CUBRID 트랜잭션 처리 원리

Date: 2015-07-01

(주)큐브리드 기술본부

목차

1. 트랜잭션이란?

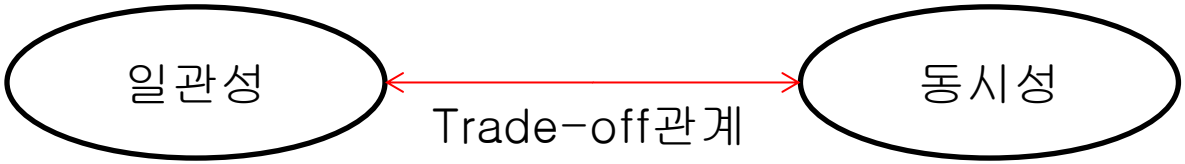
업무적으로 의미가 있는 최소 작업 단위

ex) 계좌이체

```
UPDATE ACCNT
SET AMOUNT = AMOUNT - 500
WHERE ACCNT_NO = 1
;

UPDATE ACCNT
SET AMOUNT = AMOUNT + 500
WHERE ACCNT_NO = 2
;
```

하나의 트랜잭션으로 수행(일관성)

- 동시성이란 여러 개의 트랜잭션이 동시에 수행될 수 있게 하는 성질!



일관성이 높아지면, 동시성은 떨어진다.
동시성이 높아지면, 일관성은 떨어진다.

CUBRID

2. 트랜잭션 처리

– 커밋과 롤백
커밋 : 하나의 작업에서 진행되는 여러 연산을 모두 반영
롤백 : 하나의 작업에서 진행되는 여러 연산을 모두 취소

– 트랜잭션 관리
잠금 보유자 : Lock을 획득한 보유자
잠금 대기자 : Lock을 획득하려고 대기 중 인자

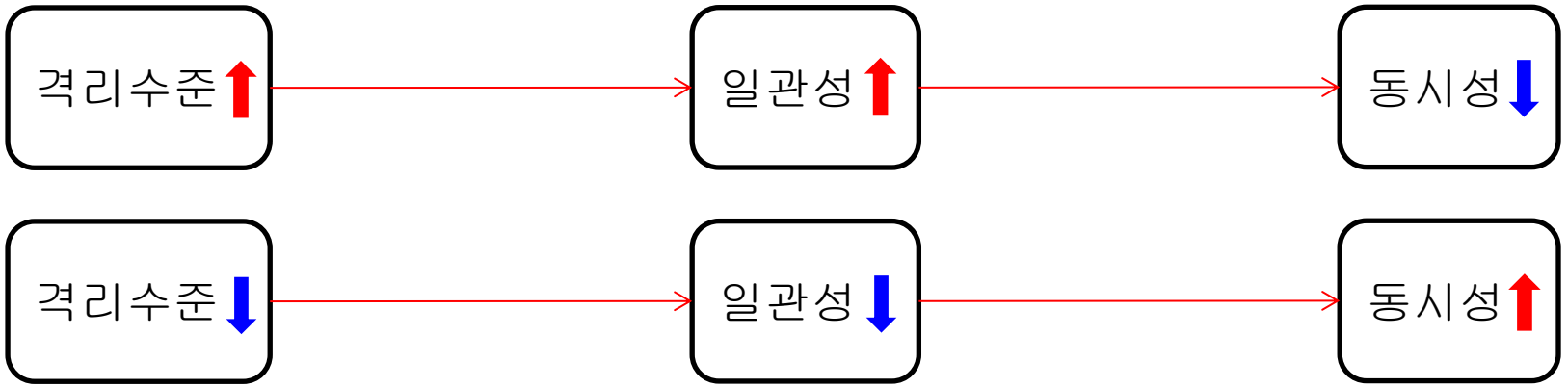
– 교착 상태 : 여러 개의 트랜잭션이 동시에 수행, 서로 상대방이 사용중인 데이터를 갱신하려고 대기중인 상태
–> 교착상태 발생시 CUBRID는 트랜잭션 중 가장 최근에 수행된 트랜잭션을 롤백 시켜 자동으로 해결한다.

