12.1 InnoDB의 기본 잠금 방식

생성일	@2021년 8월 8일 오후 4:32
태그	

MySQL 에서 InnoDB → 쿼리의 패턴별로 사용하는 잠금이 다름

InnoDB 제외한 스토리지 엔진 → 대부분 테이블 잠금을 지원

1. SELECT

REPEATABLE-READ 이하의 트랜잭션 격리 수준

InnoDB 테이블에 대한 SELECT 쿼리는 기본적으로 아무 잠금을 사용하지 않음

SERIALIZABLE 격리 수준

모든 SELECT 쿼리에 자동적으로 `LOCK IN SHARE MODE`가 덧붙여져서 실행되는 효과를 냄

이 격리 수준에서 모든 SELECT 쿼리는 읽기 잠금을 걸고 레코드를 읽는다

⇒ 이 격리 수준에서는 처리 성능이 떨어짐

⇒ 하지만 대부분의 InnoDB 격리 수준은 REPEATABLE-READ, READ-COMMITTED 격리수준이므로 걱정안해도됨

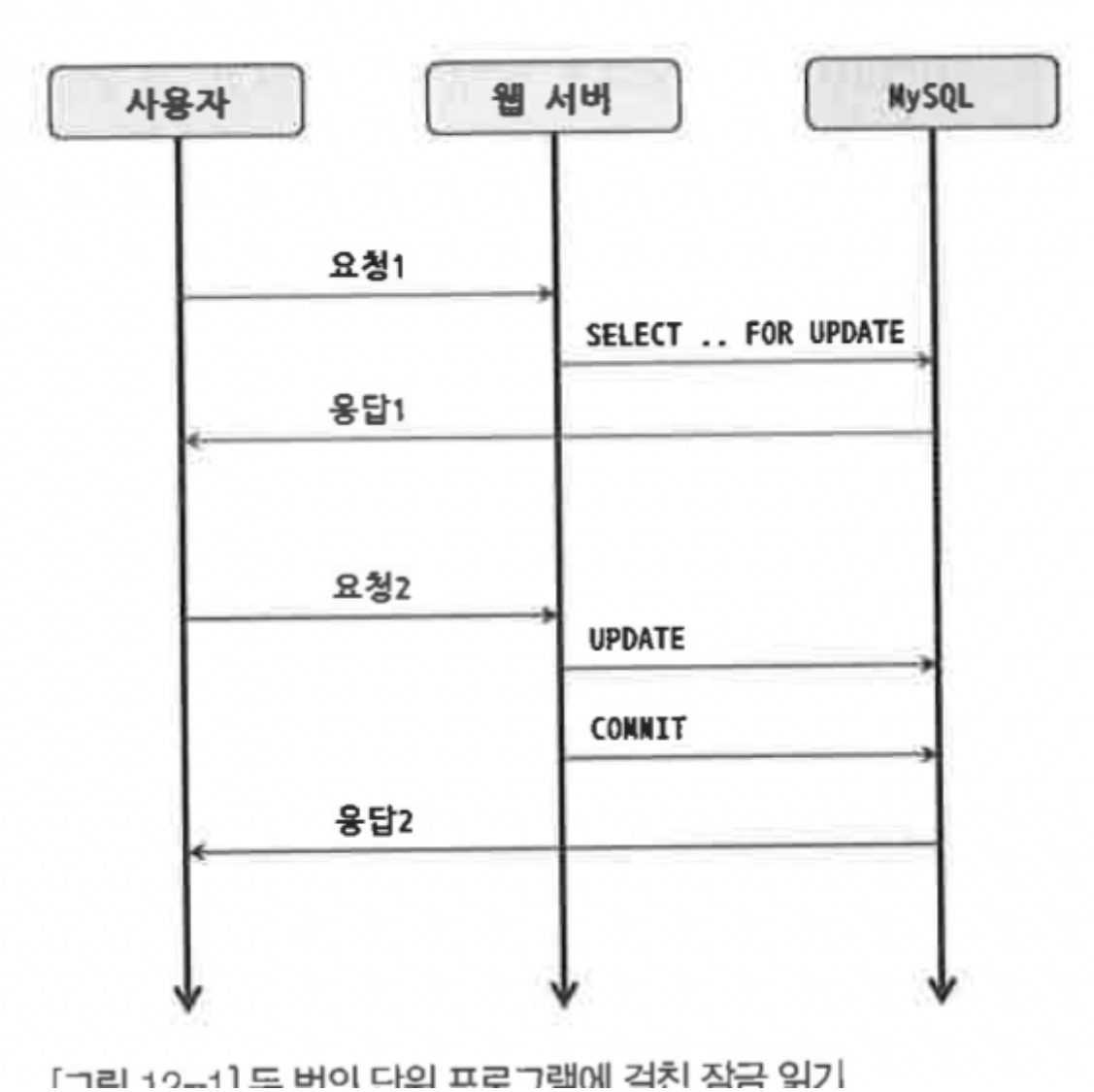
SELECT 쿼리로 읽은 레코드를 잠그는 방법

1. 읽기 모드 잠금
2. 쓰기 모드 잠금

```
SELECT * FROM employees WHERE emp_no=10001 LOCK IN SHARE MODE;
SELECT * FROM employees WHERE emp_no=10001 FOR UPDATE;
```

- 둘 다 해당 SELECT 로 읽은 것을 다른 커넥션의 트랜잭션에서 변경하지 못하게 막음
- COMMIT이나 ROLLBACK이 실행되면 잠금이 해제된다
  - 반드시 하나의 트랜잭션에서만 유효하다

- 여러번의 사용자 요청으로 완료되는 프로그램에서는 위의 두 읽기 잠금을 사용해서는 안된다



[그림 12-1] 두 번의 다음 프로그램에 걸친 잠금 읽기

- 커넥션 풀을 사용하는 웹 프로그램에서, 첫 번째 요청 이후 그 커넥션은 다른 용도로 사용할 수 없기 때문
