

## 4.5 MySQL의 격리 수준

🕒 생성일	@2021년 6월 6일 오후 4:08
🏷 태그	

Isolation Level : 동시에 여러 트랜잭션이 처리 될 때, 특정 트랜잭션이 다른 트랜잭션에서 변경하거나 조회하는 데이터를 볼 수 있도록 허용할지 말지를 결정하는 것이다.

### DB에서 발생할 수 있는 3가지 부정합

	DIRTY READ	NON-REPEATABLE READ	PHANTOM READ
READ UNCOMMITTED	발생	발생	발생
READ COMMITTED	발생하지 않음	발생	발생
REPEATABLE READ	발생하지 않음	발생하지 않음	발생 (InnoDB는 발생하지 않음)
SERIALIZABLE	발생하지 않음	발생하지 않음	발생하지 않음

#### [격리 수준]

격리수준이 높아질 수록 동시성도 떨어진다

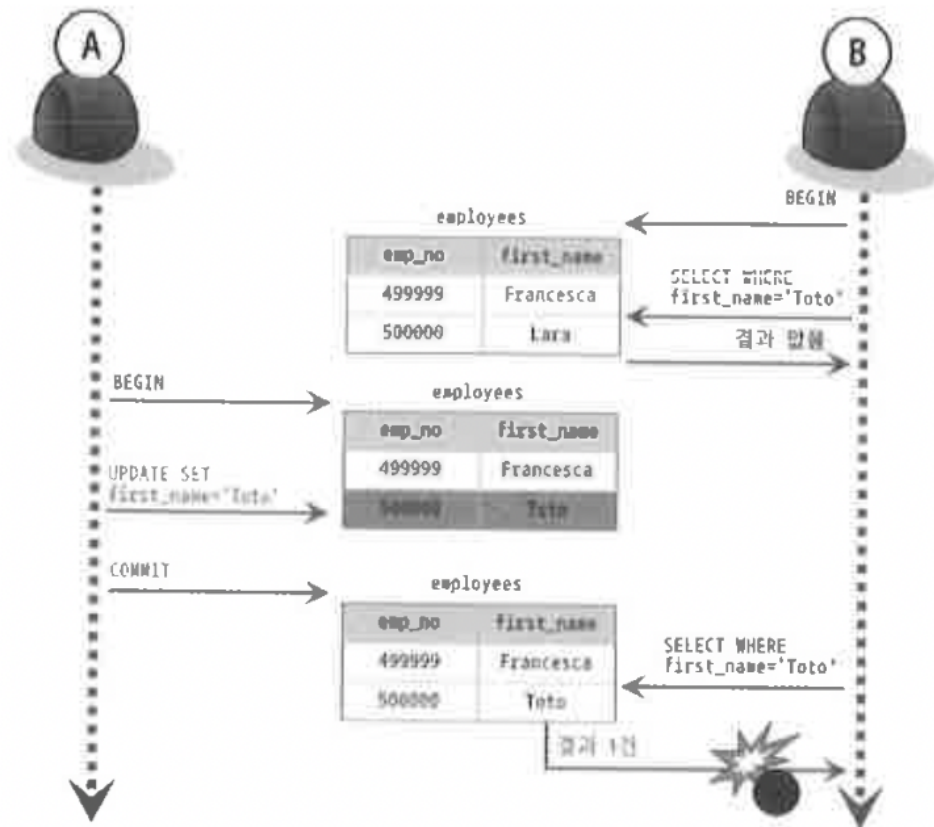
##### 1. READ UNCOMMITTED = Dirty Read

: 커밋되지 않는 상태에서도 타 트랜잭션에서 변경 데이터를 read 할 수 있다

##### 2. READ COMMITTED = Oracle

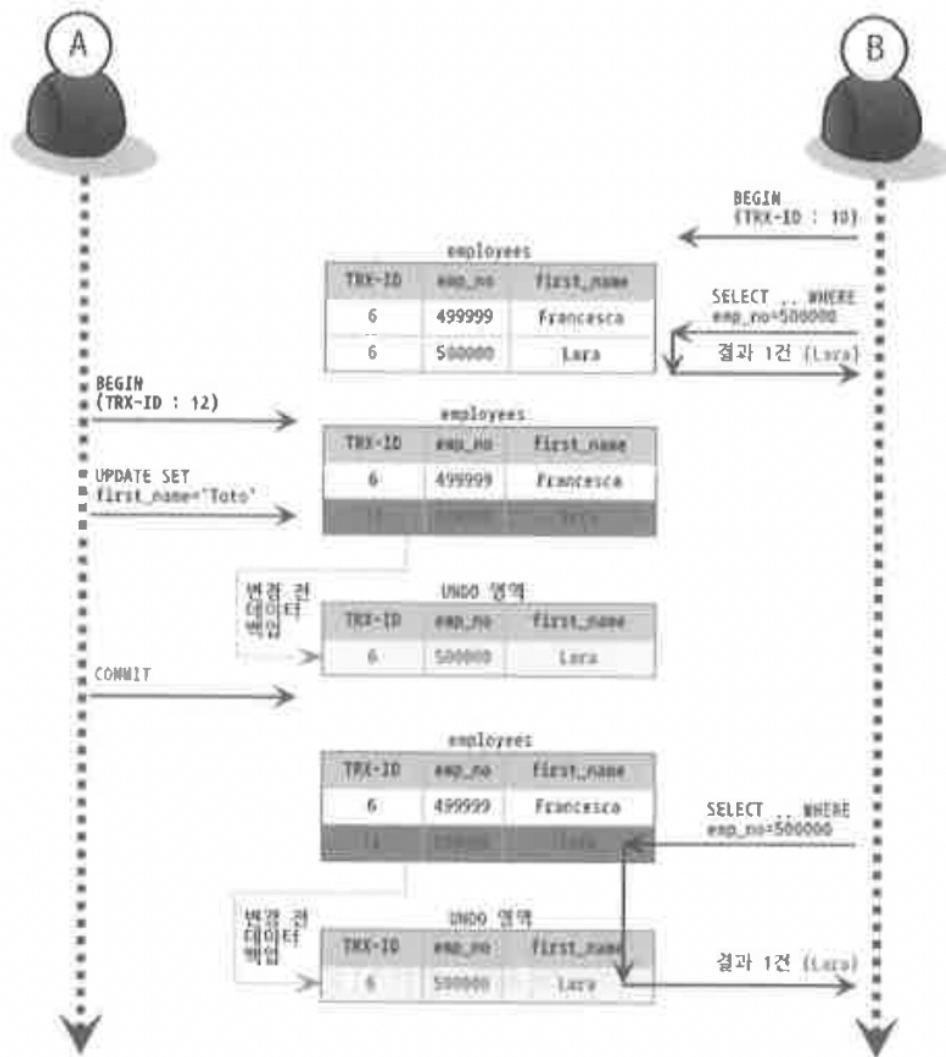
: 온라인에서 가장 많이 선택되는 격리 수준, commit이 완료된 데이터만 조회할 수 있다. 커밋되기전에는 언두영역으로 백업된 데이터를 타 트랜잭션에서 read한다

- non-repeatable read 발생가능 (한 트랜잭션에서 같은 select 쿼리 실행시 같은 결과를 가져와야한다는 repeatable read 정합성에 어긋남) → 은행 트랜잭션에서 문제가 될 수 있다



