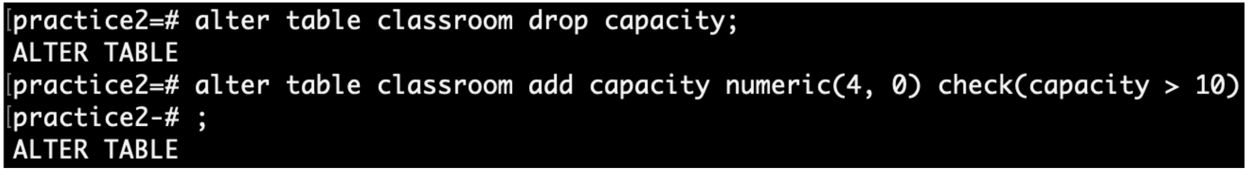
HW4

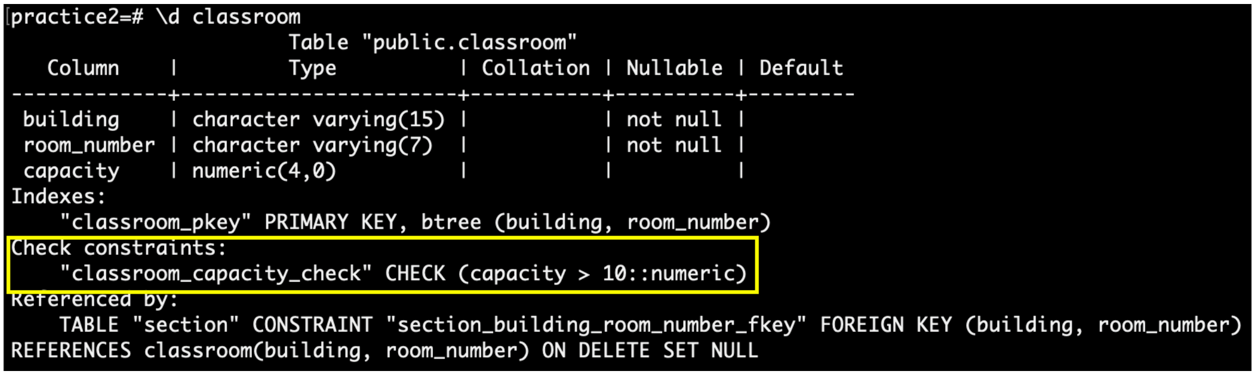
Name: 서천함 ID: 2017120486

Exercise 1

- Set up a new integrity constraint:

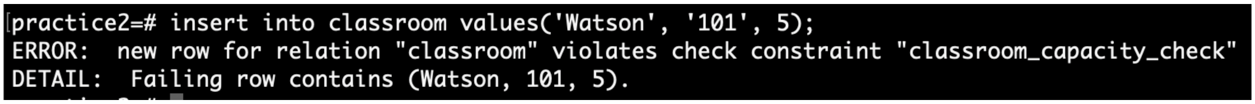


```
[practice2=# alter table classroom drop capacity;
ALTER TABLE
[practice2=# alter table classroom add capacity numeric(4, 0) check(capacity > 10)
[practice2-# ;
ALTER TABLE
```



