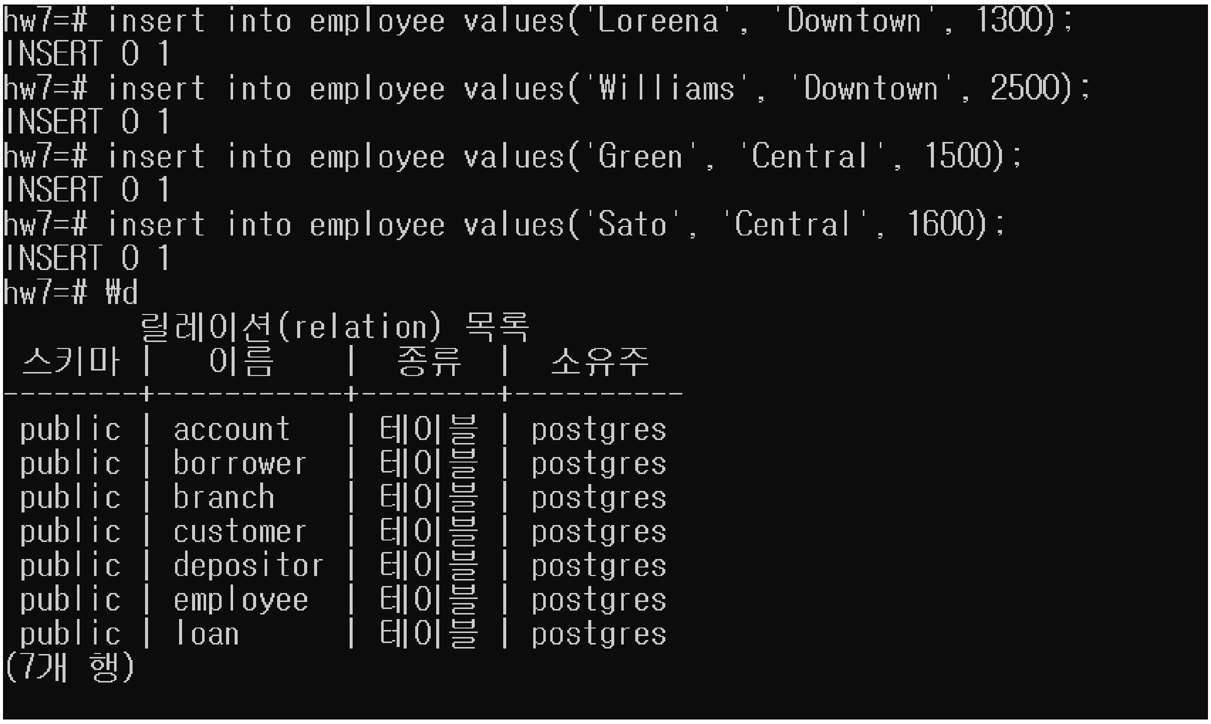
DataBase Assignment

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);
INSERT 0 1
hw7=# insert into employee values('Sato', 'Central', 1600);
INSERT 0 1
hw7=# ₩d
       릴레이션(relation) 목록
 스키마 |   이름    |  종류  |  소유주
--------+-----------+--------+----------
 public | account   | 테이블 | postgres
 public | borrower  | 테이블 | postgres
 public | branch    | 테이블 | postgres
 public | customer  | 테이블 | postgres
 public | depositor | 테이블 | postgres
 public | employee  | 테이블 | postgres
 public | loan      | 테이블 | postgres
(7개 행)
```

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

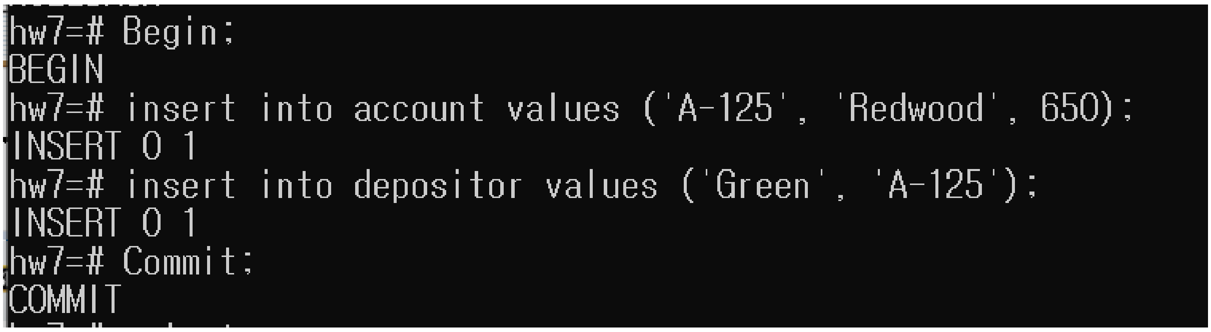
1.

Begin;

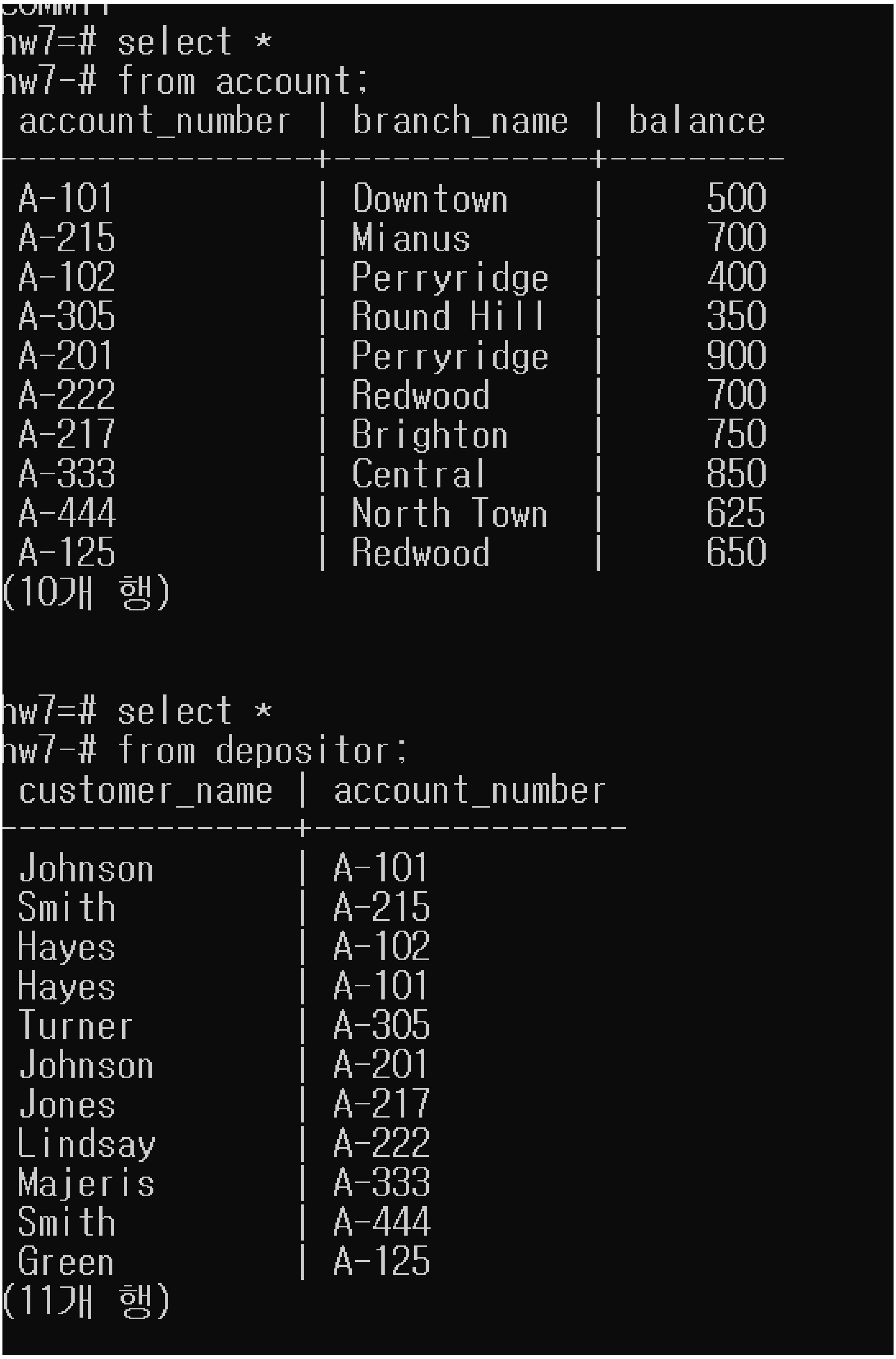
insert into account values (‘A-125’, ‘Redwood’, 650);

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

Commit;



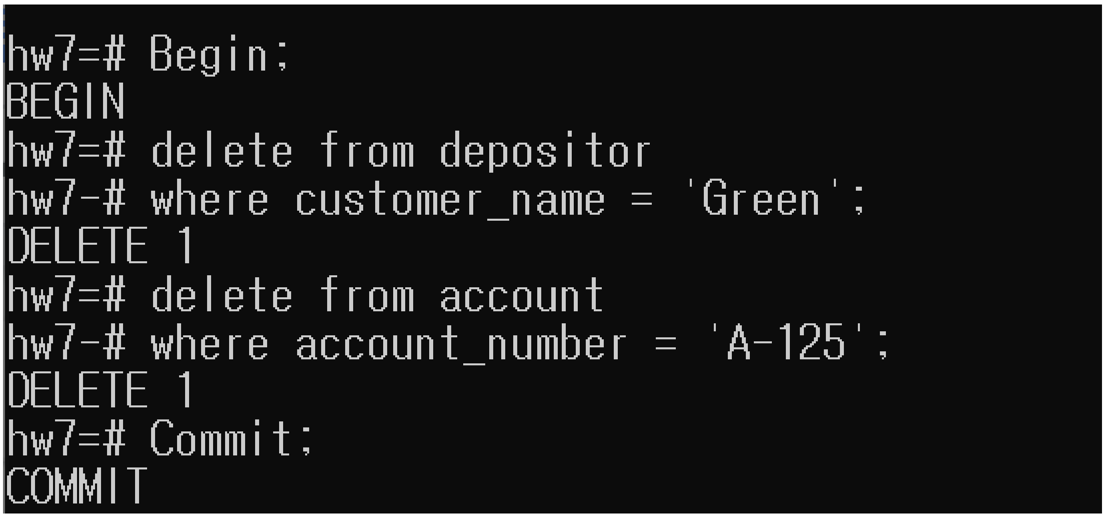
```
hw7=# Begin;
BEGIN
hw7=# insert into account values ('A-125', 'Redwood', 650);
INSERT 0 1
hw7=# insert into depositor values ('Green', 'A-125');
INSERT 0 1
hw7=# Commit;
COMMIT
```



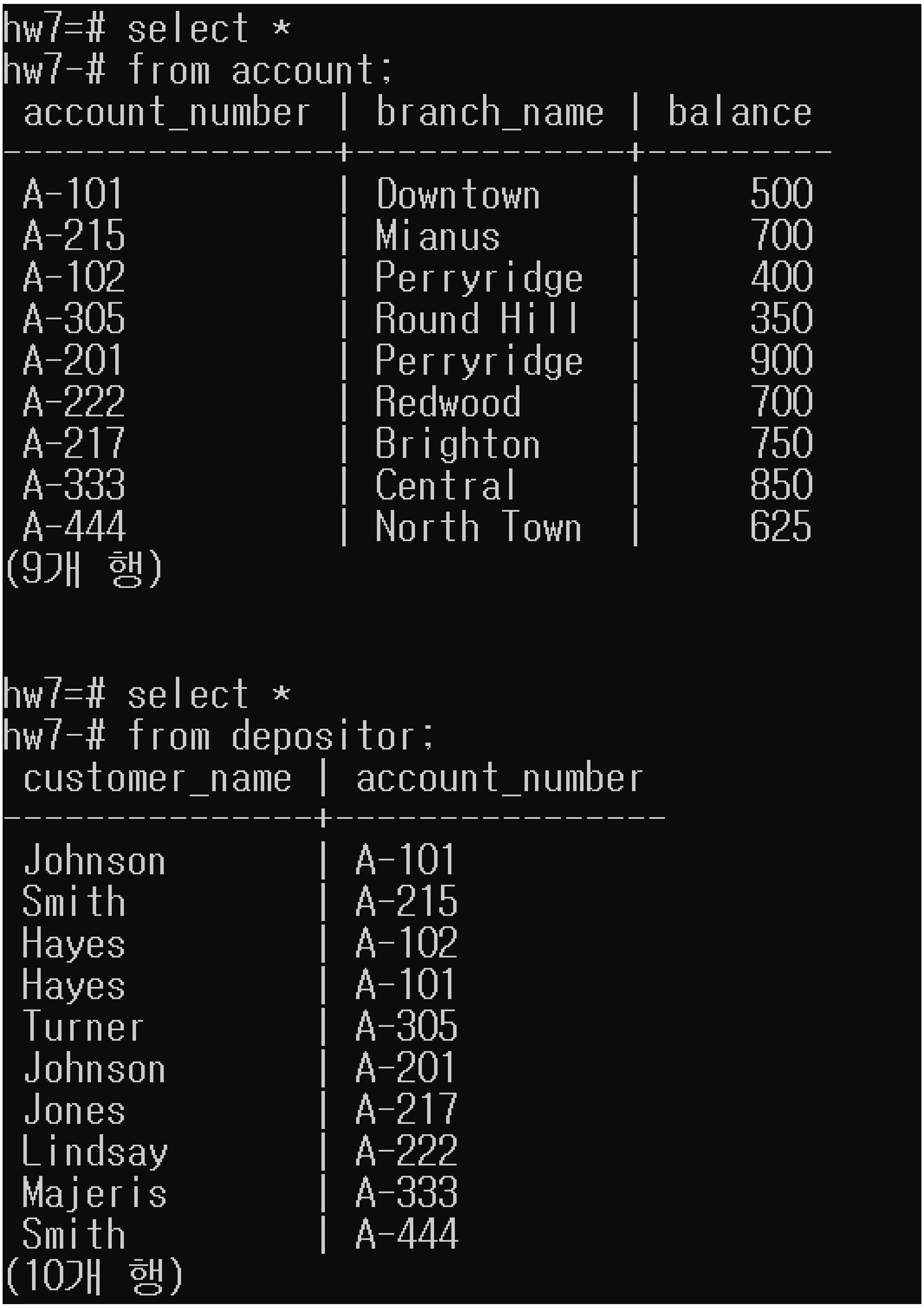
```
COMMIT
hw7=# select *
hw7-# from account;
 account_number | branch_name | balance
----------------+-------------+---------
 A-101          | Downtown    |     500
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-305          | Round Hill  |     350
 A-201          | Perryridge  |     900
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-444          | North Town  |     625
 A-125          | Redwood     |     650
(10개 행)


hw7=# select *
hw7-# from depositor;
 customer_name | account_number
---------------+----------------
 Johnson       | A-101
 Smith         | A-215
 Hayes         | A-102
 Hayes         | A-101
 Turner        | A-305
 Johnson       | A-201
 Jones         | A-217
 Lindsay       | A-222
 Majeris       | A-333
 Smith         | A-444
 Green         | A-125
(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;
 account_number | branch_name | balance
----------------+-------------+---------
 A-101          | Downtown    |     500
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-305          | Round Hill  |     350
 A-201          | Perryridge  |     900
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-444          | North Town  |     625
(9개 행)


