# DataBase Assignment

hw#7 - practice #6

정보대학 컴퓨터학과 2017320108 고재영

hw7이란 데이터 베이스에 해당 과제의 bank 관련 테이블 setup.

```
hw7=# insert into employee values('Loreena', 'Downtown', 1300);
INSERT 0 1
hw7=# insert into employee values('Williams', 'Downtown', 2500);
INSERT 0 1
hw7=# insert into employee values('Green', 'Central', 1500);
hw7=# insert into employee values('Sato', 'Central', 1600);
INSERT 0 1
hw7=# \d
        릴레이션(relation) 목록
| 이름 | 종류 |
 스키마
                                 소유주
public
          account
                                postgres
 public
                       테이
          borrower
                                postgres
                       EİİOİ
 public
          branch
                                postgres
                       테이
public
          customer
                                postgres
                                postgres
public
          depositor
                       테이
public
          employee
                                postgres
public |
          Loan
                                postgres
(7개 행)
```

<Exercise 1> Make transactions for the following queries using SQL expressions 1.

Begin;

insert into account values ('A-125', 'Redwood', 650);

insert into depositor values ('Green', 'A-125');

Commit;

```
hw7=# Begin;
BEGIN
hw7=# insert into account values ('A-125', 'Redwood', 650);
INSERT 0 1
hw7=# insert into depositor values ('Green', 'A-125');
INSERT 0 1
hw7=# Commit;
COMMIT
```

```
hw7=# select ∗
hw7-# from account;
account_number | branch_name | balance
A-101
                   Downtown
                                      500
A-215
                                      700
                   Mianus
A-102
                   Perryridge
                                      400
                                      350
A-305
                   Round Hill
                                      900
A-201
                   Perryridge
A-222
                   Redwood
                                       700
A-217
                   Brighton
                                      750
A-333
                                      850
                   Central
A-444
                   North Town
                                      625
A-125
                   Redwood
                                      650
(10개 행)
hw7=# select ∗
nw7-# from depositor;
                  account_number
customer name
Johnson
                  A-101
                  A-215
Smith
                  A-102
Hayes
                  A-101
Haves
                  A - 305
Turner
                  A-201
Johnson
Jones
                  A-217
                  A-222
Lindsay
                  A-333
Majeris
Smith
                  A-444
                  A-125
Green
(11개 행)
```

```
2.

Begin;

delete from depositor

where customer_name = 'Green';

delete from account

where account_number = 'A-125';

Commit;
```

```
hw7=# Begin;
BEGIN
hw7=# delete from depositor
hw7-# where customer_name = 'Green';
DELETE 1
hw7=# delete from account
hw7-# where account_number = 'A-125';
DELETE 1
hw7=# Commit;
COMMIT
```

```
hw7=# select *
hw7-# from account;
                   branch_name | balance
 account number
 A-101
                   Downtown
                                       500
 A-215
                                       700
                   Mianus
 A-102
                   Perryridge
                                       400
                   Round Hill
 A-305
                                       350
                                       900
                   Perryridge
 A-201
 A-222
                                       700
                   Redwood
                                       750
 A-217
                   Brighton
 A-333
                   Central
                                       850
 A-444
                   North Town
                                       625
(9개 행)
hw7=# select *
hw7-# from depositor;
 customer name
                  account number
                  A-101
 Johnson
 Smith
                  A-215
 Hayes
                  A-102
                  A-101
 Haves
                  A - 305
 Turner
 Johnson
                  A-201
 Jones
                  A-217
 Lindsay
                  A-333
 Majeris
 Smith
                  A-444
(10개 행)
```