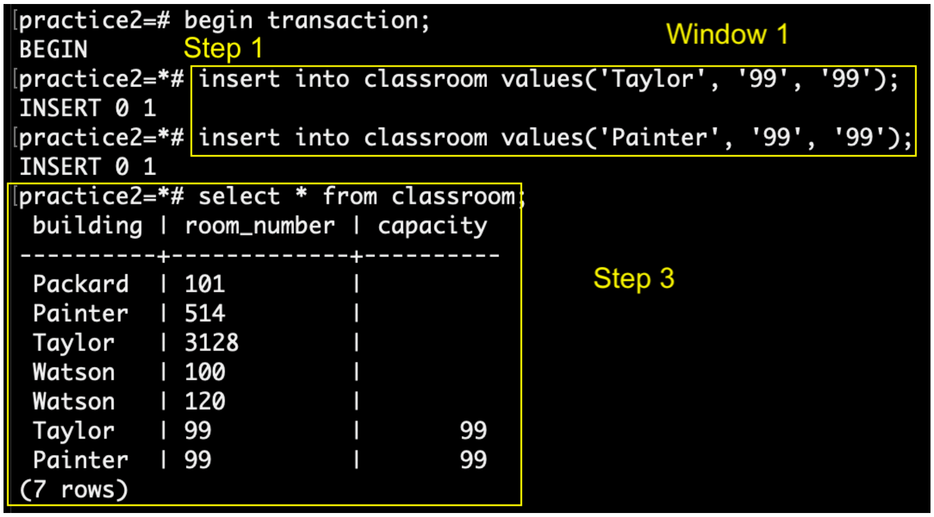
```
[practice2=# \d classroom
                      Table "public.classroom"
   Column    |          Type          | Collation | Nullable | Default
-------------+------------------------+-----------+----------+---------
 building    | character varying(15) |           | not null |
 room_number | character varying(7)  |           | not null |
 capacity    | numeric(4,0)          |           |          |
Indexes:
    "classroom_pkey" PRIMARY KEY, btree (building, room_number)
Check constraints:
    "classroom_capacity_check" CHECK (capacity > 10::numeric)
Referenced by:
    TABLE "section" CONSTRAINT "section_building_room_number_fkey" FOREIGN KEY (building, room_number)
REFERENCES classroom(building, room_number) ON DELETE SET NULL
```

- Insertion: violation

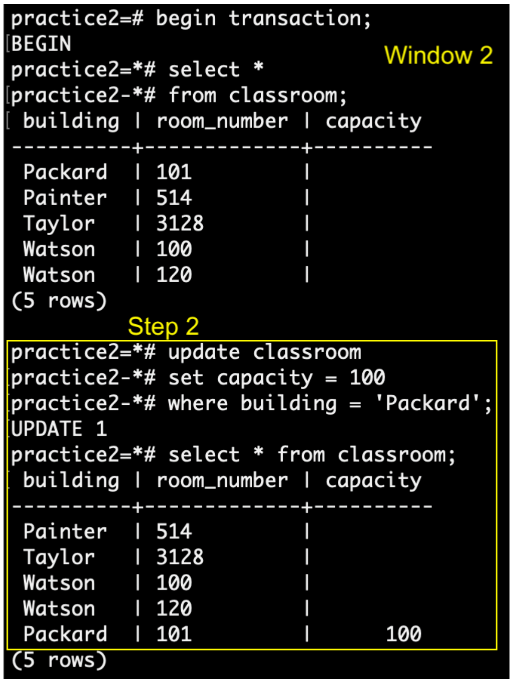


```
[practice2=# insert into classroom values('Watson', '101', 5);
ERROR:  new row for relation "classroom" violates check constraint "classroom_capacity_check"
DETAIL:  Failing row contains (Watson, 101, 5).
```

Exercise 2



```
[practice2=# begin transaction;
 BEGIN
[practice2=*# insert into classroom values('Taylor', '99', '99');
 INSERT 0 1
[practice2=*# insert into classroom values('Painter', '99', '99');
 INSERT 0 1
[practice2=*# select * from classroom;
  building | room_number | capacity
 ----------+-------------+----------
  Packard  | 101         |
  Painter  | 514         |
  Taylor   | 3128        |
  Watson   | 100         |
  Watson   | 120         |
  Taylor   | 99          |       99
  Painter  | 99          |       99
 (7 rows)
```