hw7=# select *
hw7-# from depositor;
 customer_name | account_number
---------------+----------------
 Johnson       | A-101
 Smith         | A-215
 Hayes         | A-102
 Hayes         | A-101
 Turner        | A-305
 Johnson       | A-201
 Jones         | A-217
 Lindsay       | A-222
 Majeris       | A-333
 Smith         | A-444
(10개 행)
```

3.

Begin;

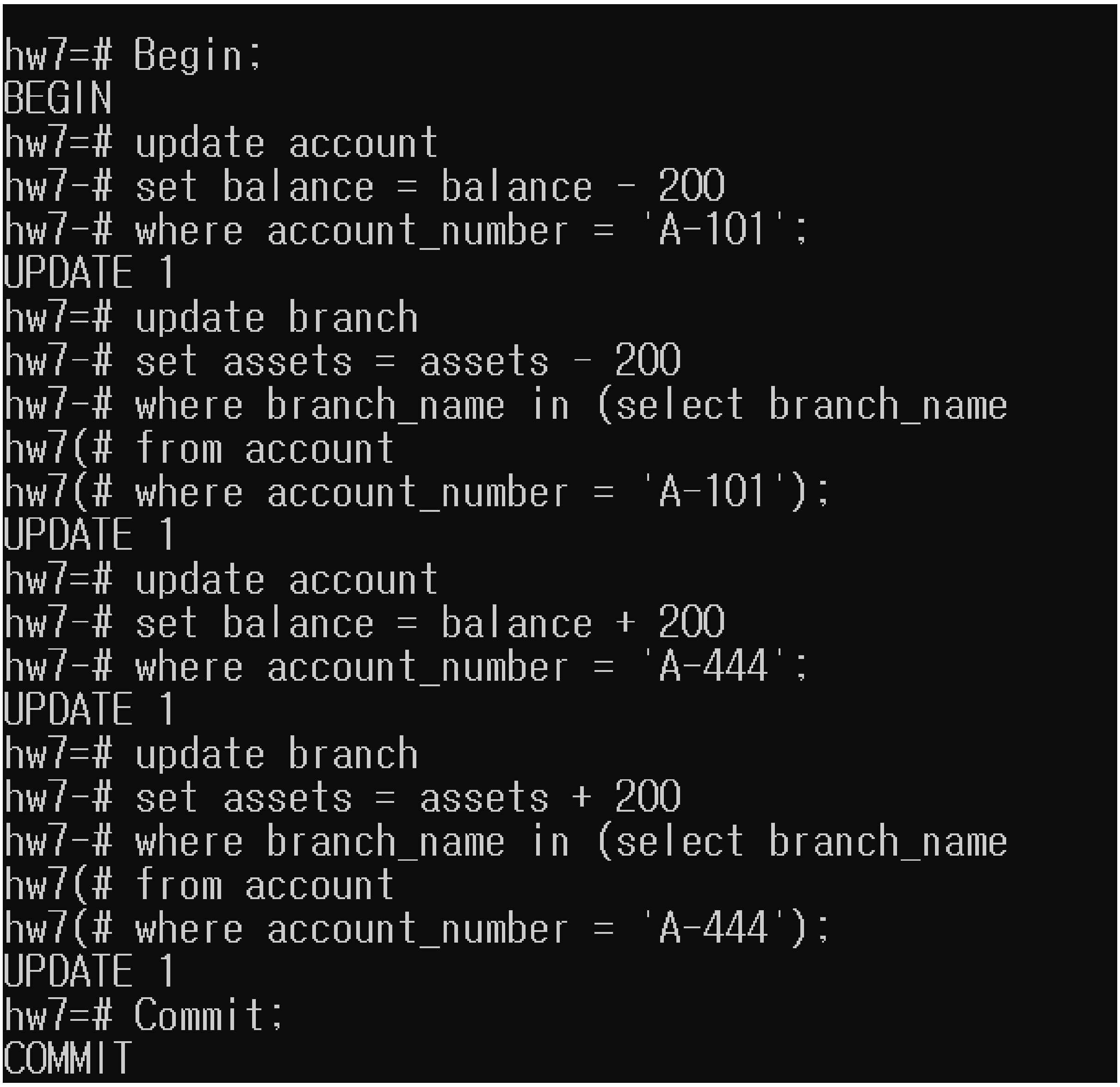
update account

set balance = balance - 200

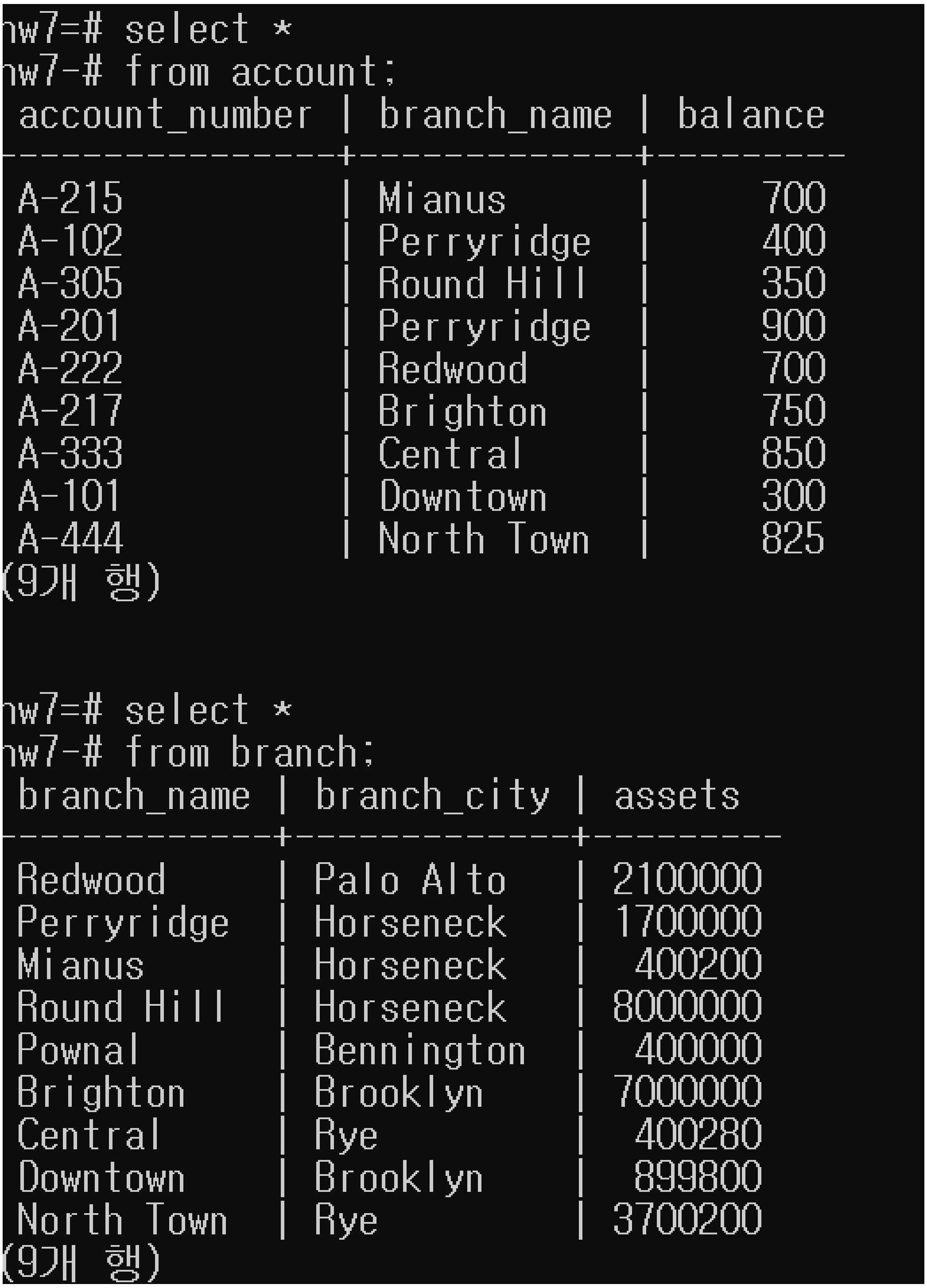
where account_number = 'A-101';

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
```



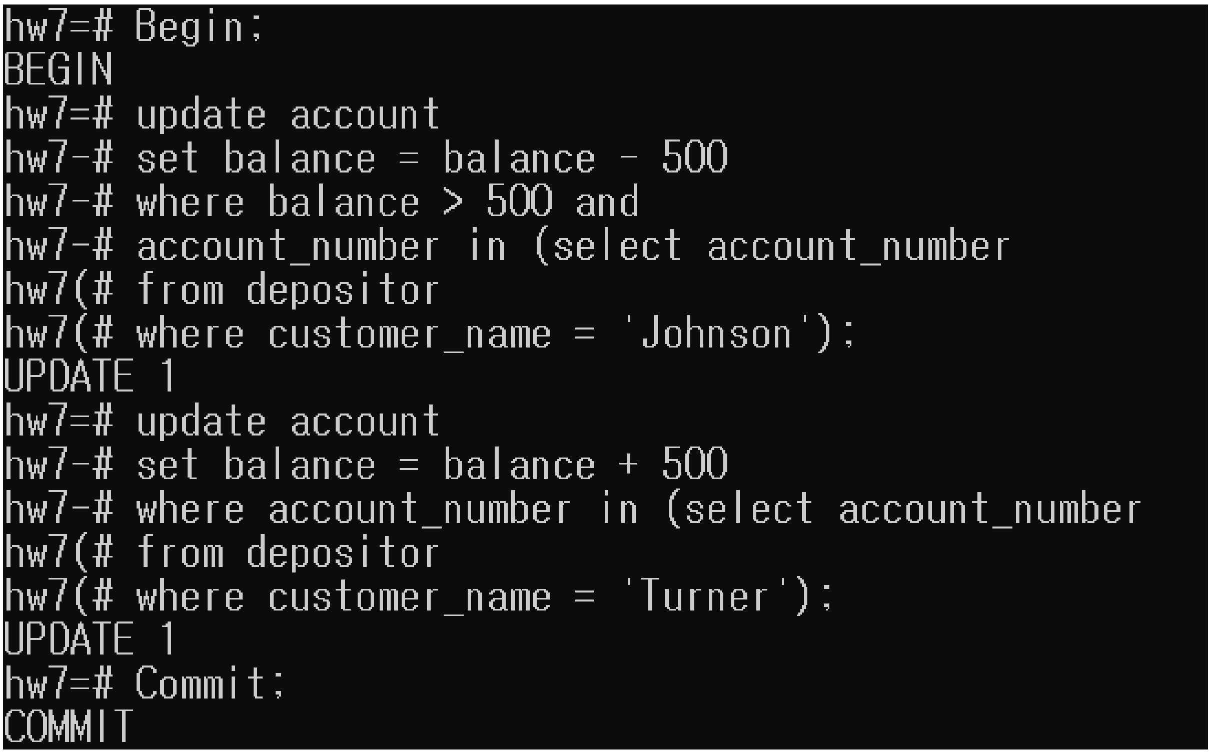
```
hw7=# select *
hw7-# from account;
 account_number | branch_name | balance
----------------+-------------+---------
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-305          | Round Hill  |     350
 A-201          | Perryridge  |     900
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-101          | Downtown    |     300
 A-444          | North Town  |     825
(9개 행)


hw7=# select *
hw7-# from branch;
 branch_name | branch_city | assets
-------------+-------------+---------
 Redwood     | Palo Alto   | 2100000
 Perryridge  | Horseneck   | 1700000
 Mianus      | Horseneck   |  400200
 Round Hill  | Horseneck   | 8000000
 Pownal      | Bennington  |  400000
 Brighton    | Brooklyn    | 7000000
 Central     | Rye         |  400280
 Downtown    | Brooklyn    |  899800
 North Town  | Rye         | 3700200
(9개 행)
```

