Homework 2

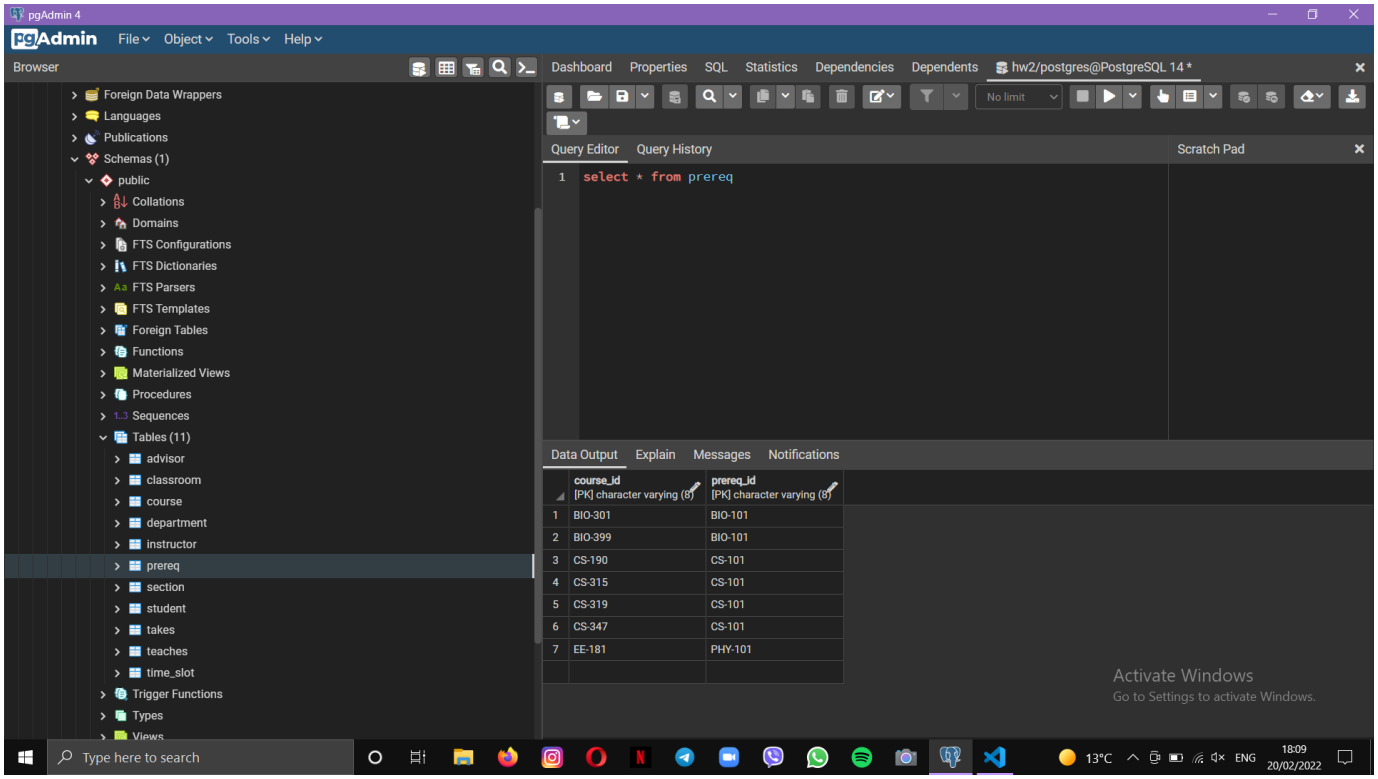
by Mher Movsisyan

1. Postgres Data Setup (5 pts)

Use the SQL scripts and set up sample data in your local installation of PostgreSQL Server.

- https://www.db-book.com/university-lab-dir/sample_tables-dir/DDL.sql
- https://www.db-book.com/university-lab-dir/sample_tables-dir/smallRelations/smallRelationsInsertFile.sql

When set up is complete, connect to your database via pgAdmin, select some data to make sure that the data setup is complete. Take a screenshot (it has to be a screenshot, not a photo using the phone) and include it in your answers.



2. SQL Queries (5 pts)

(a) Find the title of all the courses that may be taken after CS-101

In [ ]:

```
select title
from course
where course_id in (
    select course_id
    from prereq
    where prereq_id = 'CS-101'
);
```

Output:

title
Game Design
Robotics
Image Processing
Database System Concepts

(b) Find the instructor names that teach at least one course in Spring 2018.