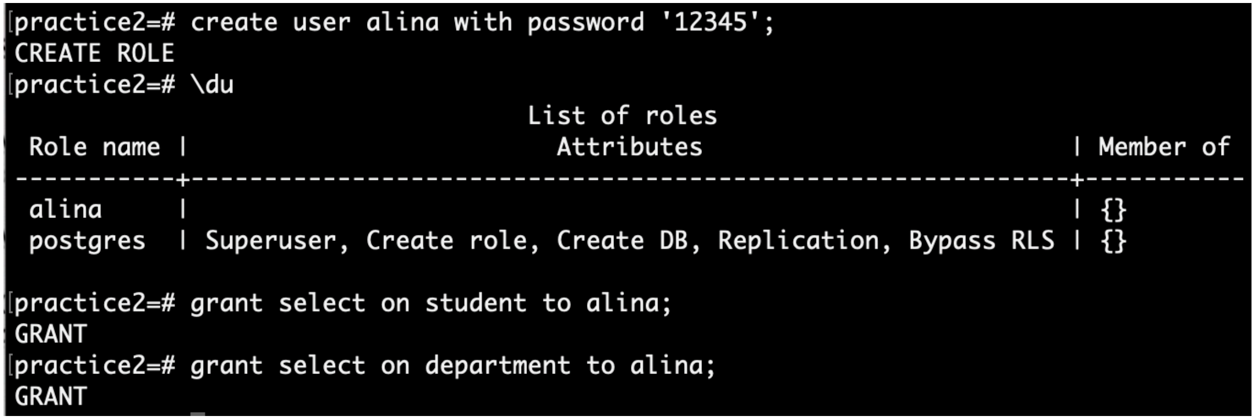
```
practice2=# begin transaction;
[BEGIN
practice2=*# select *
[practice2-*# from classroom;
[ building | room_number | capacity
----------+-------------+----------
 Packard  | 101         |
 Painter  | 514         |
 Taylor   | 3128        |
 Watson   | 100         |
 Watson   | 120         |
(5 rows)

practice2=*# update classroom
practice2-*# set capacity = 100
practice2-*# where building = 'Packard';
UPDATE 1
practice2=*# select * from classroom;
 building | room_number | capacity
----------+-------------+----------
 Painter  | 514         |
 Taylor   | 3128        |
 Watson   | 100         |
 Watson   | 120         |
 Packard  | 101         |      100
(5 rows)
```

- Step 1: in Window 1, I inserted 2 new tuples in table *classroom*;
- Step 2: in Window 2, I updated the capacity of every building named ‘Packard’ and check all the tuples in the table *classroom*. However, the insertion in Window 1 didn’t show up.
- Step 3: in Window 1, I checked the table *classroom* but the updates in Window 2 didn’t show up.
  - ⇨ The commands in different transactions are executed in an isolated manner.

Exercise 3

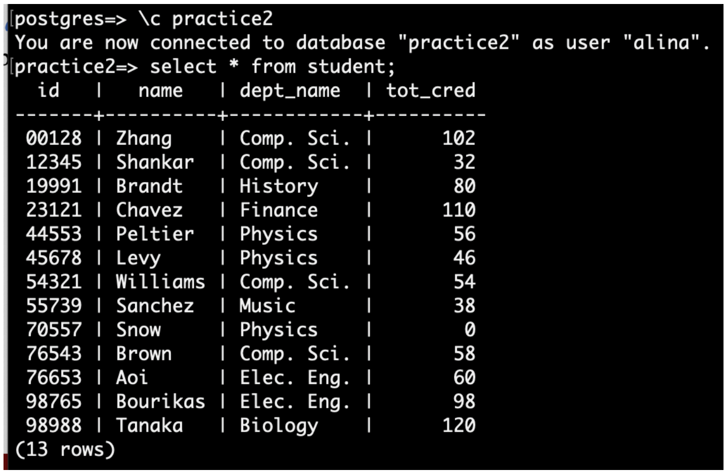
- Step 1: Create a new user named ‘alina’ with password
- Step 2: grant the selection privileges on table *student, department* to ‘alina’;



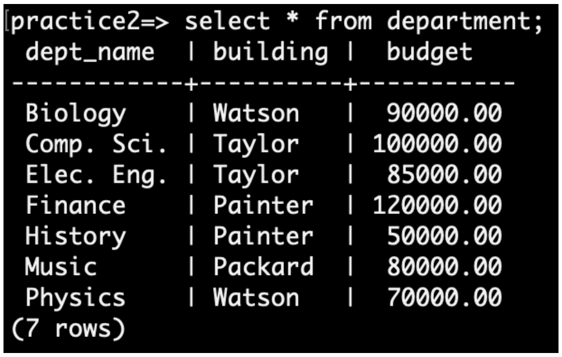
```
[practice2=# create user alina with password '12345';
CREATE ROLE
[practice2=# \du
                                   List of roles
 Role name |                         Attributes                         | Member of
-----------+------------------------------------------------------------+-----------
 alina     |                                                            | {}
 postgres  | Superuser, Create role, Create DB, Replication, Bypass RLS | {}

[practice2=# grant select on student to alina;
GRANT
[practice2=# grant select on department to alina;
GRANT
```