– 잠금 대기 제한시간 : 잠금 대기가 무한정 지속되는 상황을 해결

CUBRID

3. 트랜잭션 격리 수준

- 트랜잭션 격리 수준이란 트랜잭션이 동시에 진행 중일때 "읽기 일관성"을 어느 선까지 보장할 것인지 명시하는 기준



- 읽기 일관성에 관련해 발생되는 상황

Dirty Read : Tx1이 아직 커밋 하지 않은 데이터를 Tx2가 그대로 읽음

Non-Repeatale Read : Tx1이 같은 데이터를 반복적으로 Select하는 사이 Tx2가 해당 데이터를 update 혹은 delete시킴

Phantom Read : Tx1이 특정 범위의 데이터를 반복해 읽는 사이 해당 조건을 만족하는 새로운 데이터를 Tx2가 Insert시킴

CUBRID

3. 트랜잭션 격리 수준

- CUBRID가 제공하는 격리성 수준

순번	격리 수준	설명	비고
1	Read Uncommited(3)	더티읽기,반복불가능한읽기,유령읽기 허용	Cubrid의 기본 설정임
2	Read Commited(4)	반복불가능한읽기,유령읽기 허용	Oracle의 기본 설정임
3	Repeatable Read(5)	유령읽기 허용	
4	Serializable(6)	전부 차단	일관성이 가장 높음

- Cubrid의 격리 수준 설정

cubrid.conf파일의 "isolation_level=숫자" 설정

특정 응용프로그램만 설정시에는 "set transaction isolation level 숫자" 명령으로 설정

CUBRID

4. 잠금관리

– Cubrid Lock의 종류(Cubrid 격리 수준 3)

순번	락 이름	잠금 보유자(Tx1)	잠금 대기자(Tx2)
1	공유 잠금 (S_LOCK)	읽기 중	읽기 가능, 쓰기 불가
2	베타 잠금 (X_LOCK)	갱신 중	**읽기 가능**, 쓰기 불가
3	갱신 잠금 (U_LOCK)	갱신, 삭제 대상 조회 중	**읽기 가능**, 쓰기 불가
4	의도 공유 잠금 (IS_LOCK)	S_LOCK 획득 시 Table에 IS_LOCK 획득	스키마 변경 불가, 모든 행 갱신 불가, 일부 행 갱신 가능, 모든 행 조회 가능
5	의도 베타 잠금 (IX_LOCK)	X_LOCK 획득 시 Table에 IX_LOCK 획득	스키마 변경 불가, 모든 행 갱신 불가, 일부 행 갱신 가능, 모든 행 조회 불가
6	공유 의도 베타 잠금(SIX_LOCK)	IS_LOCK+IX_LOCK 획득 시 SIX_LOCK 획득	스키마 변경 불가, 모든 행 갱신 불가, 일부 행 갱신 불가, 모든 행 조회 불가, 일부 행 조회 가능
7	다음 키 공유 잠금 (NS_LOCK)	삽입 중 자신의 키와 다음 키에 NS_LOCK 획득, 삽입 직후 다음 키에 NS_LOCK은 해제, 현재 키는 NS_LOCK유지	**읽기 가능, NS_LOCK이 걸려있으면 쓰기 불가**
8	다음 키 베타 잠금 (NX_LOCK)	갱신, 삭제 대상 조회 중 대상 건 까지는 X_LOCK, 다음 키 까지는 NX_LOCK 획득	**읽기 가능**, 쓰기 불가

CUBRID

4. 잠금관리

– Cubrid Lock의 종류(Cubrid 격리 수준 4)

순번	락 이름	잠금 보유자(Tx1)	잠금 대기자(Tx2)
1	공유 잠금 (S_LOCK)	읽기 중	읽기 가능, 쓰기 불가
2	베타 잠금 (X_LOCK)	갱신 중	**읽기 불가**, 쓰기 불가
3	갱신 잠금 (U_LOCK)	갱신, 삭제 대상 조회 중	**읽기 불가**, 쓰기 불가
4	의도 공유 잠금 (IS_LOCK)	S_LOCK 획득 시 Table에 IS_LOCK 획득	스키마 변경 불가, 모든 행 갱신 불가, 일부 행 갱신 가능, 모든 행 조회 가능
5	의도 베타 잠금 (IX_LOCK)	X_LOCK 획득 시 Table에 IX_LOCK 획득	스키마 변경 불가, 모든 행 갱신 불가, 일부 행 갱신 가능, 모든 행 조회 불가
6	공유 의도 베타 잠금(SIX_LOCK)	IS_LOCK+IX_LOCK 획득 시 SIX_LOCK 획득	스키마 변경 불가, 모든 행 갱신 불가, 일부 행 갱신 불가, 모든 행 조회 불가, 일부 행 조회 가능
7	다음 키 공유 잠금 (NS_LOCK)	삽입 중 자신의 키와 다음 키에 NS_LOCK 획득, 삽입 직후 다음 키에 NS_LOCK은 해제, 현재 키는 NS_LOCK유지	**NS_LOCK이 걸려있으면 읽기 불가, NS_LOCK이 걸려있으면 쓰기 불가**
8	다음 키 베타 잠금 (NX_LOCK)	갱신, 삭제 대상 조회 중 대상 건 까지는 X_LOCK, 다음 키 까지는 NX_LOCK 획득	**읽기 불가**, 쓰기 불가

CUBRID

5. 유령읽기란?