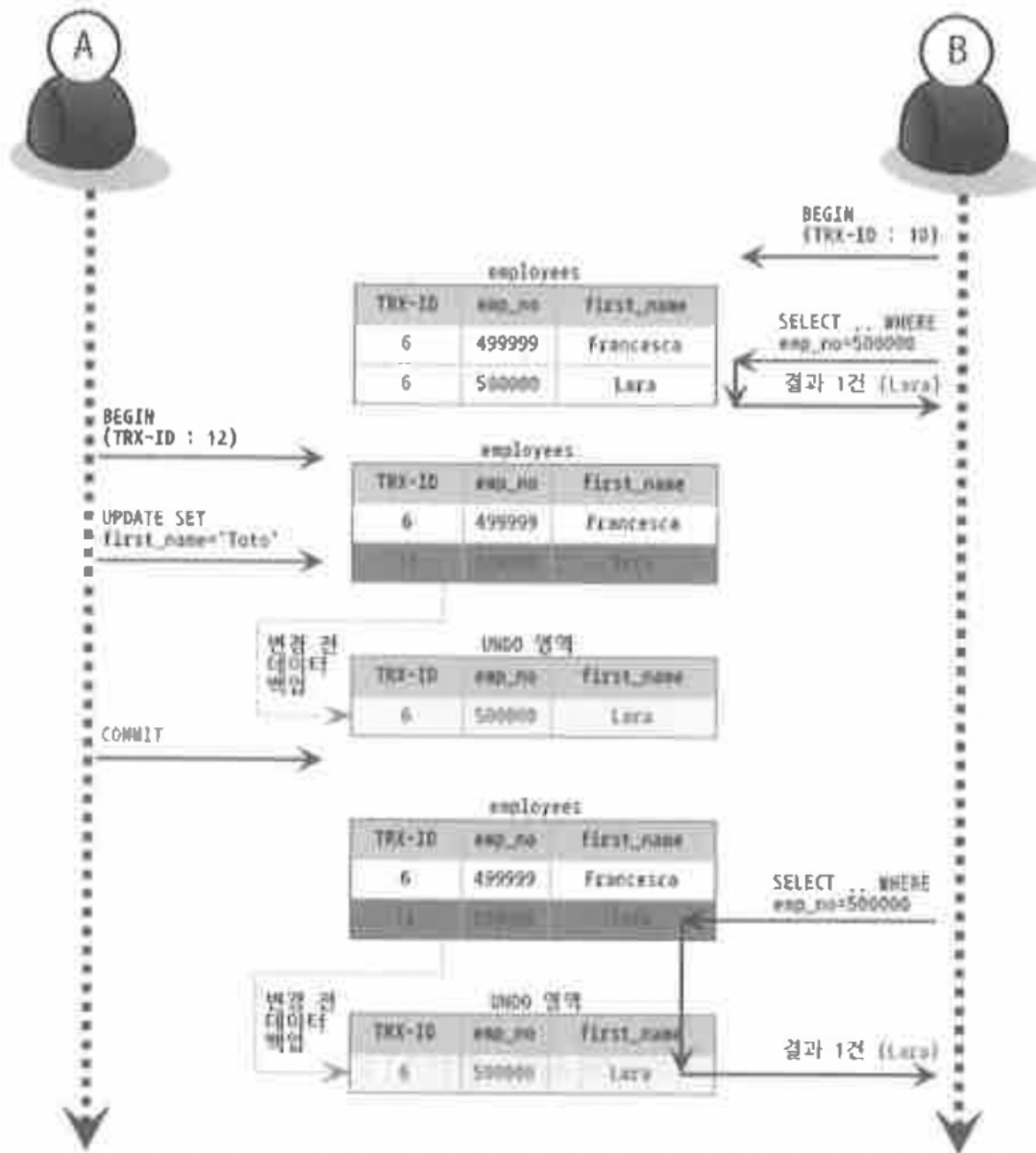
[그림 4-6] NOT-REPEATABLE READ

- phantom read 발생가능
3. REPEATABLE READ = MySQL InnoDB 스토리지 엔진  
: MVCC 를 위해 언두영역에 백업된 데이터로 한 트랜잭션 내에서는 동일한 결과를 보여줄 수 있도록 보장함



[그림 4-7] REPEATABLE READ

- 트랜잭션 번호의 구간 내에서 백업된 언두 데이터가 보존됨으로써, 자신의 트랜잭션 번호보다 작은 트랜잭션 번호에서 변경한 데이터만 보게 됨
- Repeatable Read 트랜잭션은 처음으로 SELECT를 수행한 시간을 기록한 뒤 그 이후에는 모든 SELECT 마다 해당 시점을 기준으로 **Consistent Read**를 수행
- 그러므로 트랜잭션 도중 다른 트랜잭션이 COMMIT 되더라도 새로이 COMMIT 된 데이터는 보이지 않게됨
  - 그 이유는 첫 **SELECT** 시에 생성된 **SNAPSHOT**을 읽기 때문



[그림 4-7] REPEATABLE READ

- Phantom read는 존재한다 : 다른 트랜잭션에서 수행한 변경 작업에 의해 레코드가 보였다가 안보였다가 하는 현상... 레코드에 쓰기 잠금을 걸어야하는데 언두 레코드에는 잠금을 걸 수 없기 때문..

#### 4. SERIALIZABLE

- 트랜잭션이 완료될 때까지 다른 트랜잭션이 해당되는 영역에 대한 수정 및 입력을 할 수 없다.
- 완벽한 LOCK 을 건다.