In [ ]:
```
select name
from instructor
where id in (
    select distinct id
    from teaches
    where year = '2018'
    and semester = 'Spring'
);
```

Output:

name
Srinivasan
Wu
Mozart
El Said
Katz
Brandt

(c) Find the name of all the students that have taken any course that “Shankar” (student with ID 12345) has taken.

In [ ]:
```
select name
from student
where id in (
    select distinct id
    from takes
    where course_id in (
        select course_id
        from takes
        where id = '12345'));
```

Output:

name
Zhang
Shankar
Levy
Williams
Brown
Bourikas

3. Foreign Key Constraints (5 pts)

What is the difference between the DDL statements in (a) and (b)? Demonstrate how each one behaves when deleting data and include an screenshot of the results in your answers.

(a)

In [ ]:
```
DROP TABLE IF EXISTS table_b;
DROP TABLE IF EXISTS table_a;

CREATE TABLE table_a (
    col_a INTEGER NOT NULL,
    col_b VARCHAR(10) UNIQUE
);

CREATE TABLE table_b (
    col_c NUMERIC,
    col_d VARCHAR(10) REFERENCES table_a(col_b)
);
```

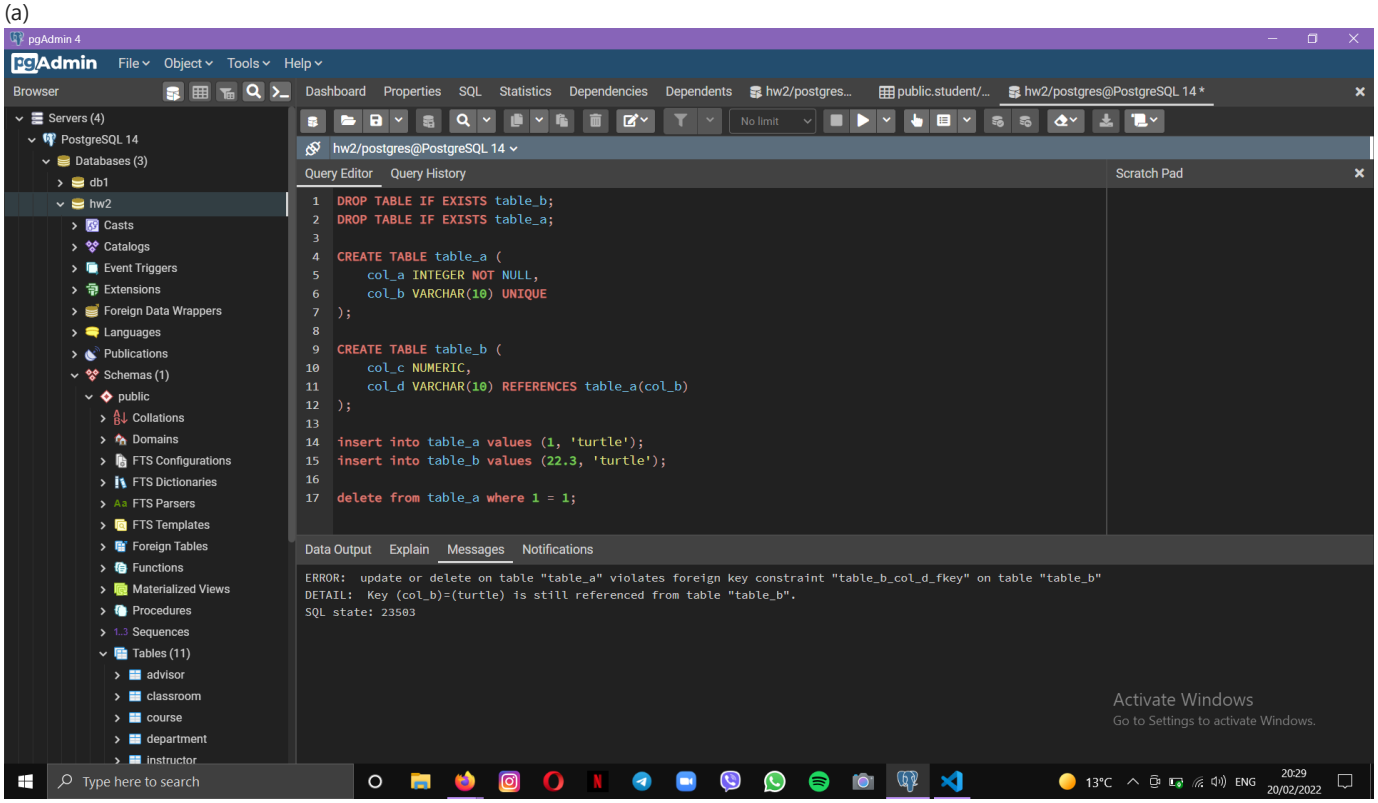
(b)