- 유령읽기(Cubrid 격리수준 3)

```
create table tbl1(id int primary key);
insert into tbl1 VALUES (1), (2), (3), (20);
```

시간	TX1	TX2
↓	SELECT * FROM TBL1;	
		INSERT INTO TBL1 VALUES(8);
		COMMIT;
	SELECT * FROM TBL1;	

유령 읽기 발생!

```
=== <Result of SELECT Command in Line 1> ===

            id
=============
             1
             2
             3
            20
             8
```


CUBRID

5. 유령읽기란?

- 유령읽기(Cubrid 격리수준 6)

```
create table tbl1(id int primary key);
insert into tbl1 VALUES (1), (2), (3), (20);
```

시간	TX1	TX2
↓	SELECT * FROM TBL1;	
		INSERT INTO TBL1 VALUES(8); 대기중!
	COMMIT;	대기풀림
	SELECT * FROM TBL1;	대기중!
	대기풀림	COMMIT;

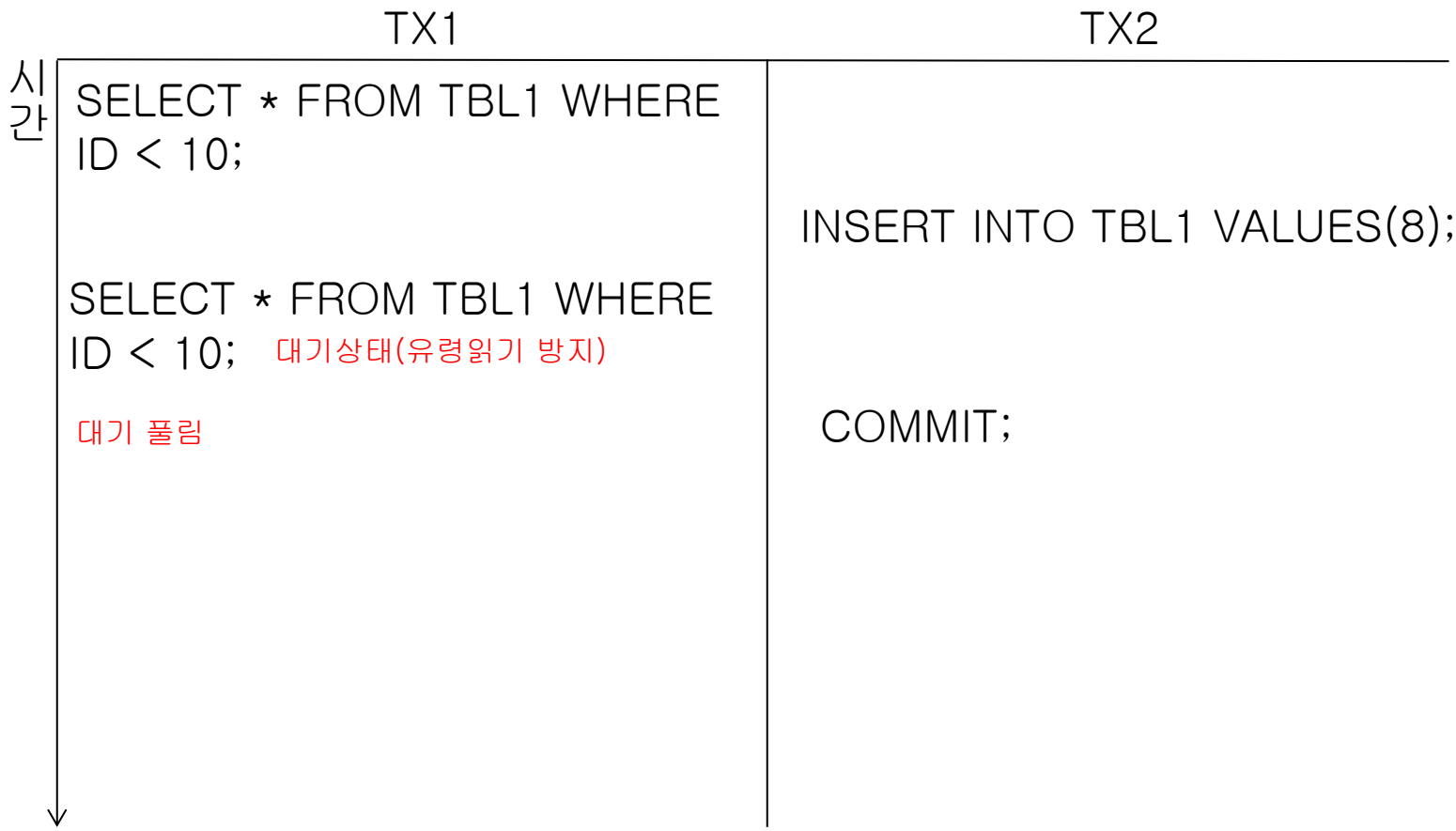
```
=== <Result of SELECT Command in Line 1> ===

            id
=============
             1
             2
             3
            20
             8
```


CUBRID

5. 유령읽기란?

