

SWE3003-41: Database Homework #3

Deadline: Dec. 8 11:59PM, 2021 (firm deadline, no late submission is allowed)

Submission: icampus; (submit in **PDF form**)

- ch16.example.txt에서 "deadlock" 부분 실습을 진행한 후, Oracle이 deadlock 상황을 어떻게 검사하고 이를 해결하는지 설명하시오. (검사 및 해결 각 1점)

wait-for-graph를 이용하여 deadlock을 detect한다.

자동적으로 rollback을 시켜서 하나의 statement에 대해 이전의 상태로 되돌려 놓는다.

```

SQL> create table account (id number, balance number);
테이블이 생성되었습니다.
SQL> insert into account values (1, 100);
1 개의 행이 만들어졌습니다.
SQL> insert into account values (2, 200);
1 개의 행이 만들어졌습니다.
SQL> commit;
커밋이 완료되었습니다.
SQL> update account set balance = balance + 10
  2 where id = 1;
1 행이 업데이트되었습니다.
SQL> update account set balance = balance + 10
  2 where id = 2;
update account set balance = balance + 10
ORA-00060: 자원 대기중 교착 상태가 감출되었습니다.
SQL> select * from account;
   ID  BALANCE
--  --
    1      110
    2      200
SQL>
  
```

```

SQL*Plus: Release 19.0.0.0.0 - Production on 금 12월 3 07:51:00 2021
Version 19.3.0.0.0
Copyright (c) 1982, 2019, Oracle. All rights reserved.
사용자명 입력: conn /as sysdba
다음을 입력하십시오:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
SQL> update account set balance = balance + 1,1
SP2-0734: "update s, "(으)로 시작되는 할 수 없는 명령 - 나머지 줄은 무시되었습니다.
SQL> update account set balance = balance + 1,1
  2 where id = 2;
1 행이 업데이트되었습니다.
SQL> update account set balance = balance + 1,1
  2 where id = 1;
  
```

- 18장에서 설명한 WAL (write-ahead logging) 프로토콜 관련 아래 답하시오(각 1점).

a) WAL 프로토콜이란?

Undo recovery 를 위해 존재하는 프로토콜로, 어떤 page 를 buffer 에서 disk 에 쓰기 전에 log buffer 에서 undo 할 수 있는 log 를 log device 에다가 쓰는 프로토콜이다.

b) WAL 프로토콜을 위배하면 어떤 문제가 있나?

Page를 disk에 쓴 상태에서 undo할 수 있는 log가 log device에 없으면, System crash가 날 경우 undo를 할 수 있는 방법이 없다.

c) WAL 프로토콜이 커밋되지 않는 트랜잭션에 대해 failure시 어떻게 atomicity를 보장하는지 설명하시오.

System 이 crash 가 날 경우, 만약 어떤 page 가 disk 에 쓰였다면, device 의 undo log 를 확인해 원래의 page 상태로 되돌려준다.

d) 비휘발성 DRAM (Non-volatile DRAM)이 있다고 가정하고, undo log를 NV-DRAM에

저장하는 경우에도 WAL 프로토콜이 필요한가? 이를 정당화하시오.

WAL을 하는 이유는 system crash가 날 경우, log buffer에 있는 data가 날라가기 때문에, log device에 써주는 것이다. 따라서 DRAM이 non-volatile인 경우에는 undo log가 날라가지 않아 system crash가 난 경우에도 undo를 할 수 있어서 WAL 프로토콜은 필요가 없다.

3. 이론 강의 16.4절, 16.6.3절에 다룬 single version 기반의 DBMS에서 아래 두 트랜잭션이 동시에 SQL을 수행할 때 (T1, T2세션에서 각 SQL 문은 아래 표에 나타난 순서대로 이슈된다고 가정), T2가 read committed로 동작할 때, DBMS에 의해 실제 스켈줄링은 어떻게 되는지 설명하고(3점), T1, T2 수행이 다 끝난 뒤, account 1의 balance 값(1점)은 얼마인가?

create table account (id number, balance number);

insert into account values (1, 100);

commit;

T1	T2 (read committed mode)
2. update account set balance = balance + 10 where id = 1; 6. commit;	1. Select * from account where id = 1; 3. Select * from account where id = 1; 4. Update account set balance = balance + 10 where id = 1; 5. Commit;

- T2에서는 ID = 1, balance 는 100 인 column 하나가 출력된다.
 - 2, 3. ID=1 에 해당하는 balance 를 10 더하면 ID = 1, balance 는 110 이 되고, single version 이고 read committed mode 이기 때문에 R-W Conflicts 를 지원하지 않아 T2 도 T1 과 마찬가지로 ID=1 인 곳에 balance 110 이 쓰이게 된다.
 4. T1 이 ID=1 의 balance 를 바꾸었으므로 lock 이걸려 진행이 되지 않는다.
 6. T1 의 6.인 commit 이 먼저 실행되면서 lock 을 풀고, 이때 T1 에서는 ID = 1 의 balance 는 110 으로 보인다.
 5. T1 이 commit 되고 lock 이 풀리면서 T2 의 5.인 commit 이 실행될 수 있게 된다. 이때의 ID=1 balance 는 120 이되게 된다.
- 따라서 T1, T2 가 다 끝나고 나서의 balance 값은 120 이 된다.