- 또, 사용자가 두번째 요청하지 않고 웹 브라우저 종료시, 첫번째 요청에서 잠긴 레코드는 영원히 잠금 해제가 안됨
- 해제되지 않는 잠금의 누수 발행하면,,, 잠금 대기 상태 커넥션들 때문에 커넥션 수가 계속 증가되면서 DB 서버 망,,

LOCK IN SHARE MODE

: 읽기 잠금만 거는 잠금. 잠금을 획득한 트랜잭션에서도 변경하려면 쓰기 잠금을 다시 획득해야함.

- 쓰기 잠금을 획득하기 위해 기다려야 할 수도 있음
    - 여기서 쓰기 잠금을 획득하는 과정은 데드락 유발의 일반적인 형태이다

그래서..

FOR UPDATE

: SELECT 쿼리 문장으로 읽은 레코드를 쓰기 모드로 잠금 것

- 읽은 다음 변경까지 해야한다면 LOCK IN SHARE MODE보다 얘를 쓰는 것이 더 좋다.
- 다만, 다른 트랜잭션에 대한 영향도가 크기 때문에!! 반드시 읽은 레코드를 변경해야 할 때만 사용하는 것이 좋다

2. INSERT, UPDATE, DELETE

일반적으로 위 DML 쿼리는 모두 기본적으로 쓰기 잠금을 사용

- 필요시에는 읽기 잠금을 사용할 수도 있다
- 즉, 커넥션의 AutoCommit이 활성화된 상태라 하더라도 SQL을 처리하기 위해 잠금을 걸고 해제하는 작업은 생략하지 않는다.

AutoCommit 활성화 모드

각 SQL 문장별로 자동으로 트랜잭션이 시작되고 종료됨

- 이 모드에서도 BEGIN이나 START TRANSACTION 명령을 실행해 명시적으로 트랜잭션을 시작할 수도 있는데, 이때는 반드시 수동으로 트랜잭션을 종료해야함

AutoCommit 비활성화된 상태

SQL 문장의 실행과 함께 자동으로 트랜잭션이 시작되지만, 종료는 반드시 COMMIT이나 ROLLBACK명령을 사용해 수동으로 종료시켜야한다

InnoDB의 레코드별 잠금

UPDATE, DELETE 문장을 실행할 때, SQL 문장이 스캔하는 인덱스의 모든 레코드에 잠금을 건다.

