라클 DB: Read Committed(디폴트), Serializable 가능

Read Uncommitted

- ◆ 격리 레벨 0
- ◆ 커밋되지 않은 읽기
- ◆ 트랜잭션 처리중이거나 아직 커밋 안된 데이터를 다른 트랜잭션이 읽는것 허용
- ◆ 일관성 유지 불가
- ◆ 발생 가능한 부정합
 - Dirty Read
 - Non Repeatable Read
 - Phantom Read

Read Committed

◆ 격리 레벨 1

◆ 커밋된 데이터만 읽는다

◆ Dirty Read 방지됨

- 이미 커밋 완료된 값만 읽기때문에 dirty read방지
- 트랜잭션1 에서 변경하고 있는 도중에 트랜잭션2에서 읽기 시도하면 트랜잭션2는 트랜잭션1이 변경한 값을 읽는것이 아니라 이미 커밋된값을 (undo영역이라는 백업된 영역에서 읽는다) 가져간다.

◆ 발생 가능한 부정합

- Non Repeatable Read
- Phantom Read

트랜잭션 1	트랜잭션 2
read(A) A=A+100 write(A) commit	read(A)
	read(A)

Repeatable Read

◆ 격리 레벨 2

◆ 트랜잭션이 범위 내에서 조회한 데이터 내용이 항상 동일함을 보장함

◆ 자신의 트랜잭션 번호보다 낮은 트랜잭션 번호에서 커밋된 것만 볼 수 있음

- 자신보다 이전에 실행되고 커밋된 트랜잭션이 만들어낸 결과만 볼 수 있다는 의미

◆ Dirty Read, Non Repeatable Read 방지

- undo 영역에 변경 전의 값이 백업되므로 undo를 읽어 읽은 값은 동일하게 됨

◆ 발생 가능한 부정합

- Phantom Read: undo 영역의 변경 전 데이터를 가져오는것이 아니라 현재 레코드의 값을 가져오게 됨

트랜잭션 1	트랜잭션 2
insert into t values(3,3)	select * from t;
commit;	select * from t;

Serializable

- ◆ 격리 레벨 3
- ◆ 가장 엄격한 격리 수준으로 완벽한 읽기 일관성 모드 제공
- ◆ 다른 사용자는 트랜잭션 영역에 해당되는 데이터에 대한 수정, 입력 불가능

격리 레벨별 발생 가능한 부정합 정리

격리 수준	Dirty Read	Non Repeatable Read	Phantom Read
Read Uncommitted	O	O	O
Read Committed	X	O	O
Repeatable Read	X	X	O
Serializable	X	X	X

참고자료

- ◆ 영남대학교 컴퓨터공학과 데이터베이스 수업자료
- ◆ <https://dar0m.tistory.com/225>
- ◆ <https://tecoble.techcourse.co.kr/post/2022-11-07-mysql-isolation/>
- ◆ <https://jokerkwu.tistory.com/125>
- ◆ <https://medium.com/@10x.developer.kr/db-%ED%8A%B8%EB%9E%9C%EC%9E%AD%EC%85%98-%EA%B2%A9%EB%A6%AC-%EC%88%98%EC%A4%80-%EA%B7%B8%EB%A6%BC%EA%B3%BC-%EC%98%88%EC%8B%9C%EB%A1%9C-%EC%89%BD%EA%B2%8C-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0-1%ED%8E%B8-5bef68de8b7b>
- ◆ https://www.postgresql.kr/blog/pg_phantom_read.html
- ◆ <https://steady-coding.tistory.com/562>