3.

```
Begin;

update account

set balance = balance - 200

where account_number = 'A-101';

update branch
```

```
set assets = assets - 200
where branch_name in (select branch_name
from account
where account_number = 'A-101');
update account
set balance = balance + 200
where account_number = 'A-444';
update branch
set assets = assets + 200
where branch_name in (select branch_name
from account
where account_number = 'A-444');
Commit;
```

```
hw7=# Begin;
BEGIN
hw7=# update account
hw7-# set balance = balance - 200
hw7-# where account number = 'A-101';
UPDATE 1
hw7=# update branch
hw7-# set assets = assets - 200
hw7-# where branch name in (select branch name
hw7(# from account
hw7(# where account number = 'A-101');
UPDATE 1
hw7=# update account
hw7-# set balance = balance + 200
hw7-# where account number = 'A-444';
UPDATE 1
hw7=# update branch
hw7-# set assets = assets + 200
hw7-# where branch_name in (select branch_name
hw7(# from account
hw7(# where account_number = 'A-444');
UPDATE 1
hw7=# Commit;
COMMIT
```

```
nw7=# select ∗
nw7-# from account;
                   branch_name
                                  balance
account number |
A-215
                   Mi anus
                                      700
A-102
                   Perryridge
                                      400
A - 305
                   Round Hill
                                      350
A-201
                   Perryridge
                                      900
A-222
                                       700
                   Redwood
                   Brighton
A-217
                                      750
A-333
                                      850
                   Central
                                      300
A-101
                   Downtown
A-444
                   North Town
                                      825
(9개 행)
nw7=# select ⋆
nw7-# from branch;
              | branch_city |
branch name
                               assets
Redwood
               Palo Alto
                               2100000
Perryridge
                               1700000
               Horseneck
Mi anus
                                400200
               Horseneck
               Horseneck
Round Hill
                               8000000
Pownal
                                400000
               Bennington
               Brooklyn
Brighton
                               7000000
               Rye
                                400280
Central
               Brooklyn
Downtown
                                899800
North Town
               Rve
                               3700200
'9개 행)
```

<Exercise 2> Using the transaction operations, execute following queries on PostgreSQL 1.

\*\*위의 트랜잭션을 수행하기 전에, 1번 exercise의 (3)번 문제에서 값을 조작했기에 johnson이 가지고 있는 두 계좌 중, 하나의 계좌는 500보다 적은 balance를 가진

상태입니다. 따라서, 500을 인출할 계좌는 500보다 많이 가지고 있는 계좌로 선정하도록 하겠습니다.

```
hw7=# Begin;
BEGIN
hw7=# update account
hw7-# set balance = balance - 500
hw7-# where balance > 500 and
hw7-# account_number in (select account_number
hw7(# from depositor
hw7(# where customer_name = 'Johnson');
UPDATE 1
hw7=# update account
hw7-# set balance = balance + 500
hw7-# where account_number in (select account_number
hw7(# from depositor
hw7(# where customer_name = 'Turner');
UPDATE 1
hw7=# Commit;
COMMIT
```

hw7=# select * from account;		
account_number		balance
A-215 A-102 A-222 A-217 A-333 A-101 A-444 A-201 A-305 (9기 행)	•	700 400 700 750 850 300 825 400 850
Johnson Smith Hayes Hayes Turner Johnson Jones Lindsay Majeris Smith (10개 행)	A-101 A-215 A-102 A-101 A-305 A-201 A-217 A-222 A-333 A-444	

실행 결과로, johnson의 두 계좌 중, A-101에는 balance가 300만 들어있기 때문에 A-201에서 500을 인출하여 Turner의 A-305 계좌에 전달되었습니다.

# 2.

# Begin;

Increase all balance by 500 in the account table, and make a system crash before committing

You can make a crash by just closing the window

Then, open PostgreSQL again, and check whether the update is applied

```
hw7=# Begin;
BEGIN
hw7=# update account
hw7-# set balance = balance + 500;
UPDATE 9
hw7=# _
```

commit으로 정상적인 종료를 하지 않고 crash시키겠습니다.