- Step 3: Login in as user “alina”; test the granted access to table *student, department*;

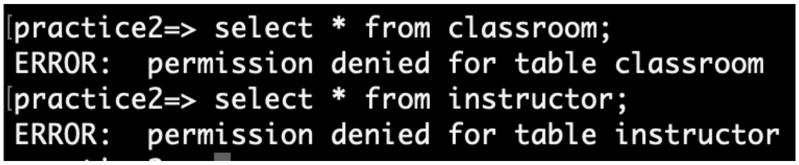


```
[postgres=> \c practice2
You are now connected to database "practice2" as user "alina".
[practice2=> select * from student;
  id   |   name   | dept_name  | tot_cred
-------+----------+------------+----------
 00128 | Zhang    | Comp. Sci. |      102
 12345 | Shankar  | Comp. Sci. |       32
 19991 | Brandt   | History    |       80
 23121 | Chavez   | Finance    |      110
 44553 | Peltier  | Physics    |       56
 45678 | Levy     | Physics    |       46
 54321 | Williams | Comp. Sci. |       54
 55739 | Sanchez  | Music      |       38
 70557 | Snow     | Physics    |        0
 76543 | Brown    | Comp. Sci. |       58
 76653 | Aoi      | Elec. Eng. |       60
 98765 | Bourikas | Elec. Eng. |       98
 98988 | Tanaka   | Biology    |      120
(13 rows)
```



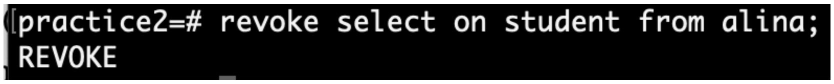
```
[practice2=> select * from department;
 dept_name  | building |  budget
------------+----------+-----------
 Biology    | Watson   |  90000.00
 Comp. Sci. | Taylor   | 100000.00
 Elec. Eng. | Taylor   |  85000.00
 Finance    | Painter  | 120000.00
 History    | Painter  |  50000.00
 Music      | Packard  |  80000.00
 Physics    | Watson   |  70000.00
(7 rows)
```

- Step 4: Test some non-authorized accesses: all denied;



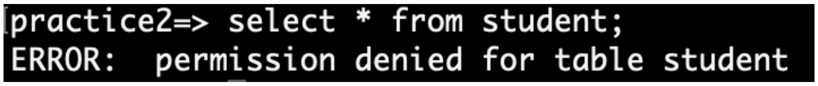
```
[practice2=> select * from classroom;
ERROR:  permission denied for table classroom
[practice2=> select * from instructor;
ERROR:  permission denied for table instructor
```

- Step 5: Revoke the access privilege of selection from user ‘alina’;



```
[practice2=# revoke select on student from alina;
REVOKE
```

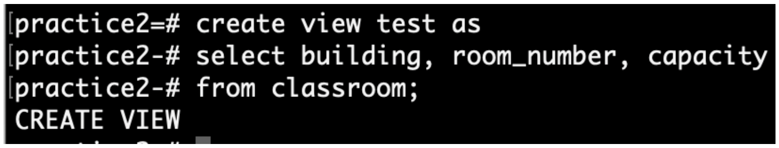
- Step 6: Login in as user ‘alina’ and test the access to table *student* again: denied;



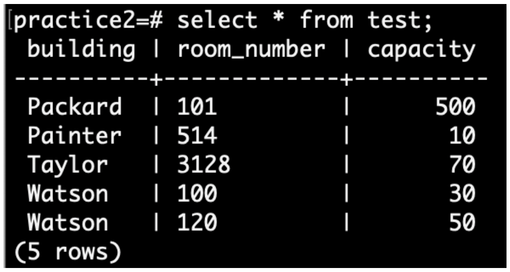
```
[practice2=> select * from student;
ERROR:  permission denied for table student
```