- 유령읽기(Cubrid 격리수준 6)

```
create table tbl1(id int primary key);
insert into tbl1 VALUES (1), (2), (3), (20);
```



CUBRID

6. 스키마 잠금

- 스키마 안정 잠금 (SCH_S) : 질의 컴파일 수행 시 획득, 다른 트랜잭션이 해당 스키마 수정을 방지
- 스키마 수정 잠금 (SCH-M) : DDL 작업 수행 시 획득, 다른 트랜잭션이 수정된 스키마에 접근하는 것을 금지

7. 잠금 상태 확인

- "cubrid lockdb databasename" 명령어

  잠금 관련 시스템 파라미터 설정 값 확인

```
*** Lock Table Dump ***
 Lock Escalation at = 100000, Run Deadlock interval =  1.00
```

  Lock Escalation at : 락 에스컬레이션이 발생하는 조건, 행 잠금 개수와 키 잠금 개수의 합이 100,000개 초과시 행 수준 잠금이 테이블 수준 잠금으로 대체됨(테이블 잠금으로 대체되면 동시성이 떨어짐)

  Run Deadlock interval : 교착상태 트랜잭션이 있는지 탐지하는 주기, 1초마다 교착상태가 있는지 확인 후 둘 중 하나를 자동으로 Rollback 시킴

7. 잠금 상태 확인

- "cubrid lockdb databasename" 명령어

  현재 데이터베이스 서버에 접속중인 클라이언트 확인

```
Transaction (index  1, csql, DBA@newTest1|26128)
Isolation REPEATABLE CLASSES AND READ UNCOMMITTED INSTANCES
State TRAN_ACTIVE
Timeout_period : Infinite wait

Transaction (index  2, csql, DBA@newTest1|27770)
Isolation REPEATABLE CLASSES AND READ UNCOMMITTED INSTANCES
State TRAN_ACTIVE
Timeout_period : Infinite wait
```

Transaction : 트랜잭션 정보(ID, 프로그램, 계정@호스트|프로세스번호)
Isolation :해당 트랜잭션의 격리수준
State : 트랜잭션의 상태
Timeout_period : 잠금 대기 시간 제한 설정

7. 잠금 상태 확인

- "cubrid lockdb databasename" 명령어

객체 잠금 상태 확인

```
Object Lock Table:
        Current number of objects which are locked    = 7
        Maximum number of objects which can be locked = 10000

OID = -862|   660|-32509
Object type: Index key of class ( 0|    69|  29) = test1.
Index name: pk_test1_id
Total mode of holders =   NX_LOCK, Total mode of waiters =   NX_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  1
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NX_LOCK, Count =   1
LOCK WAITERS:
    Tran_index =   2, Blocked_mode =  NX_LOCK
                      Start_waiting_at = Fri Jul  3 17:53:03 2015
                      Wait_for_secs = -1.00
```

Current number of objects which are locked : 잠금 객체의 개수
OID : 볼륨번호|페이지번호|슬롯번호
Object type : 객체의 종류, test1은 테이블임
Index name : 인덱스 명
Total mode of holders : tbl1에 NX_LOCK이 잡혀있음
Total mode of waiters : tb1에 NX_LOCK을 얻고 싶은 상태
Num holders : 잠금 개수
Num blocked-holders : 차단된 잠금 개수(상위객체에 의해 차단된 잠금 개수, 예를 들어 X_LOCK보유해서 진짜로 갱신하려니 테이블 락이 걸려있는 상태)
Num waiters : 대기자 개수
Lock holder : 잠금 보유자 정보
Lock waiter : 잠금 대기자 정보

CUBRID

여기서 잠깐!

Cubrid	Oracle
update test1 set val=200 where id = 20 and val =20;	update test1 set val=200 where id = 20 and val =20;
drop table tbl1; DDL을 쳐도 Commit 하지 않음	drop table tbl1; DDL을 치는 순간 커밋!
rollback;	rollback;
select val from test1 where id = 20;	select val from test1 where id = 20;
결과 = 20	결과 = 200

CUBRID

7. 잠금 상태 확인(INSERT VS EQUAL SELECT)

- "cubrid lockdb databasename" 명령어
  잠금 상태 확인 실습(cubrid 격리수준 3)

```
create table tbl (id int primary key, a int);
insert into tbl values(10, 10);
insert into tbl values(20, 20);
insert into tbl values(40, 40);
```

TX1 | TX2

시간

insert into tbl values(30, 30);