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

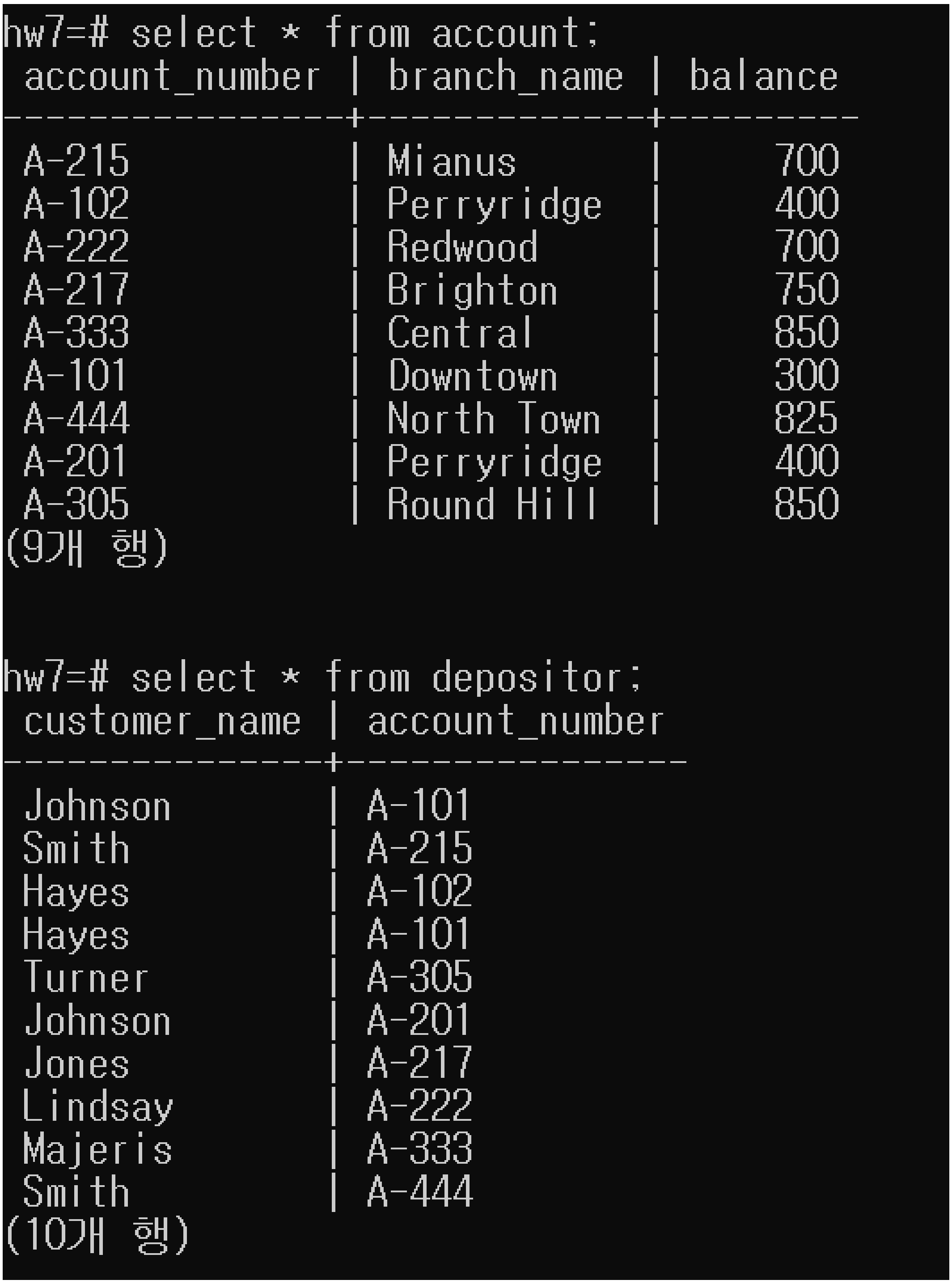
1.

**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 | branch_name | balance
----------------+-------------+---------
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-101          | Downtown    |     300
 A-444          | North Town  |     825
 A-201          | Perryridge  |     400
 A-305          | Round Hill  |     850
(9개 행)


hw7=# select * from depositor;
 customer_name | account_number
---------------+----------------
 Johnson       | A-101
 Smith         | A-215
 Hayes         | A-102
 Hayes         | A-101
 Turner        | A-305
 Johnson       | A-201
 Jones         | A-217
 Lindsay       | A-222
 Majeris       | A-333
 Smith         | A-444
(10개 행)
```

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

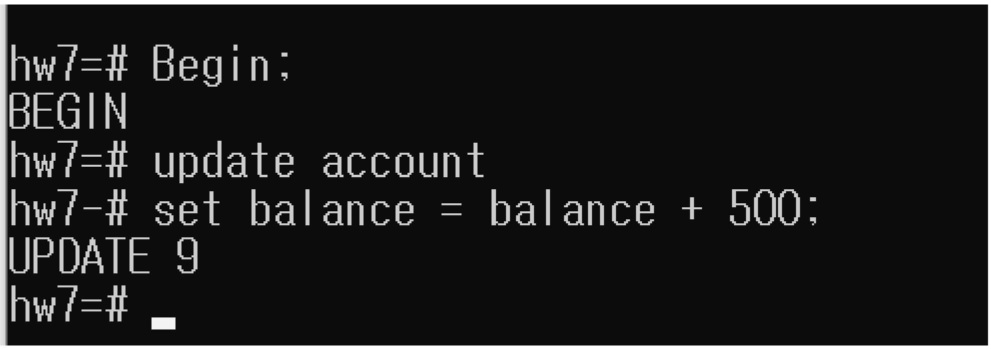
2.

Begin;

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

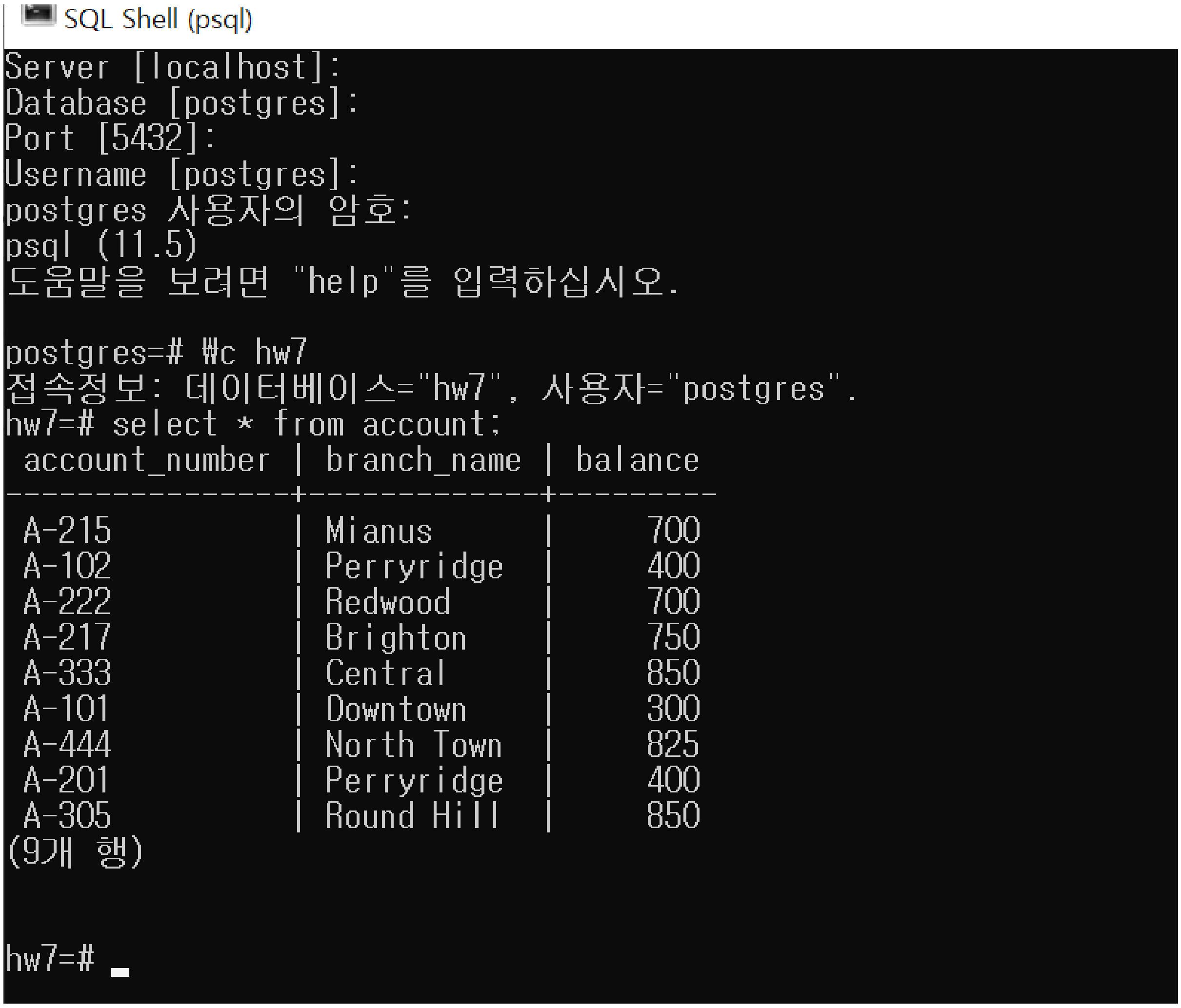
You can make a crash by just closing the window

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.