# SQL Shell (psql)

```
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
postgres 사용자의 암호:
psql (11.5)
도움말을 보려면 "help"를 입력하십시오.
postgres=# ₩c hw7
접속정보: 데이터베이스="hw7", 사용자="postgres".
hw7=# select * from account;
 account number | branch name | balance
 A-215
                  Mianus
                                    700
 A-102
                  Perryridge
                                    400
 A-222
                  Redwood
                                    700
                                    750
 A-217
                  Brighton
                  Central
                                    850
 A-333
                  Downtown
                                    300
 A-101
                  North Town
 A-444
                                    825
A-201
                  Perryridge
                                    400
 A-305
                 Round Hill
                                    850
(9개 행)
hw7=# _
```

재실행을 시켰더니 500이 더해진 업데이트는 이루어지지 않았습니다.

<Exercise 3>

Open two PostgreSQL windows simultaneously.

In the window A, begin a transaction with the isolation level Read Committed.

# 1. Dirty read

In the window B, begin a transaction and increase A 215's balances by 1000 WITHOUT Commit.

Is the update in the window B applied to the transaction of the window A? If yes, why? If no, why?

SQL Shell (psql)

```
Server [localhost]:
Database [postgres]:
Port [5432]:
Jsername [postgres]:
Dostgres 사용자의 암호:
Dostgres 사용자의 암호:
Dostgres=# #c hw7
도움말을 보려면 "help"를 입력하십시오.
Dostgres=# #c hw7
접속정보: 데이터베이스="hw7", 사용자="postgres".
hw7=# Begin;
BEGIN
hw7=# update account
hw7-# set balance = balance + 1000
hw7-# set balance = balance + 'A-215';
JPDATE 1
hw7=# _
```

# SQL Shell (psql)

```
Server [localhost]:
Database [postgres]:
Port [5432]:
Jsername [postgres]:
postgres 사용자의 암호:
osal (11.5)
도움말을 보려면 "help"를 입력하십시오.
oostgres=# ₩c hw7
접속정보: 데이터베이스="hw7", 사용자="postgres".
nw7=# Begin;
BEGIN
nw7=# set transaction isolation level read committed;
SET
nw7=# select * from account;
account number | branch name | balance
A-215
                 Mianus
                                    700
A-102
                 Perryridge
                                    400
                 Redwood
A-222
                                    700
                 Brighton
                                    750
A-217
A-333
                 Central
                                    850
A-101
                 Downtown
                                    300
A - 4444
                 North Town
                                    825
                 Perryridge
A-201
                                    400
A - 305
                 Round Hill
                                    850
(9개 행)
```

업데이트가 전혀 적용되지 않았다. isolation level이 read committed한 상태에서는 dirty read의 발생을 막아주기 때문이다.

## 2. Non repeatable read

In the window A, Check the A 201's balance using an appropriate SQL query and, In the window B, begin a new transaction and increase A 201's balance by 1000 WITH

#### Commit.

Is the update applied to the A 201's balance in the transaction of the window A? If yes, why? If no, why?

```
SQL Shell (psql)
                                                           SQL Shell (psql)
                                                          Server [localhost]:
                                                         Database [postgres]:
Port [5432]:
Username [postgres]:
hw7=# select balance
hw7-# from account
hw7-# where account_number = 'A-201';
                                                         postgres 사용자의 암호:
psql (11.5)
도움말을 보려면 "help"를 입력하십시오.
 balance
                                                         (1개 행)
hw7=# select *
hw7-# from account
hw7-# where account_number = 'A-201';
                                                         hw7=# update account
hw7-# set balance = balance + 1000
hw7-# where account_number = 'A-201';
UPDATE 1
 account_number
                     | branch name | balance
A-201
(1개 행)
                       Perryridge
                                                400
                                                          hw7=# Commit;
                                                         hw7=#
hw7=# select *
hw7-# from account
hw7-# where account_number = 'A-201';
 account_number | branch_name |
                                               1400
 A-201
                       Perryridge
(1개 행)
hw7=#
```

업데이트가 적용되었다. isolation level이 read committed한 상태에서 non repeatable read가 발생 가능하기 때문이다. 무엇보다 commit으로 종료되었기 때문에, 해당 업데이트된 값을 받는다.