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =  NS_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1

OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =  X_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  X_LOCK, Count =   1
```

→ id = 30,40에 대해 NS_LOCK 획득

→ id = 30에 대해 X_LOCK 획득

select * from tbl where id = 30;

```
=== <Result of SELECT Command in Line 1> ===

            id            a
=========================
            30           30

1 rows selected. (0.003401 sec)

1 command(s) successfully processed.
```


CUBRID

7. 잠금 상태 확인(INSERT VS EQUAL SELECT)

- "cubrid lockdb databasename" 명령어
  잠금 상태 확인 실습(cubrid 격리수준 4)

```
create table tbl (id int primary key, a int);
insert into tbl values(10, 10);
insert into tbl values(20, 20);
insert into tbl values(40, 40);
```

TX1 | TX2

시간

insert into tbl values(30, 30);

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =    NS_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1

OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =    X_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =   X_LOCK, Count =   1
```

→ id = 30,40에 대해 NS_LOCK 획득

→ id = 30에 대해 X_LOCK 획득

select * from tbl where id = 30;

```
OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =    X_LOCK, Total mode of waiters =    S_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  1
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =   X_LOCK, Count =   1
LOCK WAITERS:
    Tran_index =   2, Blocked_mode =   S_LOCK
                      Start_waiting_at = Wed Jul  1 18:19:28 2015
                      Wait_for_secs = -1.00
```

id = 30에 대해 S_LOCK을 얻고 싶지만 Tx1이 X_LOCK을 획득해서 대기중

CUBRID

7. 잠금 상태 확인(INSERT VS UPDATE)

- "cubrid lockdb databasename" 명령어
  잠금 상태 확인 실습 (cubrid 격리수준 3)

```
create table tbl (id int primary key, a int);
insert into tbl values(10, 10);
insert into tbl values(20, 20);
insert into tbl values(40, 40);
```

TX1 | TX2

시간 ↓

insert into tbl values(30, 30);

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =  NS_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1

OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =  X_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  X_LOCK, Count =   1
```

→ id = 30,40에 대해 NS_LOCK 획득

→ id = 30에 대해 X_LOCK 획득

update tbl set a=1000 where id = 30;

```
OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =  X_LOCK, Total mode of waiters =  U_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  1
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  X_LOCK, Count =   1
LOCK WAITERS:
    Tran_index =   2, Blocked_mode =  U_LOCK
                      Start_waiting_at = Wed Jul  1 18:25:20 2015
                      Wait_for_secs = -1.00
```

id = 30에 대해 U_LOCK을 얻고 싶지만 Tx1이 X_LOCK을 획득해서 대기중

CUBRID

7. 잠금 상태 확인(INSERT VS UPDATE)

- "cubrid lockdb databasename" 명령어
  잠금 상태 확인 실습 (cubrid 격리수준 4)

```
create table tbl (id int primary key, a int);
insert into tbl values(10, 10);
insert into tbl values(20, 20);
insert into tbl values(40, 40);
```

TX1 | TX2

시간

insert into tbl values(30, 30);

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =    NS_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1

OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =    X_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =   X_LOCK, Count =   1
```

→ id = 30,40에 대해 NS_LOCK 획득

→ id = 30에 대해 X_LOCK 획득

update tbl set a=1000 where id = 30;

```
OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =    X_LOCK, Total mode of waiters =  U_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  1
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =   X_LOCK, Count =   1
LOCK WAITERS:
    Tran_index =   2, Blocked_mode =   U_LOCK
                      Start_waiting_at = Wed Jul  1 18:25:20 2015
                      Wait_for_secs = -1.00
```

id = 30에 대해 U_LOCK을 얻고 싶지만 Tx1이 X_LOCK을 획득해서 대기중

CUBRID

7. 잠금 상태 확인(INSERT VS RANGE SELECT)

- "cubrid lockdb databasename" 명령어
  잠금 상태 확인 실습 (cubrid 격리수준 3)

```
create table tbl (id int primary key, a int);
insert into tbl values(10, 10);
insert into tbl values(20, 20);
insert into tbl values(40, 40);
```

TX1 | TX2

시간

insert into tbl values(30, 30);

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =  NS_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1

OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =  X_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  X_LOCK, Count =   1
```

→ id = 30,40에 대해 NS_LOCK 획득

→ id = 30에 대해 X_LOCK 획득

select * from tbl where id between 20 and 30;

```
csql> select * from tbl where id between 20 and 30;

=== <Result of SELECT Command in Line 1> ===

            id            a
=========================
            20           20
            30           30

2 rows selected. (0.003846 sec)

1 command(s) successfully processed.
```


CUBRID

7. 잠금 상태 확인 (INSERT VS RANGE SELECT)

- "cubrid lockdb databasename" 명령어
  잠금 상태 확인 실습 (cubrid 격리수준 4)