```
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
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-101          | Downtown    |     300
 A-444          | North Town  |     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.

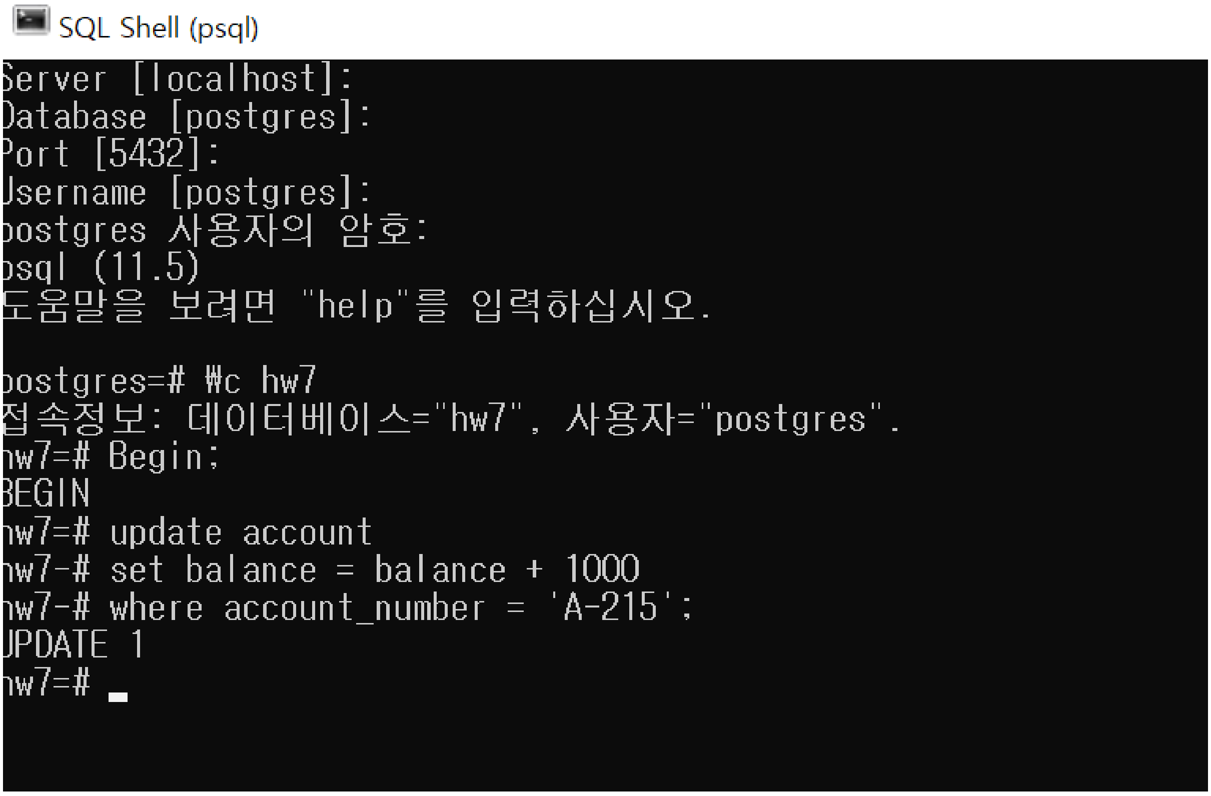
In the window A, begin a transaction with the isolation level Read Committed.

1. Dirty read

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

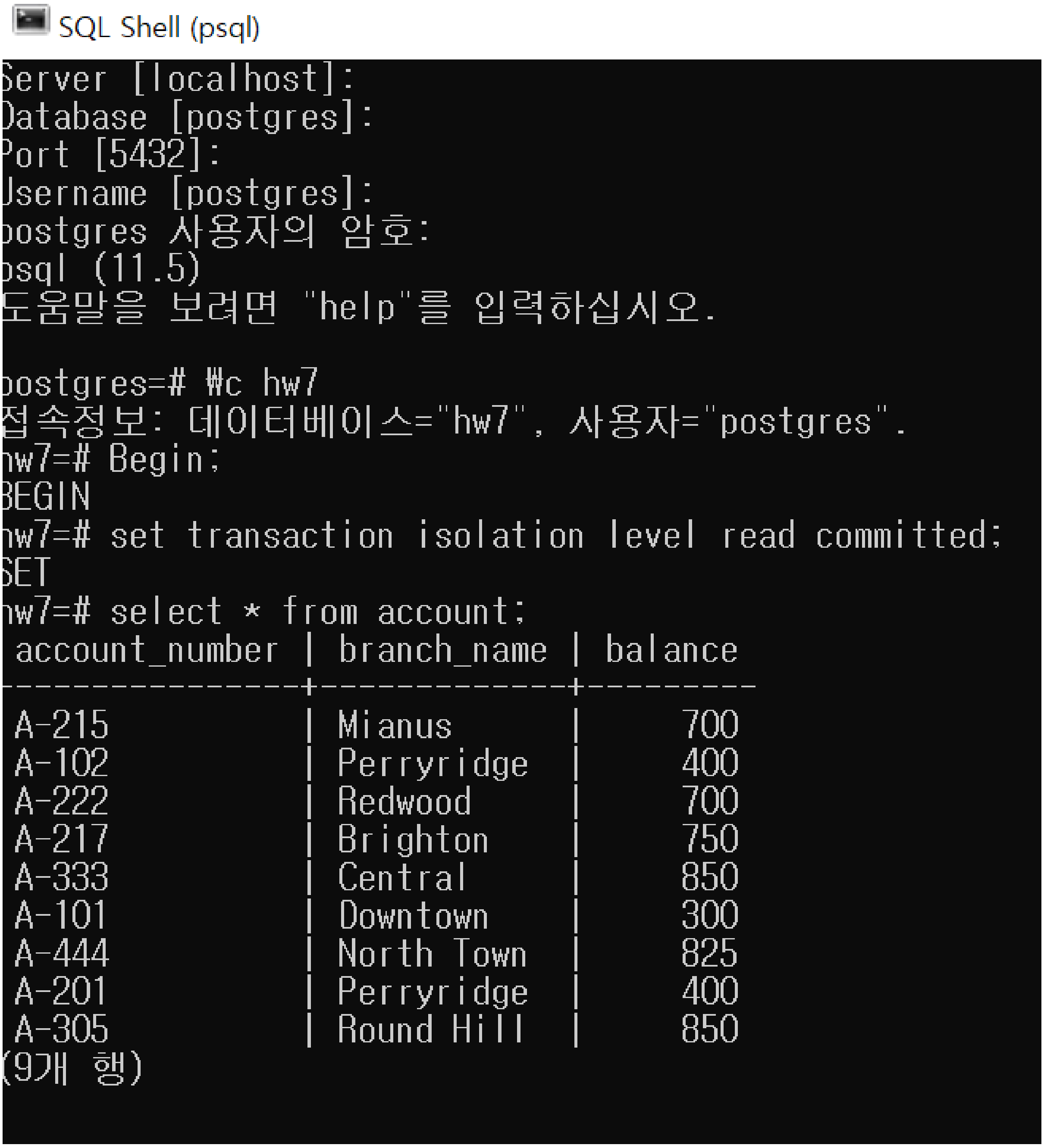
Is the update in the window B applied to the transaction of the window A?