Exercise 4

- Step 1: Create a view named ‘test’

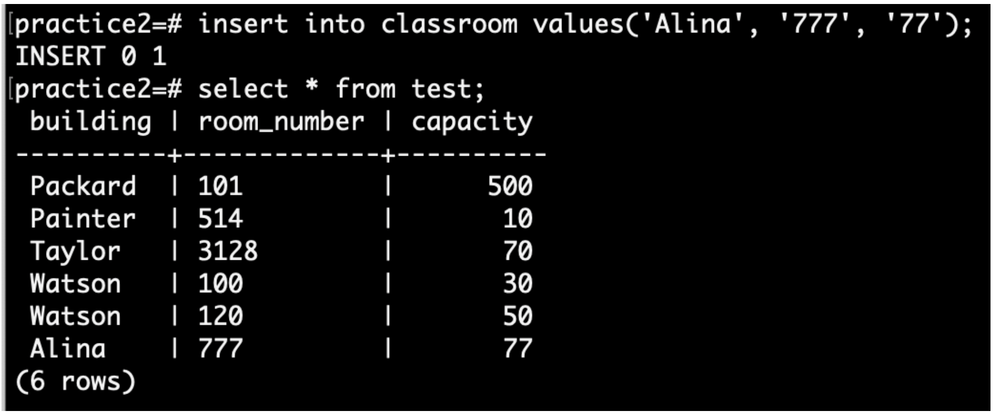


```
[practice2=# create view test as
[practice2-# select building, room_number, capacity
[practice2-# from classroom;
CREATE VIEW
```



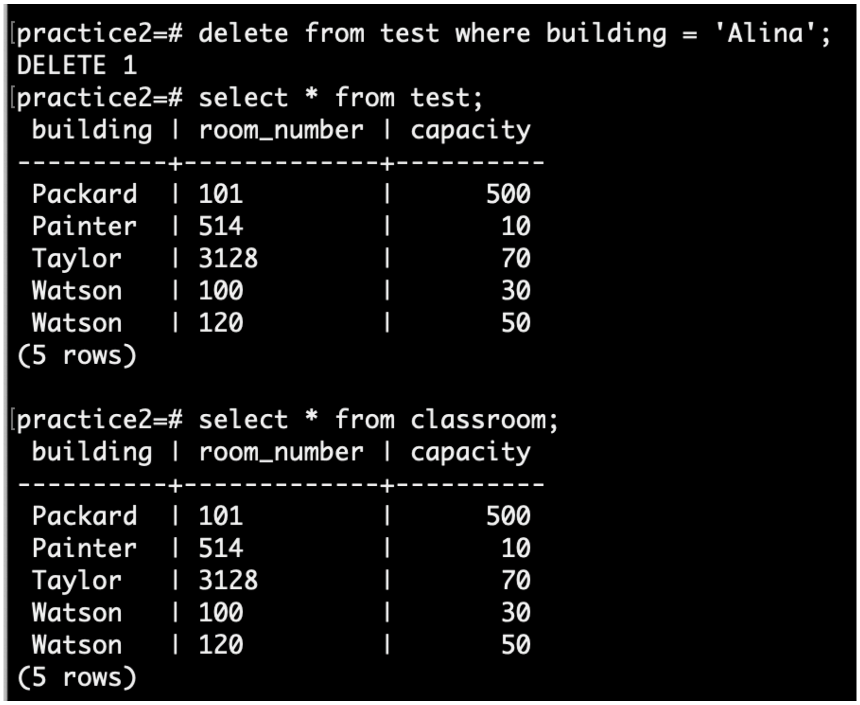
```
[practice2=# select * from test;
 building | room_number | capacity
----------+-------------+----------
 Packard  | 101         |      500
 Painter  | 514         |       10
 Taylor   | 3128        |       70
 Watson   | 100         |       30
 Watson   | 120         |       50
(5 rows)
```

- Step 2: Update the source relation by inserting a new tuple into table *classroom*; We can find that as the actual relations used in the view definition updated, the view is kept up-to-date.