```
create table tbl (id int primary key, a int);
insert into tbl values(10, 10);
insert into tbl values(20, 20);
insert into tbl values(40, 40);
```

TX1 | TX2

시간

insert into tbl values(30, 30);

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =    NS_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1

OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =    X_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =   X_LOCK, Count =   1
```

→ id = 30,40에 대해 NS_LOCK 획득

→ id = 30에 대해 X_LOCK 획득

select * from tbl where id between 20 and 30;

```
OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =    X_LOCK, Total mode of waiters =    S_LOCK.
Num holders=  1, Num blocked holders=  0, Num waiters=  1
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =   X_LOCK, Count =   1
LOCK WAITERS:
    Tran_index =   2, Blocked_mode =   S_LOCK
                      Start_waiting_at = Wed Jul  1 18:27:19 2015
                      Wait_for_secs = -1.00
```

id = 30에 대해 S_LOCK을 얻고 싶지만 Tx1이 X_LOCK을 획득해서 대기중

CUBRID

7. 잠금 상태 확인(INSERT VS RANGE UPDATE)

- cubrid lockdb databasename 명령어

  잠금 상태 확인 실습 (cubrid 격리수준 3)

```
create table tbl (id int primary key, a int);
insert into tbl values(10, 10);
insert into tbl values(20, 20);
insert into tbl values(40, 40);
```

TX1 | TX2

시간

insert into tbl values(30, 30);

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =    NS_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1

OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =    X_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =   X_LOCK, Count =   1
```

→ id = 30,40에 대해 NS_LOCK 획득

→ id = 30에 대해 X_LOCK 획득

update tbl set a = 1000 where id between 20 and 25;

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =    NS_LOCK, Total mode of waiters =   NX_LOCK
Num holders=  1, Num blocked-holders=  0, Num waiters=  1
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1
LOCK WAITERS:
    Tran_index =   2, Blocked_mode =  NX_LOCK
                      Start_waiting_at = Wed Jul  1 18:30:13 2015
                      Wait_for_secs = -1.00
```

← id = 30에 대해 NX_LOCK을 얻고 싶지만 Tx1이 NS_LOCK을 획득해서 대기중

CUBRID

7. 잠금 상태 확인(INSERT VS RANGE UPDATE)

- cubrid lockdb databasename 명령어

  잠금 상태 확인 실습 (cubrid 격리수준 4)

```
create table tbl (id int primary key, a int);
insert into tbl values(10, 10);
insert into tbl values(20, 20);
insert into tbl values(40, 40);
```

TX1

TX2

시간

insert into tbl values(30, 30);

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =    NS_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1

OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =    X_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =   X_LOCK, Count =   1
```

→ id = 30,40에 대해 NS_LOCK 획득

→ id = 30에 대해 X_LOCK 획득

update tbl set a = 1000 where id between 20 and 25;

id = 30에 대해 NX_LOCK을 얻고 싶지만 Tx1이 NS_LOCK을 획득해서 대기중

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =    NS_LOCK, Total mode of waiters =   NX_LOCK
Num holders=  1, Num blocked-holders=  0, Num waiters=  1
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1
LOCK WAITERS:
    Tran_index =   2, Blocked_mode =  NX_LOCK
                      Start_waiting_at = Wed Jul  1 18:30:13 2015
                      Wait_for_secs = -1.00
```


CUBRID

8. 느린 질의 탐지

- cubrid tranlist 명령어

```
[leeko1984@newTest1]:~#cubrid tranlist leeko1984
Enter DBA password:
Tran index          User name      Host name      Process id    Program name             Query time    Tran time             Wait
for lock holder      SQL_ID       SQL Text
------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------
   1(ACTIVE)              DBA       newTest1           6689    broker1_cub_cas_1              0.00         0.00
         -1     *** empty ***
   3(ACTIVE)              DBA       newTest1          25857    query_editor_cub_cas_1         0.00         0.00
         -1     *** empty ***
   4(ACTIVE)              DBA       newTest1           7665    csql                           0.00        84.79
         -1     *** empty ***
   5(ACTIVE)              DBA       newTest1           7702    csql                          52.79        52.79
          4     4255c06da6593   update tbl set a = 1000 where i
------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------
```

Tran index : 트랜잭션ID(트랜잭션상태)
User name : DB사용자 이름
Host name : 장비명
Process id : 트랜잭션을 실행중인 프로세스id
Program name : 트랜잭션을 실행중인 프로세스명
Query time : 수행중인 SQL의 총 수행 시간
Tran time : 현재 트랜잭션의 총 수행시간
wait for lock holder : 해당 잠금을 보유중인 트랜잭션ID
SQL_ID : SQL Text에 대한 아이디
SQL Text : 수행중인 질의문