If yes, why? If no, why?



SQL Shell (psql)

```
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
postgres 사용자의 암호:
psql (11.5)
도움말을 보려면 "help"를 입력하십시오.

postgres=# ₩c hw7
접속정보: 데이터베이스="hw7", 사용자="postgres".
hw7=# Begin;
BEGIN
hw7=# update account
hw7-# set balance = balance + 1000
hw7-# where account_number = 'A-215';
UPDATE 1
hw7=# _
```



```
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=# Begin;
BEGIN
hw7=# set transaction isolation level read committed;
SET
hw7=# select * from account;
 account_number | branch_name | balance
----------------+-------------+---------
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-101          | Downtown    |     300
 A-444          | North Town  |     825
 A-201          | Perryridge  |     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

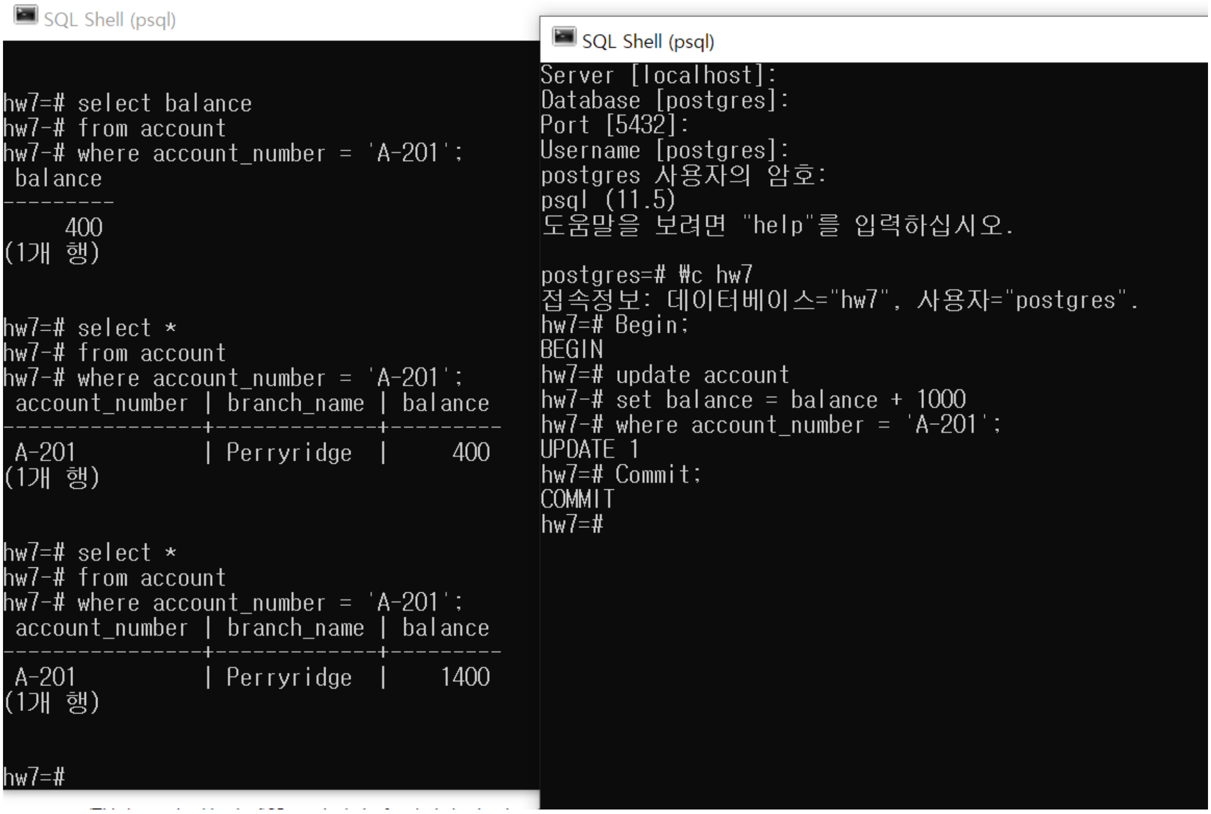
In the window A, Check the A 201's balance using an appropriate SQL query and,

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?



```
hw7=# select balance
hw7-# from account
hw7-# where account_number = 'A-201';
 balance
---------
     400
(1개 행)


hw7=# select *
hw7-# from account
hw7-# where account_number = 'A-201';
 account_number | branch_name | balance
----------------+-------------+---------
 A-201          | Perryridge  |     400
(1개 행)


hw7=# select *
hw7-# from account
hw7-# where account_number = 'A-201';
 account_number | branch_name | balance
----------------+-------------+---------
 A-201          | Perryridge  |    1400
(1개 행)


hw7=#
```

```
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=# Begin;
BEGIN
hw7=# update account
hw7-# set balance = balance + 1000
hw7-# where account_number = 'A-201';
UPDATE 1
hw7=# Commit;
COMMIT
hw7=#
```

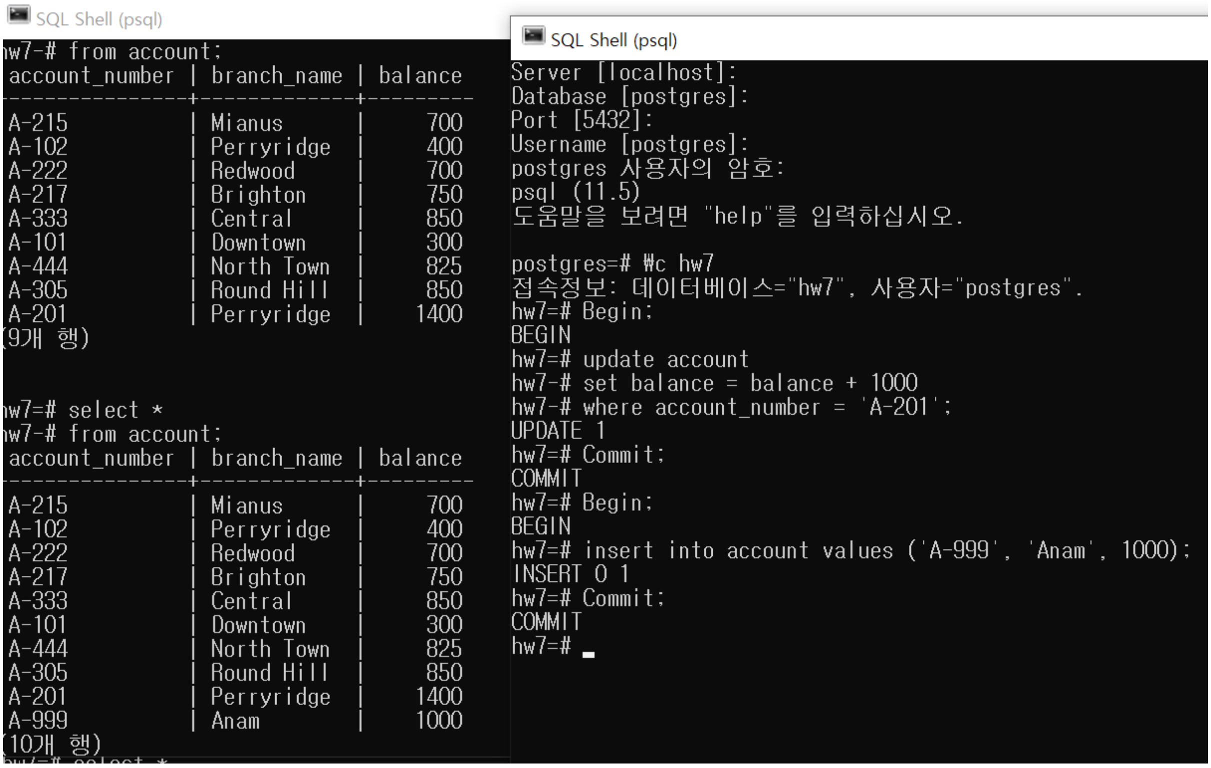
업데이트가 적용되었다. isolation level이 read committed한 상태에서 non repeatable read가 발생 가능하기 때문이다. 무엇보다 commit으로 종료되었기 때문에, 해당 업데이트된 값을 받는다.

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?