## 3. Phantom read

In the window A, Check all tuples in the Account relation using an appropriate SQL query and, In the window B, begin a transaction and insert an account into account relation with values ('A 999', ' Anam ', 1000)

Is the insertion applied to the Account relation in the transaction of the window A? If yes, why? If no, why?

```
SQL Shell (psql)
                                                                        SQL Shell (psql)
nw7-# from accoun
| account_number
                                                                      Server [localhost]:
Database [postgres]:
Port [5432]:
                            branch_name
                                                    balance
                                                         700
400
700
750
850
300
825
850
1400
                             Mianus
                            Perryridge
Redwood
Brighton
                                                                      Username [postgres]:
postgres 사용자의 암호:
psql (11.5)
도움말을 보려면 "help"를 입력하십시오.
 A-222
A-217
A-333
                             Central
    101
                             Downtown
                                                                      postgres=# ₩c hw7
접속정보: 데이터베이스="hw7", 사용자="postgres".
hw7=# Begin;
   -444
                            North Town
 A-305
A-201
                             Round Hill
                            Perryridge
                                                                      BEGIN
hw7=# update account
hw7-# set balance = balance + 1000
hw7-# where account_number = 'A-201';
9개 행)
nw7=# select *
nw7-# from account;
                                                                       UPDATE 1
                                                                      hw7=# Commit;
COMMIT
 account_number
                            branch_name | balance
A-215
A-102
A-222
A-217
A-333
                                                                      hw7=# Begin;
                             Mianus
                                                          700
400
700
750
850
300
                                                                      BEGIN
hw7=# insert into account values ('A-999', 'Anam', 1000);
INSERT 0 1
hw7=# Commit;
COMMIT
                             Perryridge
                            Redwood
                             Brighton
                             Central
    101
                             Downtown
                                                         825
850
1400
                            North Town
Round Hill
    444
                                                                      hw7=# _
    305
   -201
                             Perryridge
 4-999
                             Anam
                                                         1000
 10개 행)
```

A-999 / Anam / 1000인 튜플이 Phantom하게 들어왔다. isolation level이 read committed한 상태에서 phantom read가 발생 가능하기 때문이다.

## <Exercise 4>

Make two transactions with the isolation level Serializable.

Open two PostgreSQL windows simultaneously.

In the window A, begin a transaction with the isolation level Serializable.

=> exercise 3와의 원활한 비교를 위해, exercise 3의 초기 상태와 동일하게 시작할 것.

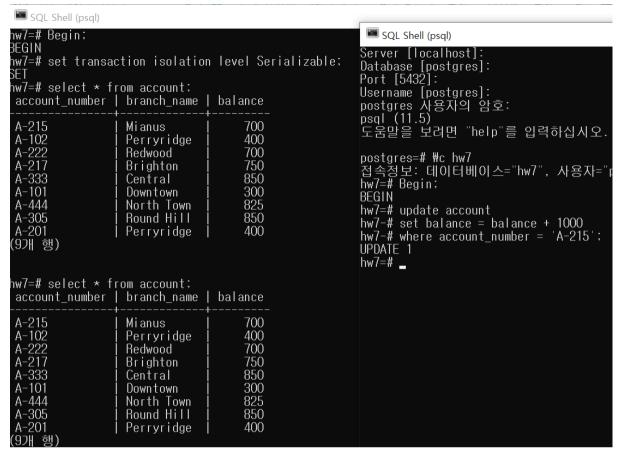
```
hw7=# Begin;
BEGIN
hw7=# update account
hw7-# set balance = balance - 1000
hw7-# where account_number = 'A-201';
UPDATE 1
hw7=# Commit;
COMMIT
hw7=# Begin;
BEGIN
hw7=# delete from account
hw7-# where account_number = 'A-999';
DELFTE 1
hw7=# Commit;
COMMIT
hw7=# select * from account;
                                  balance
 account number | branch name
A-215
                                      700
                   Mianus
A-102
                                      400
                   Perryridge
                                       700
                   Redwood
A-222
                   Brighton
                                       750
A-217
A-333
                   Central
                                      850
                   Downtown
A-101
                                      300
                                      825
A-444
                   North Town
                   Round Hill
                                      850
A - 305
                   Perryridge
 A-201
                                      400
```

동일한 초기화 상태로 만들어줍니다.