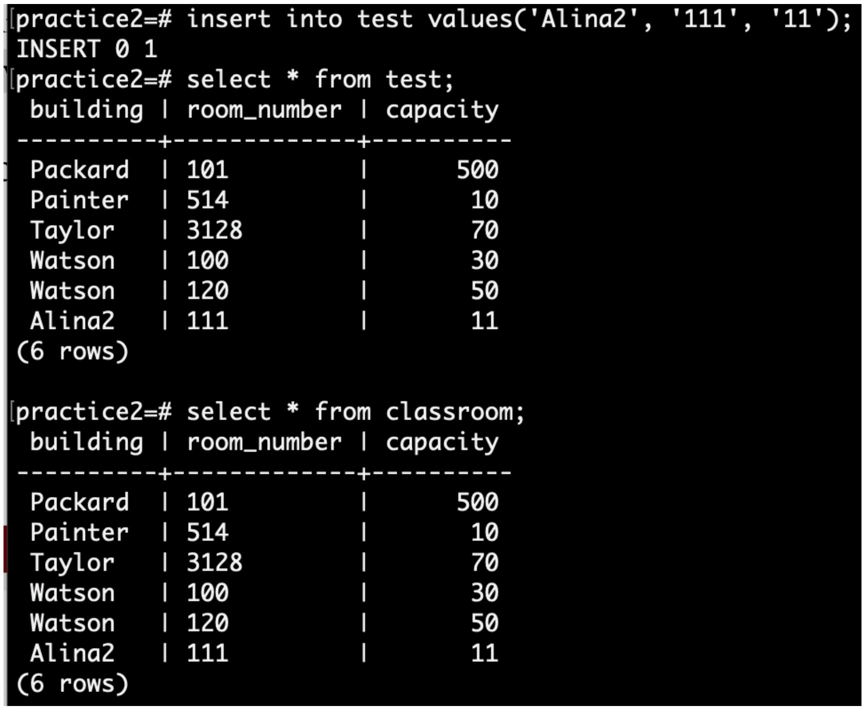
```
[practice2=# insert into classroom values('Alina', '777', '77');
INSERT 0 1
[practice2=# select * from test;
 building | room_number | capacity
----------+-------------+----------
 Packard  | 101         |      500
 Painter  | 514         |       10
 Taylor   | 3128        |       70
 Watson   | 100         |       30
 Watson   | 120         |       50
 Alina    | 777         |       77
(6 rows)
```

- Step 3: Update the view by deleting the newly added tuple and adding a new tuple into the view ‘test’; as we insert / delete the view ‘test’, the actual relation *classroom* updated as well, since *classroom* is the actual relation from which the view ‘test’ constructed from.



```
[practice2=# delete from test where building = 'Alina';
DELETE 1
[practice2=# select * from test;
 building | room_number | capacity
----------+-------------+----------
 Packard  | 101         |      500
 Painter  | 514         |       10
 Taylor   | 3128        |       70
 Watson   | 100         |       30
 Watson   | 120         |       50
(5 rows)

[practice2=# select * from classroom;
 building | room_number | capacity
----------+-------------+----------
 Packard  | 101         |      500
 Painter  | 514         |       10
 Taylor   | 3128        |       70
 Watson   | 100         |       30
 Watson   | 120         |       50
(5 rows)
```



```
[practice2=# insert into test values('Alina2', '111', '11');
INSERT 0 1
[practice2=# select * from test;
 building | room_number | capacity
----------+-------------+----------
 Packard  | 101         |      500
 Painter  | 514         |       10
 Taylor   | 3128        |       70
 Watson   | 100         |       30
 Watson   | 120         |       50
 Alina2   | 111         |       11
(6 rows)

[practice2=# select * from classroom;
 building | room_number | capacity
----------+-------------+----------
 Packard  | 101         |      500
 Painter  | 514         |       10
 Taylor   | 3128        |       70
 Watson   | 100         |       30
 Watson   | 120         |       50
 Alina2   | 111         |       11
(6 rows)
```