

03. 트랜잭션 이해

.....

트랜잭션 - DB 예제2 - 자동 커밋, 수동 커밋

...

수동 커밋 설정

```
set autocommit false; //
insert into member(member_id, money) values ('data3',10000);
insert into member(member_id, money) values ('data4',10000); commit; //
```

보통 자동 커밋 모드가 기본으로 설정된 경우가 많기 때문에, **수동 커밋 모드로 설정하는 것을 트랜잭션을 시작한다고 표현할 수 있다.**
수동 커밋 설정을 하면 이후에 꼭 commit, rollback 을 호출해야 한다.

만약 commit도 rollback도 안하면 timeout이 발생하고 자동으로 rollback하게 된다.

참고로 수동 커밋 모드나 자동 커밋 모드는 한번 설정하면 해당 세션에서는 계속 유지된다. 중간에 변경하는 것은 가능하다.

DB 락 - 조희

일반적인 조희는 락을 사용하지 않는다

데이터베이스마다 다르지만, 보통 데이터를 조희할 때는 락을 획득하지 않고 바로 데이터를 조희할 수 있다.

예를 들어서 세션1이 락을 획득하고 데이터를 변경하고 있어도, 세션2에서 데이터를 조희는 할 수 있다.

물론 세션2에서 조희가 아니라 데이터를 변경하려면 락이 필요하기 때문에 락이 돌아올 때 까지 대기해야 한다.

조희와 락 - SELECT 할때도 lock을 가져갈수 있다!!

데이터를 조희할 때도 락을 획득하고 싶을 때가 있다. 이럴 때는, **`select for update`** 구문을 사용하면 된다.
이렇게 하면 세션1이 조희 시점에 락을 가져가버리기 때문에 다른 세션에서 해당 데이터를 변경할 수 없다. 물론 이 경우도 트랜잭션을 커밋하면 락을 반납한다.

조희 시점에 락이 필요한 경우는 언제일까?

트랜잭션 종료 시점까지 해당 데이터를 다른 곳에서 변경하지 못하도록 강제로 막아야 할 때 사용한다.

예를 들어서 애플리케이션 로직에서 memberA 의 금액을 조희한 다음에 이 금액 정보로 애플리케이션에서 어떤 계산을 수행한다.

그런데 이 계산이 돈과 관련된 매우 중요한 계산이어서 계산을 완료할 때 까지 memberA 의 금액을 다른곳에서 변경하면 안된다. 이럴 때 조희 시점에 락을 획득하면 된다.

여기서 락을 획득해도 다른 세션에서 조희는 할 수 있지만 변경을 불가능하다.