# 1. Dirty read

In the window B, begin a transaction and increase A 215's balances by 1000 WITHOUT Commit.

Is the update in the window B applied to the transaction of the window A? If yes, why? If no, why?



update가 적용되지 않는다. dirty read를 막아주기 때문이다.

## 2. Non repeatable read

In the window A, Check the A 201's balance using an appropriate SQL query and, In the window B, begin a new transaction and increase A 201's balance by 1000 WITH Commit.

Is the update applied to the A 201's balance in the transaction of the window A? If yes, why? If no, why?

```
SQL Shell (psql)
                                                             SQL Shell (psql)
                                                            Server [localhost]:
                                                750
850
300
825
850
                        Brighton
                                                            Database [postgres]:
Port [5432]:
                        Central
   101
                        Downtown
                                                           Username [postgres]:
postgres 사용자의 암호:
psql (11.5)
도움말을 보려면 "help"를 입력하십시오.
   444
                        North Town
                        Round Hill
   -305
                                                400
   -201
                        Perryridge
                                                           postgres=# ₩c hw7
접속정보: 데이터베이스="hw7", 사용자="postgres".
hw7=# Begin;
BEGIN
nw7=# Begin;
경고: 이미 트랜잭션 작업이 진행 중입니다
경고:
3EGIN
                                                           hw7=# update account
hw7-# set balance = balance + 1000
hw7-# where account_number = 'A-215';
UPDATE 1
nw7=# select ∗
nw7-# from account
nw7-# where account_number = 'A-201';
account number | branch name |
                                          balance
                                                           hw7=# rollback
hw7-# ;
ROLLBACK
A-201
(1개 행)
                     | Perryridge |
                                                400
                                                            hw7=# Begin;
                                                            BEGIN
                                                            hw7=# update account
hw7-# set balance = balance + 1000
nw7=# select *
nw7-# from account
                                                            hw7-# where account_number = 'A-201';
nw7-# where account_number = 'A-201';
                                                            UPDATE 1
                                                           hw7=# Commit;
COMMIT
 account_number | branch_name
                                          balance
A-201
(1개 행)
                       Perryridge
                                                400
                                                            hw7=#
าพ7=#
```

update가 적용되지 않는다. isolation level이 serializable한 상태에서는 non repeatable read의 발생을 방지해주기 때문이다.

#### 3. Phantom read

In the window A, Check all tuples in the Account relation using an appropriate SQL query and, In the window B, begin a transaction and insert an account into account relation with values

('A 999', ' Anam ', 1000)

Is the insertion applied to the Account relation in the transaction of the window A? If yes, why? If no, why?

```
SQL Shell (psql)
                                                                                    SQL Shell (psql)
            경고:
BEGIN
hw7=# select *
hw7-# from account;
account_number | b
  A-215
A-102
A-222
                                                                      700
400
700
750
850
300
825
850
                                  Perryridge
Redwood
Brighton
Central
                                                                                 postgres=# #c hw7
접속정보: 데이터베이스="hw7", 사용자="postgres".
hw7=# Begin:
BEGIN
hw7-# insert into account values ('A-999', 'Anam', 1000);
INSERT 0 1
bw7-# Commission
     -222
-217
-333
                                   Downtown
                                  North Town
Round Hill
Perryridge
     -305
                                                                      400
                                                                                 hw7=# Commit;
COMMIT
  9개 행)
                                                                                  hw7=#
hw7=# select *
hw7-# from account;
account_number | b
                                   branch name | balance
                                  Mianus
Perryridge
Redwood
Brighton
Central
 A-215
A-102
A-222
A-217
A-333
                                                                      700
400
700
750
850
300
825
850
400
                                  Downtown
North Town
Round Hill
Perryridge
     -101
-444
      305
      201
```

update가 적용되지 않는다. isolation level이 serializable한 상태에서는 phantom read의 발생을 방지해주기 때문이다.