

170922

BIGDATA ENGINEERING 2

EXERCISE

MySQL 엔진의 잠금

| 종류 | 특징 |
|------------------------|---|
| 글로벌 락 (Global Lock) | <ul style="list-style-type: none">- MySQL에서 제공하는 잠금 중 가장 범위가 크다.- 글로벌 락 획득 시 SELECT를 제외한 DDL, DML 대기 상태- 영향 범위 : MySQL 서버 전체- 모든 테이블에 영향을 미치므로 가급적 사용 자제 |
| 테이블 락 (Table Lock) | <ul style="list-style-type: none">- 개별 테이블 단위로 설정되는 잠금- 묵시적 테이블 락 : 명령으로 특정 테이블 락 획득- 명시적 테이블 락 : 데이터 변경되는 테이블에 자동적으로 락 설정, 변경 후 즉시 잠금 해제 |
| 유저 락 (User Lock) | <ul style="list-style-type: none">- 단순히 사용자가 지정한 문자열에 대해 락 획득/해제 |
| 네임 락 (Name Lock) | <ul style="list-style-type: none">- 데이터베이스 객체의 이름을 변경할 때 획득하는 잠금- 명시적으로 획득/해제하는 것이 아님- RENAME 명령어를 통해 변경하는 경우 자동 획득 |

MySQL 엔진의 잠금

| 종류 | 명령 | |
|------------------------|---|--|
| 글로벌 락 (Global Lock) | FLUSH TABLES WITH READ LOCK <ul style="list-style-type: none"> - 위 명령 실행 전에 잠금을 걸고 있는 SQL문이 있다면 먼저 실행된 SQL문이 완료되고 해당 트랜잭션이 완료될 때까지 대기 - FLUSH 명령어는 내부 캐시 메모리를 비우기 때문에 작업 중인 테이블의 작업 완료 후 실행되어야 함 | |
| 테이블 락 (Table Lock) | 명시적 테이블 락 | LOCK TABLES (table_name) [READ WRITE] UNLOCK TABLES |
| | 묵시적 테이블 락 | 쿼리가 실행되는 동안 자동으로 획득/해제 |
| 유저 락 (User Lock) | <ul style="list-style-type: none"> - SELECT GET_LOCK ('mylock', 2); : 락 획득 - SELECT IS_FREE_LOCK ('mylock'); : 락 확인 - SELECT RELEASE_LOCK ('mylock'); : 락 반환 | |
| 네임 락 (Name Lock) | <ul style="list-style-type: none"> - RENAME 명령을 2번로 나눠서 실행 시 짧은 시 테이블이 존재하지 않는 순간이 생기므로 에러 발생 RENAME TABLE class TO class_backup; RENAME TABLE student TO class; | |

참고 - MySQL 스토리지 엔진의 종류

| 종류 | 특징 |
|--------|---|
| InnoDB | <ul style="list-style-type: none">- 외래키, 제약조건, 동시성 제어, 트랜잭션 등 다양한 기능 지원으로 구조가 복잡함.- Row Level-Lock을 지원함. |
| MyISAM | <ul style="list-style-type: none">- 지원하는 기능이 많지 않아서 구조가 단순함.- Table Level-Lock을 지원함. |

예시 - 묵시적 테이블 락

Session 1

수행 시간이 오래 걸리는 조회 쿼리 실행
SELECT * FROM class WHERE sleep(100);

Session 2

Read Lock 상태 테이블에 데이터 조회
SELECT * FROM class;

실행 결과 - 정상 실행 확인

| | | | | | | | |
|---------|------|------|---|---|-------|------|----|
| C101-01 | C101 | 2014 | 1 | A | 92301 | 301호 | 40 |
| C102-01 | C102 | 2012 | 1 | A | 92001 | 209호 | 30 |
| C103-01 | C103 | 2012 | 1 | A | 92501 | 208호 | 30 |
| C103-02 | C103 | 2012 | 1 | B | 92301 | 301호 | 30 |
| C301-01 | C301 | 2012 | 2 | A | 92502 | 301호 | 30 |
| C302-01 | C302 | 2012 | 2 | A | 92501 | 209호 | 45 |
| C501-01 | C501 | 2012 | 1 | A | 92501 | 103호 | 45 |
| C501-02 | C501 | 2012 | 1 | B | 92502 | 204호 | 25 |
| C502-01 | C502 | 2012 | 2 | A | 92001 | 209호 | 30 |
| C502-02 | C502 | 2012 | 2 | B | 92301 | 103호 | 26 |

