

16.3 MySQL에서 시퀀스 구현

🕒 생성일	@2021년 8월 28일 오후 5:00
☰ 태그	

MySQL에서는 일련번호 발급을 AUTO_INCREMENT로 한다

- AUTO_INCREMENT는 테이블의 일부라서 관리하기가 용이하지만,,
- 여러 테이블에 걸친 유일한 일련번호를 만들어 낼 수 없다

가끔 오라클 처럼 MySql도 특정 테이블에 의존적이지 않고 독립적으로 일련번호만 발급하는 기능이 필요할 때도 있다.

- 그래도 테이블 엔진 잘못 선택하면 부작용 유발 될 수 있음
- 마스터-슬레이브 간의 데이터 불일치 현상 발생 가능

현재 가능한 방법 : 테이블 생성 후, 그 테이블에 시퀀스의 현재 값을 유지하는 방법

1. 시퀀스용 테이블 준비

필요한 시퀀스 별로 별도의 테이블 만들어 관리할 수도 있지만, 여러개 시퀀스를 한 테이블에 넣어도 되긴함..

- 그래도 빈번히 읽는다면, 시퀀스 별로 테이블 분리 하자

```
CREATE TABLE mysql_sequences (  
    seq_name VARCHAR(10) NOT NULL,  
    seq_currval BIGINT UNSIGNED NOT NULL,  
    PRIMARY KEY (seq_name)  
) ENGINE=MyISAM;
```

- seq_name이 프라이머리 키로 정의되어야함
- 스토리지 엔진은 MyISAM
 - InnoDB는 시퀀스 테이블의 잠금 때문에 성능 저하를 유발 할 수 있다

(MyISAM은 잠금이 없나,,,? InnoDB 시퀀스 탈때 락 안걸지 않나요? 락 아니라 무슨 특별한 락이었던거 같은데,,,)

2. 시퀀스를 위한 스토어드 함수

위 테이블에서 일련번호 가져오려면 그냥 INSERT 문장 사용해도 되지만,,,

스토어드 함수로 캡슐화 해서 구현하기

위 테이블에 레코드를 INSERT해서 프라이머리 키에 중복 발생시, seq_currval 값만 1을 증가시켜서 다시 mysql_sequences테이블에 업데이트 하는것.(**UPSERT**)

```
INSERT INTO mysql_sequences
  SET seq_name='시퀀스이름', seq_currval=(@v_current_value:=1)
on duplicate KEY
  UPDATE seq_currval=(@v_current_value:=seq_currval+1);
```

하지만 매번 이렇게 하기에는,, 실수도 발생할 수 있으니, 스토어드 함수로 캡슐화 하자

```
DELIMITER ;;

CREATE FUNCTION nextval()
  RETURNS BIGINT UNSIGNED
  MODIFIES SQL DATA
  SQL SECURITY INVOKER
BEGIN
  INSERT INTO mysql_sequences
    SET seq_name='DEFAULT', seq_currval=(@v_current_value:=1)
  ON DUPLICATE KEY
    UPDATE seq_currval=(@v_current_value:=seq_currval+1);

  RETURN @v_current_value;
END ;;
```

nextVal() 이라는 함수로..

4. 시퀀스 사용시 주의 사항

1. InnoDB와 같이 트랜잭션을 지원하는 스토리지 엔진을 사용하면 안됨
 - 시퀀스 테이블은 많은 클라이언트로부터 동시에 READ될 수 있는데, 트랜잭션을 사용할 시, 한 클라이언트에서 시퀀스 번호를 가져오는 트랜잭션 끝나기 전까지 다른 클라이언트의 트랜잭션에서 가져갈 수 없음
2. 복제가 사용되는 MySQL 서버에서는 .. 시퀀스에서 일련번호를 읽음과 동시에 그 값을 다른 테이블에 INSERT해서는 안됨

```
INSERT INTO tb_article (article_id, ...) VALUES (nextval(), ... );
```

- 여러 트랜잭션에서 위 쿼리를 쓰는데, 먼저 nextVal() 사용한 곳에서 커밋을 치기 전에 다른 트랜잭션이 위 쿼리 실행해서 커밋 치면.... 바이너리 로그에 후자의 트랜잭션이 먼저 기록되서 슬레이브와 마스터의 데이터가 뒤바뀌게됨