```
w7-# from account;
 account_number | branch_name | balance
----------------+-------------+---------
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-101          | Downtown    |     300
 A-444          | North Town  |     825
 A-305          | Round Hill  |     850
 A-201          | Perryridge  |    1400
(9개 행)


w7=# select *
w7-# from account;
 account_number | branch_name | balance
----------------+-------------+---------
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-101          | Downtown    |     300
 A-444          | North Town  |     825
 A-305          | Round Hill  |     850
 A-201          | Perryridge  |    1400
 A-999          | Anam        |    1000
(10개 행)
hw7=# select *
```

```
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=# 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=# insert into account values ('A-999', 'Anam', 1000);
INSERT 0 1
hw7=# Commit;
COMMIT
hw7=# _
```

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

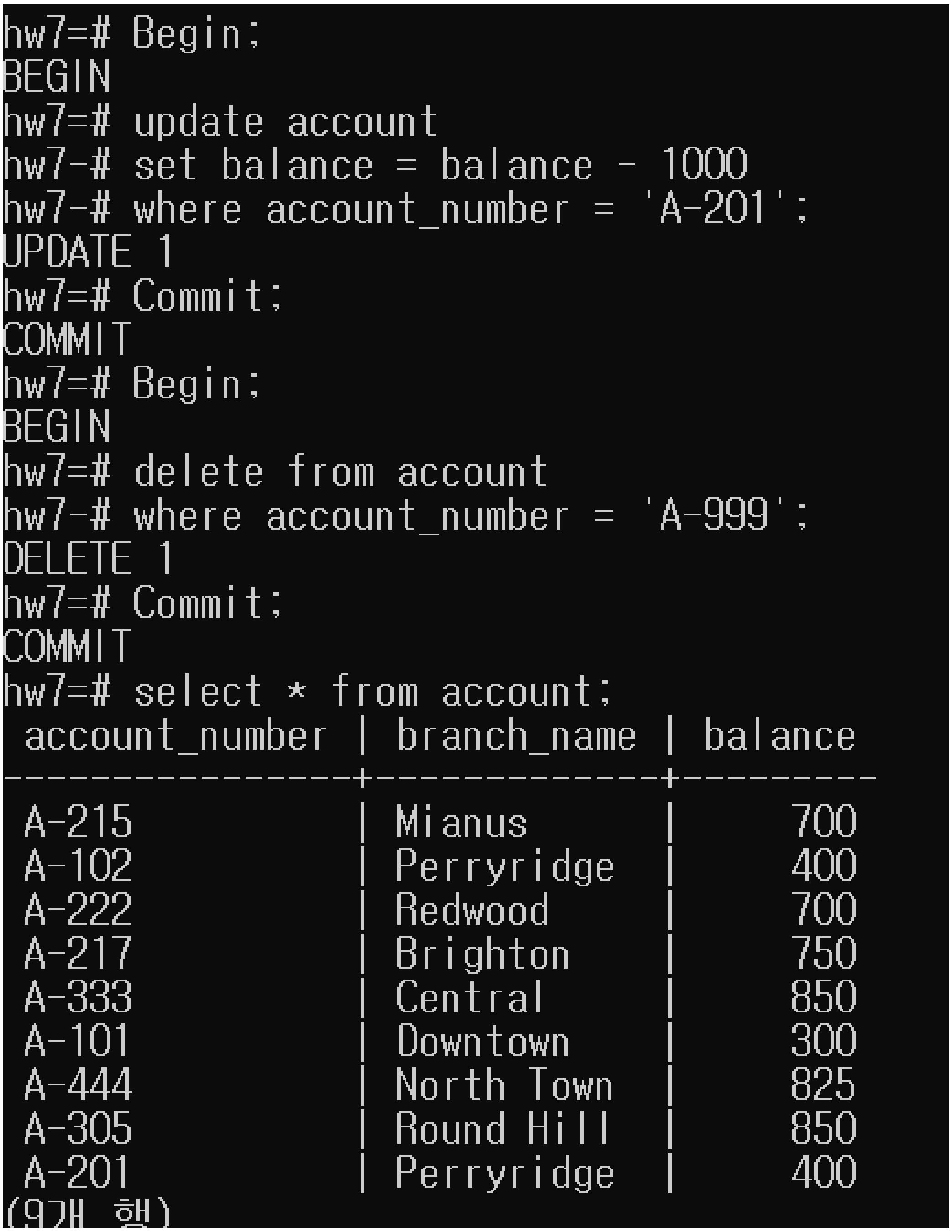
<Exercise 4>

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';
DELETE 1
hw7=# Commit;
COMMIT
hw7=# select * from account;
 account_number | branch_name | balance
----------------+-------------+---------
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-101          | Downtown    |     300
 A-444          | North Town  |     825
 A-305          | Round Hill  |     850
 A-201          | Perryridge  |     400
(9개 행)
```

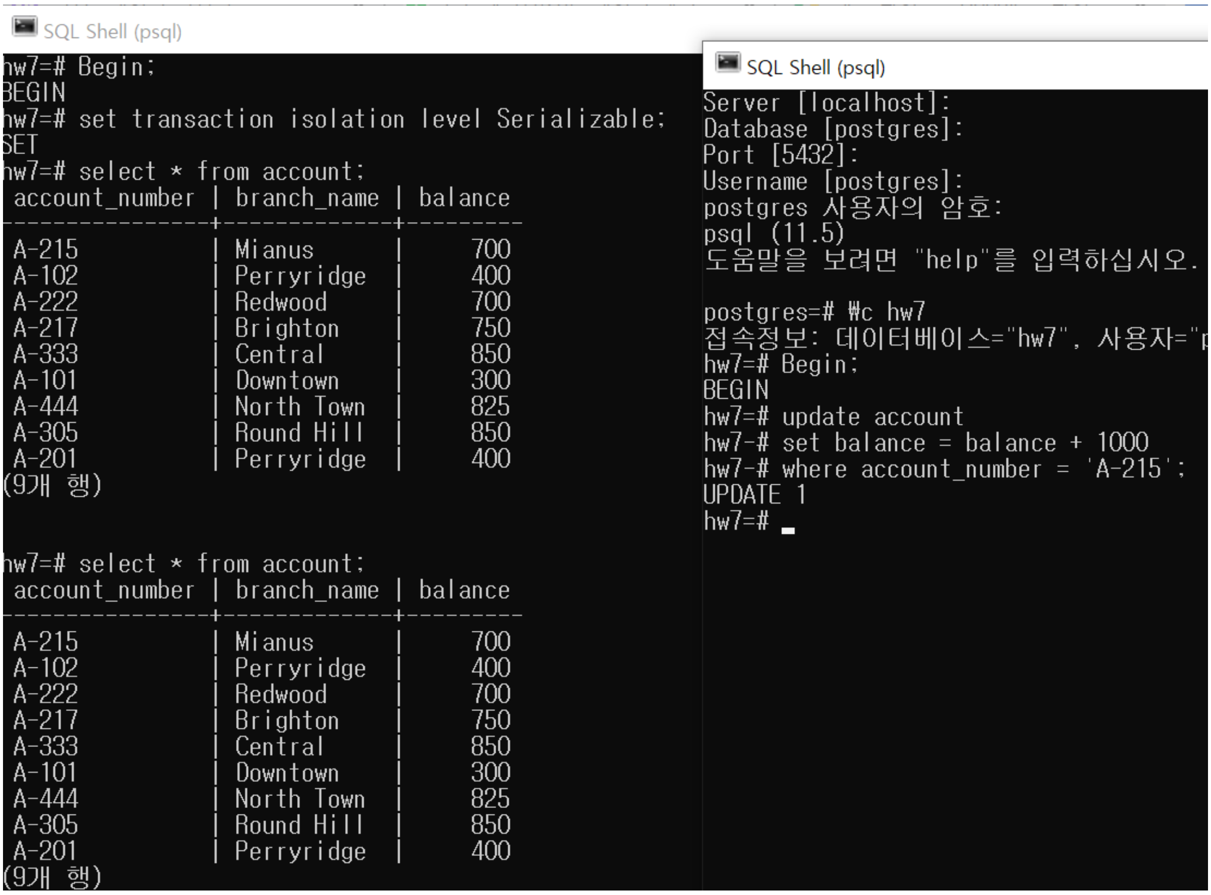
동일한 초기화 상태로 만들어줍니다.

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?



```
hw7=# Begin;
BEGIN
hw7=# set transaction isolation level Serializable;
SET
hw7=# select * from account;
 account_number | branch_name | balance
----------------+-------------+---------
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-101          | Downtown    |     300
 A-444          | North Town  |     825
 A-305          | Round Hill  |     850
 A-201          | Perryridge  |     400
(9개 행)


hw7=# select * from account;
 account_number | branch_name | balance
----------------+-------------+---------
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-101          | Downtown    |     300
 A-444          | North Town  |     825
 A-305          | Round Hill  |     850
 A-201          | Perryridge  |     400
(9개 행)
```

```
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="p
hw7=# Begin;
BEGIN
hw7=# update account
hw7-# set balance = balance + 1000
hw7-# where account_number = 'A-215';
UPDATE 1
hw7=#
```

update가 적용되지 않는다. dirty read를 막아주기 때문이다.

