

# Race condition에서 동시 쓰기 문제와 해결법

≡ 태그

## 문제점 - DB에 동시에 Write

- DB에 두 명 이상의 유저가 동시에 접근할 때 발생할 수 있는 문제
- Count 숫자를 늘리는 경우 최신 값으로 갱신 되지 않을 수 있음.
- 상태 변경 값의 경우 최신 상태 갱신이 잘못 기록될 수 있음
- 그 외 content를 업데이트 하는 경우에도 무엇이 최신의 값인지 결정하기 어려움.

## 해결책 - DB 에 lock을 건다

DB에 lock을 값을 읽고 쓰는 행위를 제어할 수 있음. lock은 쓰기 lock과 읽기 lock을 별도로 지정할 수 있음. 하지만 이렇게 lock를 사용하면 DB 사용 성능이 떨어질 수 있으므로 현재 사용성에 맞는 lock 수준을 정하는 것이 중요함.

## DB lock - 트랜잭션 수준

### 1. Read uncommitted(Level 0)

아직 commit 되지 않은 데이터를 다른 트랜잭션이 읽을 수 있음.

발생할 수 있는 문제: Dirty Read가 발생할 수 있음. commit 되지 않은 기록을 조회하기 때문에 잘못된 값을 읽어올 수도 있다.

### 2. Read committed(Level 1)

커밋이 이루어진 후에만 다른 트랜잭션이 조회할 수 있음. Oracle에서 기본으로 사용하는 격리 수준.

Non-repeatable read가 발생할 수 있음. commit 되기 이전의 기록을 읽기 때문에 commit 되고난 후의 데이터를 가져오므로 데이터 정합성 문제 발생할 수 있음

### 3. Repeatable Read(Level 2)

자신의 트랜잭션 번호보다 아래의 트랜잭션 번호에서 커밋된 내용만 보게 됨. 하나의 트랜잭션이 끝날 때까지 다른 트랜잭션에서 READ가 금지됨

Phantom Read가 발생할 수 있음. 읽기 쿼리가 두 번 발생 했는데, 그 사이 다른 서버에서 수정을 발생시켜서 값을 추가하거나 삭제하면 row 넘버가 꼬이면서 다른 값을 읽어들일 수 있다.

### 4. Serializable(Level 3)

한 트랜잭션이 Read 중이면 다른 트랜잭션은 값을 추가, 변경, 삭제 할 수 없다.

동시성 문제는 없어지지만 DB 사용 성능이 매우 떨어진다.