CUBRID

8. 느린 질의 탐지

- cubrid killtran 명령어

```
[leeko1984@newTest1]:~#cubrid killtran -i 5 leeko1984
Enter DBA password:
Ready to kill the following transactions:

Tran index          User name       Host name       Process id      Program name
------------------------------------------------------------------------------
   5(ACTIVE)              DBA       newTest1            7702              csql
------------------------------------------------------------------------------
Do you wish to proceed ? (Y/N)y
Killing transaction associated with transaction index 5
[leeko1984@newTest1]:~#
```

해당 트랜잭션을 kill시키면 아래와 같이 interrupted되면서 중지됨

```
csql> update tbl set a = 1000 where id between
csql> 20 and 25;

In the command from line 2,

ERROR: Has been interrupted.


0 command(s) successfully processed.
csql>
```


CUBRID

9. 교착상태

- 교착상태는 둘 이상의 트랜잭션이 서로 자신의 리소스를 놓지 않으면서 상대방의 리소스를 원 할때 발생한다.

- 교착상태 발생 가능성이 높은 상황

질의 수행이 느린 경우
잠금 대기 시간이 길어지는 경우
어플리케이션의 로직 문제
한 테이블에 인덱스가 많아 키 잠금이 많아지는 경우

- 교착상태 발생 가능성을 낮추기 위한 조치

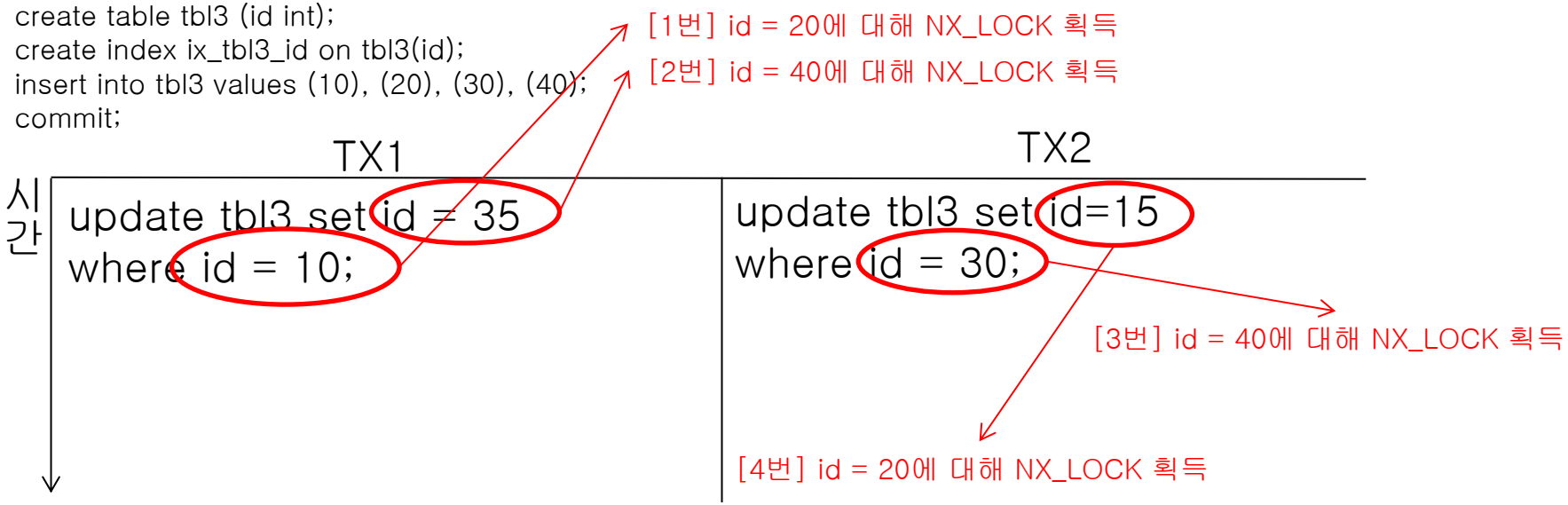
트랜잭션을 가급적 짧게 정의할 것
테이블 액세스 순서를 동일하게 할 것
DML질의인 경우 full table scan에 유의 할 것

CUBRID

9. 교착상태

- 교착상태 발생 케이스 (cubrid 격리수준 3,4)

```
create table tbl3 (id int);
create index ix_tbl3_id on tbl3(id);
insert into tbl3 values (10), (20), (30), (40);
commit;
```



[1번] -> [3번] -> [2번] -> [4번] 순서로 작업이 진행되면 교착상태 발생!