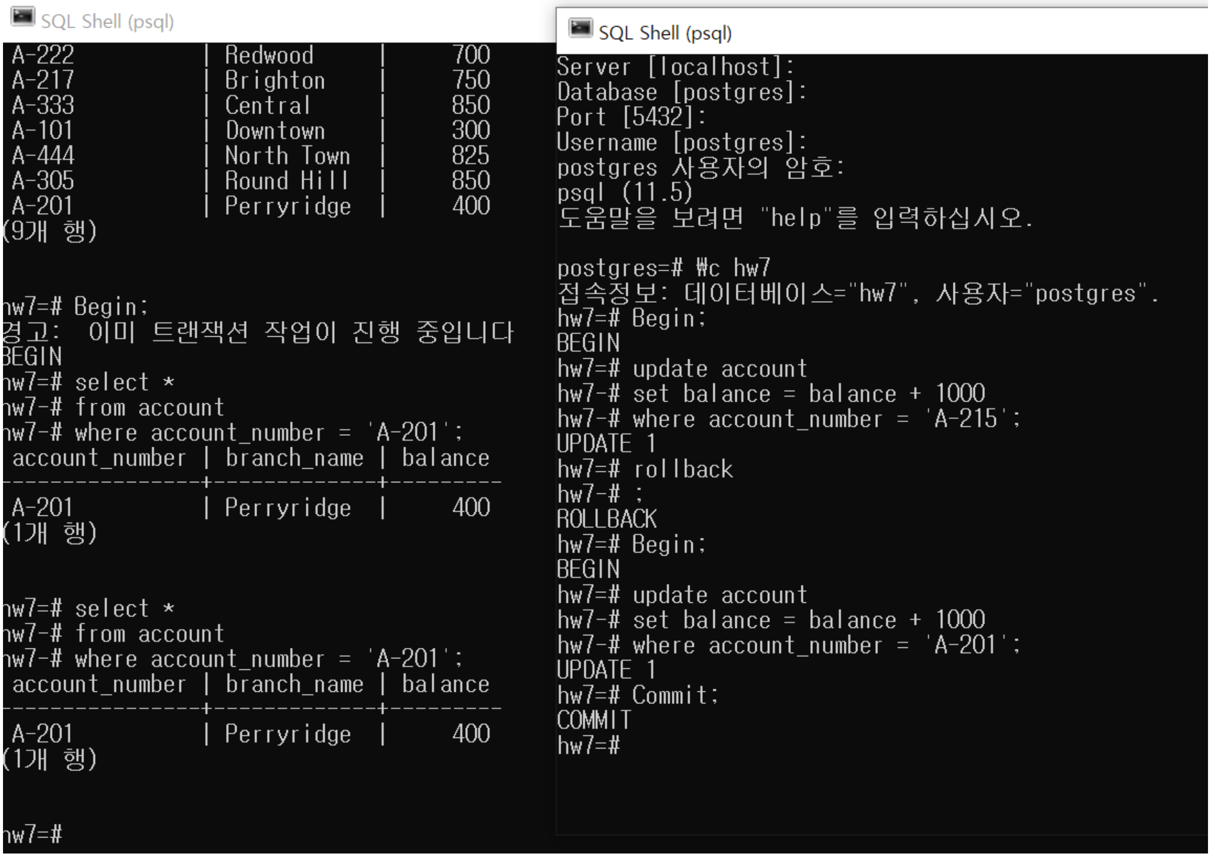
2. Non repeatable read

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?



```
A-222           | Redwood     |     700
A-217           | Brighton    |     750
A-333           | Central     |     850
A-101           | Downtown    |     300
A-444           | North Town  |     825
A-305           | Round Hill  |     850
A-201           | Perryridge  |     400
(9개 행)


hw7=# Begin;
경고:  이미 트랜잭션 작업이 진행 중입니다
BEGIN
hw7=# select *
hw7-# from account
hw7-# where account_number = 'A-201';
 account_number | branch_name | balance
----------------+-------------+---------
 A-201          | Perryridge  |     400
(1개 행)


hw7=# select *
hw7-# from account
hw7-# where account_number = 'A-201';
 account_number | branch_name | balance
----------------+-------------+---------
 A-201          | Perryridge  |     400
(1개 행)


hw7=#
```

```
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=# Begin;
BEGIN
hw7=# update account
hw7-# set balance = balance + 1000
hw7-# where account_number = 'A-215';
UPDATE 1
hw7=# rollback
hw7-# ;
ROLLBACK
hw7=# Begin;
BEGIN
hw7=# update account
hw7-# set balance = balance + 1000
hw7-# where account_number = 'A-201';
UPDATE 1
hw7=# Commit;
COMMIT
hw7=#
```

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

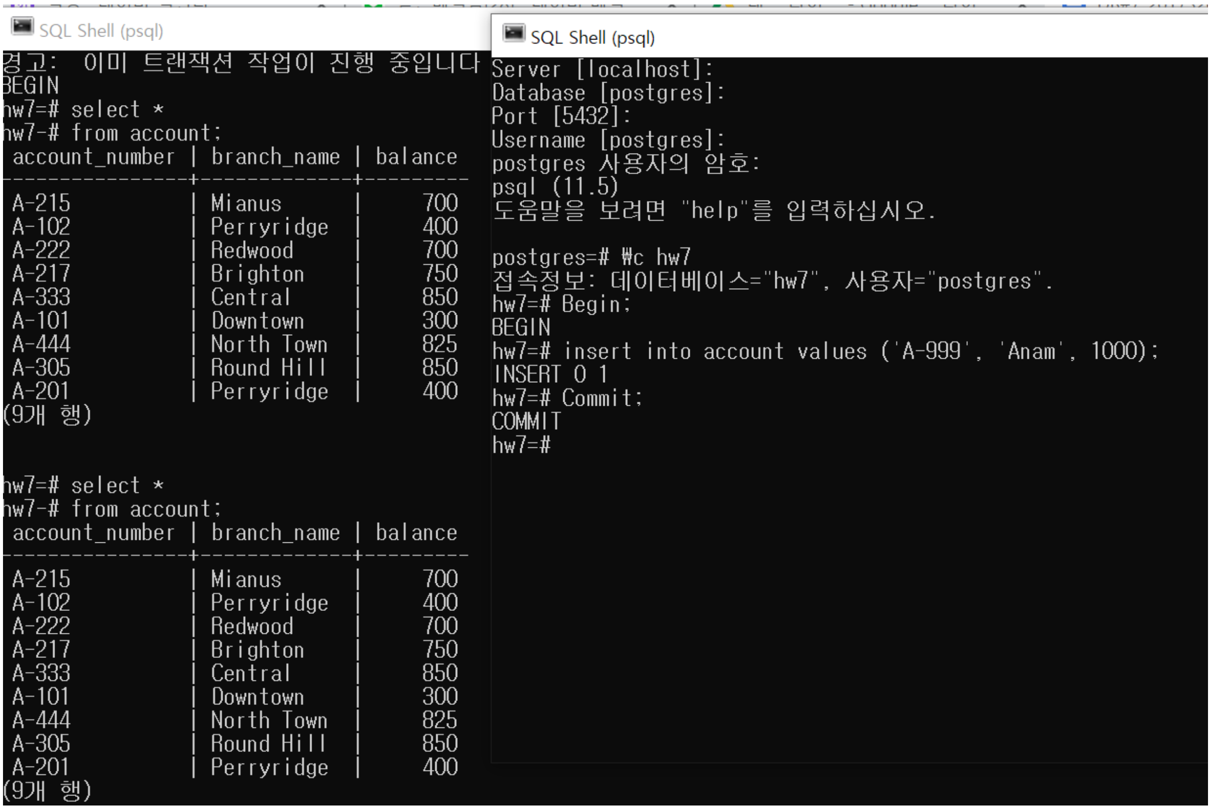
3. Phantom read

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?



```
경고:  이미 트랜잭션 작업이 진행 중입니다
BEGIN
hw7=# select *
hw7-# from account;
 account_number | branch_name | balance
----------------+-------------+---------
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-101          | Downtown    |     300
 A-444          | North Town  |     825
 A-305          | Round Hill  |     850
 A-201          | Perryridge  |     400
(9개 행)


hw7=# select *
hw7-# from account;
 account_number | branch_name | balance
----------------+-------------+---------
 A-215          | Mianus      |     700
 A-102          | Perryridge  |     400
 A-222          | Redwood     |     700
 A-217          | Brighton    |     750
 A-333          | Central     |     850
 A-101          | Downtown    |     300
 A-444          | North Town  |     825
 A-305          | Round Hill  |     850
 A-201          | Perryridge  |     400
(9개 행)
```

```
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=# Begin;
BEGIN
hw7=# insert into account values ('A-999', 'Anam', 1000);
INSERT 0 1
hw7=# Commit;
COMMIT
hw7=#
```

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