예시 - 목시적 테이블 락

Session 1

수행 시간이 오래 걸리는 조회 쿼리 실행

```
SELECT * FROM class WHERE sleep(100);
```

Session 2

Read Lock 상태 테이블에 데이터 변경

```
UPDATE class SET year = 2014 WHERE class_id='C101-01';
```

Session3

Read Lock 상태이고, Write Lock이 대기 상태에서 Select 수행

```
SELECT * FROM class LIMIT 10;
```

동작 확인

```
show processlist;
```

| | Id | User | Host | db | Command | Time | State | Info |
|---|-------|-------|----------------------|-------|---------|------|------------------------------|---|
| ▶ | 15197 | bde12 | 147.46.174.172:50661 | bde12 | Sleep | 74 | | NULL |
| | 15201 | bde12 | 147.46.174.172:51096 | bde12 | Sleep | 457 | | NULL |
| | 15202 | bde12 | 147.46.174.172:51097 | bde12 | Query | 74 | Waiting for table level lock | update class set year=2013 where class_id='C101-01' |
| | 15203 | bde12 | 147.46.174.172:51101 | bde12 | Sleep | 88 | | NULL |
| | 15204 | bde12 | 147.46.174.172:51102 | bde12 | Query | 67 | Waiting for table level lock | SELECT * FROM bde12.class LIMIT 0, 1000 |
| | 15217 | bde12 | 147.46.174.172:51672 | bde12 | Query | 79 | User sleep | select * from class where sleep(30) LIMIT 0, 1000 |

예시 – 네임 락

| Session 1 |
|--|
| ## 'my_lock'이라는 네임 락을 100초 동안 생성 SELECT GET_LOCK('mylock',100); |
| Session 2 |
| ## 다른 세션에서 my_lock을 확인 SELECT IS_FREE_LOCK('my_lock',100); |
| 동작 확인 |
| show processlist; |

예시 - 명시적 테이블 Read Lock

| Session 1 | |
|---|---|
| ## class 테이블에 Read Lock 설정 LOCK TABLES class READ; | |
| Session 1 | Session 2 |
| ## Read Lock 상태 테이블에 데이터 변경 후 결과 확인 update class set year=2014 where class_id='C101-01'; | ## 다른 session에서 Read Lock 테이블 조회 select * from class |
| Session 2 | |
| ## Lock 상태 해제 UNLOCK TABLES; | |

예시 - 명시적 테이블 Write Lock

| Session 1 | |
|--|---|
| ## class 테이블에 Write Lock 설정 LOCK TABLES class READ; | |
| Session 1 | Session 2 |
| ## Write Lock 상태 테이블에 데이터 변경 후 결과 확인 update class set year=2014 where class_id='C101-01'; | ## 다른 session에서 Write Lock 테이블 조회 select * from class; |
| | ## Process 상태 확인 show processlist |
| Session 2 | |
| ## Lock 상태 해제 UNLOCK TABLES; | |

Exercise 1

3가지 경우의 Locking 상황을 만들어 보고, show processlist; 명령어를 통해 Locking 상황을 확인해보라.

discipline@europa.snu.ac.kr로 제출

Exercise 2

아래 schedule 들에 적절한 lock/unlock 명령을 추가하여 2PL을 따르는 schedule로 나타내어라.

1)

| T_{15} | T_{16} |
|--------------|--------------|
| read(class) | read(class) |
| write(class) | write(class) |

2)

| T_{17} | T_{18} |
|-----------------------------|-----------------------------|
| read(class) write(class) | read(class) write(class) |
| write(class) | |

discipline@europa.snu.ac.kr로 제출

Exercise 2 – SQL Sample

| | | |
|----|---|--|
| 1) | T_{15} select * from course; update class set year=2014 where class_id='C101-01'; | T_{16} select * from course; update class set year=2014 where class_id='C101-01'; |
| 2) | T_{17} select * from course; update class set year=2014 where class_id='C101-01'; update class set year=2014 where class_id='C101-01'; | T_{18} select * from course; update class set year=2014 where class_id='C101-01'; |