CUBRID

9. 교착상태

교착 상태 확인 실습(2개의 트랜잭션) (cubrid 격리수준 3,4)

```
create table tbl (id int primary key, a int);
insert into tbl values(10, 10);
insert into tbl values(20, 20);
insert into tbl values(40, 40);
```

TX1 TX2

시간

insert into tbl values(30, 30);

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =  NS_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1

OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =  X_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  X_LOCK, Count =   1
```

id = 30,40에 대해 NS_LOCK 획득

id = 30에 대해 X_LOCK 획득

update tbl set a = 1000 where id between 20 and 25;

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =  NS_LOCK, Total mode of waiters =  NX_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  1
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1
LOCK WAITERS:
    Tran_index =   2, Blocked_mode =  NX_LOCK
                      Start_waiting_at = Wed Jul  1 18:30:13 2015
                      Wait_for_secs = -1.00
```

id = 30에 대해 NX_LOCK을 얻고 싶지만 Tx1이 NS_LOCK을 획득해서 대기중

update tbl set a = 2000 where id = 20;

id = 20에 대해 U_LOCK을 얻고 싶지만 Tx2가 X_LOCK을 획득해서 대기

```
csql> update tbl set a = 2000 where id = 20;

In the command from line 1,

ERROR: Your transaction (index 4, DBA@newTest1|7665) has been unilaterally
aborted by the system.
```

교착상태 자동감지로 인해 트랜잭션이 자동 종료됨

CUBRID

9. 교착상태

교착 상태 확인 실습(3개의 트랜잭션) (cubrid 격리수준 3,4)

create table tbl (id int primary key, a int);
insert into tbl values(10, 10), (20, 20), (30,30);

TX1 | TX2 | TX3

시간

insert into tbl values(40, 40);

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =   NS_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1

OID =  1|   390|  43
Object type: Instance of class ( 0|    69|  24) = tbl.
Total mode of holders =   X_LOCK, Total mode of waiters = NULL_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  0
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  X_LOCK, Count =   1
```

id = 40에 대해 NX_LOCK을 얻고 싶지만 Tx1이 NS_LOCK을 획득해서 대기중

id = 40에 대해 NS_LOCK 획득
id = 40에 대해 X_LOCK 획득

update tbl set a=2000 where id = 30;

```
OID = -622|   390|-32469
Object type: Index key of class ( 0|    69|  24) = tbl.
Index name: pk_tbl_id
Total mode of holders =   NS_LOCK, Total mode of waiters =  NX_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  1
LOCK HOLDERS:
    Tran_index =   1, Granted_mode =  NS_LOCK, Count =   1
LOCK WAITERS:
    Tran_index =   2, Blocked_mode =  NX_LOCK
                      Start_waiting_at = Wed Jul  1 18:30:13 2015
                      Wait_for_secs = -1.00
```

update tbl set a=1000 where id=10;

```
In the command from line 1,

ERROR: Your transaction (index 6, DBA@newTest1|3974) has been unilaterally
aborted by the system.


0 command(s) successfully processed.
```

교착상태 자동감지로 인해 트랜잭션이 자동 종료됨

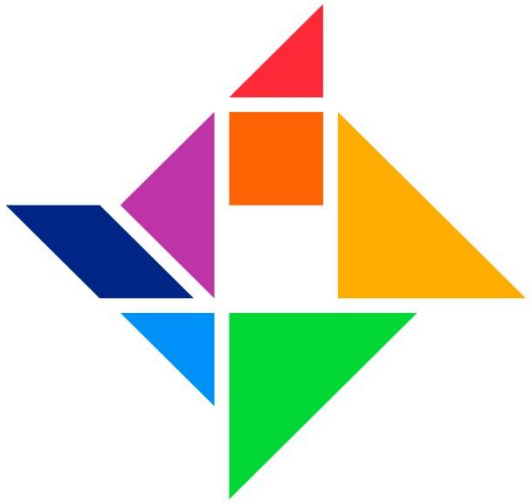
id = 30 에 대해 tx2가 U_LOCK을 잡고 있어서 U_LOCK획득 대기중

id = 20에 대해 NX_LOCK을 얻고 싶지만 Tx3가 NX_LOCK을 획득해서 대기

update tbl set a=2500
where id between 20 and 30;

```
OID =  1|   390|  45
Object type: Instance of class ( 0|    69|  27) = tbl.
Total mode of holders =    U_LOCK, Total mode of waiters =   U_LOCK.
Num holders=  1, Num blocked-holders=  0, Num waiters=  1
LOCK HOLDERS:
    Tran_index =   5, Granted_mode =   U_LOCK, Count =   1
LOCK WAITERS:
    Tran_index =   6, Blocked_mode =   U_LOCK
                      Start_waiting_at = Thu Jul  2 13:36:07 2015
                      Wait_for_secs = -1.00
```


CUBRID



CUBRID™