→ 스캔한 레코드에 잠금을 거는 것이기 때문에 WHERE 조건에 일치하지 않는 레코드도 잠금의 대상이 될 수 있음을 의미

→ 스토리지 엔진에 전달되는 조건들은 실제 SQL 문장의 조건들보다 적을 수도 (인덱스를 적절히 사용할 수 있는 조건만 전달하기 때문)

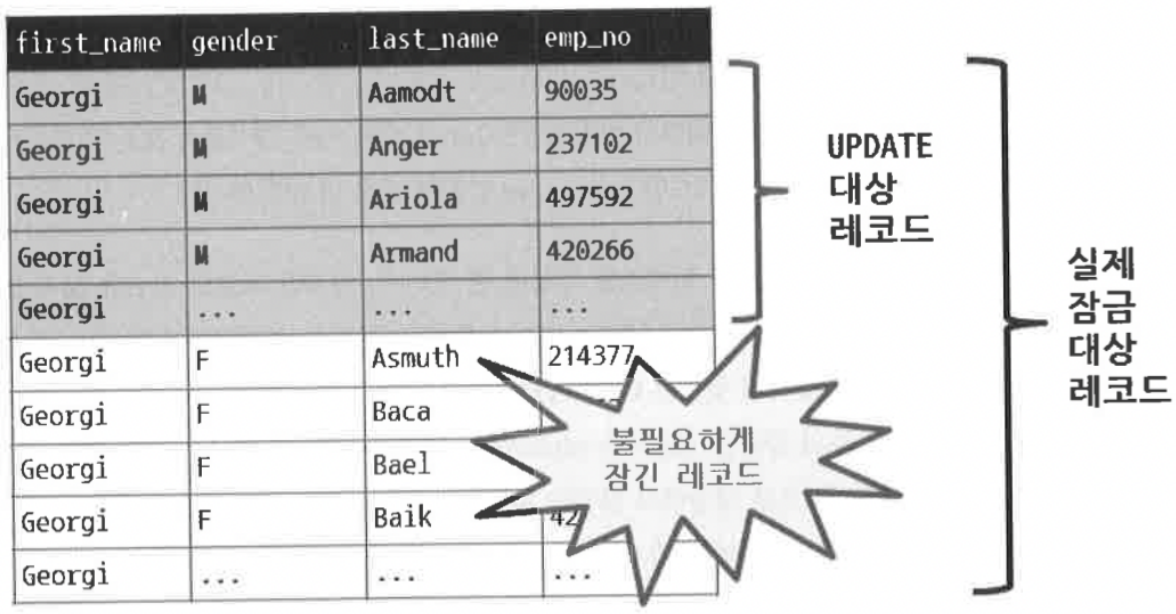
예제

```
UPDATE employees
SET last_name='...'
WHERE first_name='Georgi' AND gender='F';
```

이름이 조지, 젠더가 여성인 사원만 변경하는 쿼리

id	select_type	table	type	Key	key_len	Ref	rows	Extra
1	SIMPLE	employees	ref	ix_firstname	44	const	253	Using where

근데 인덱스는 firstname 만 걸려있음



[그림 12-2] UPDATE나 DELETE 쿼리에서 잠금 대상과 변경 대상 구분

- 레코드에 잠금을 거는 주체는 InnoDB 스토리지 엔진이고, 업데이트할 레코드를 최종 결정하는 주체는 MySQL 엔진임
- 인덱스가 firstname에만 걸려있기때문에 Georgi 사원 레코드들을 다 잠금 → 젠더가 남성인 애들까지 불필요하게 잠그게 되는 것,,,

⇒ 상당히 비 효율적이므로 꼭,,,, 튜닝 검토를 하자

인덱스가 아예 없으면 테이블의 모든 레코드에 잠금을 건다..

바이너리 로그

복제를 사용하지 않는 MySQL 서버는 binlog가 필요하지 않을 수 있고, binlog 기록하지 않으면 REPEATABLE-READ가 아니라 READ-COMMITTED로 사용할 수도 있음.

- READ-COMMITTED 일시, 인덱스와 관계 없이 **실제 변경되는 레코드만** 잠금을 걸게 된다
  - 인덱스를 이용해 검색하면서 인덱스 조건에 일치하는 레코드는 잠금 걸었다가, 나머지 조건에 일치하지 않아서 변경 대상이 아니면 다시 잠금을 해제하는 방식
  - 결국 인덱스를 최대한 사용할 수 있게 하자(격리 수준에 상관없이)