In [ ]:

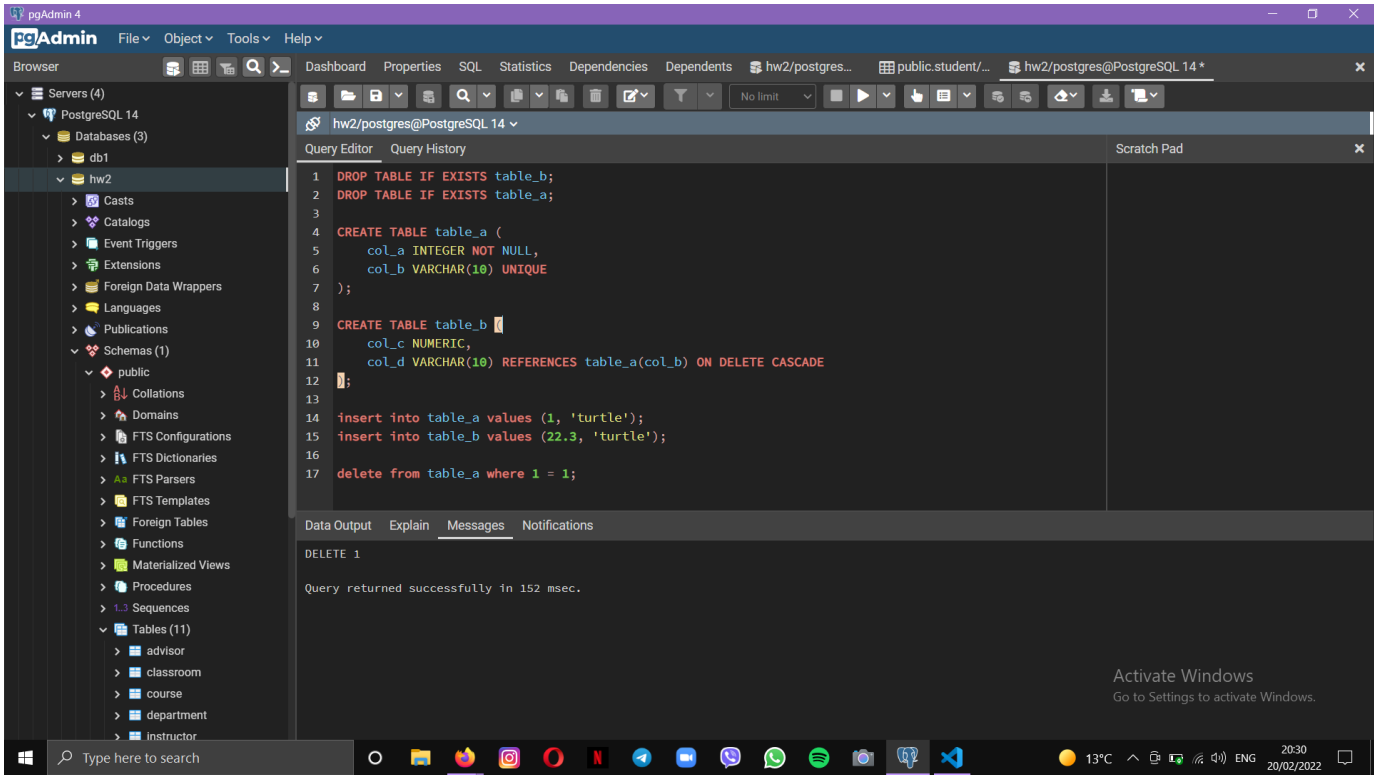
```
DROP TABLE IF EXISTS table_b;
DROP TABLE IF EXISTS table_a;

CREATE TABLE table_a (
    col_a INTEGER NOT NULL,
    col_b VARCHAR(10) UNIQUE
);

CREATE TABLE table_b (
    col_c NUMERIC,
    col_d VARCHAR(10) REFERENCES table_a(col_b) ON DELETE CASCADE
);
```

In case of (b), when we delete data from table_a, the data in table_b will be deleted as well. Yet, in case of (a), when we delete data from table_a, it will result in an error.

(a)

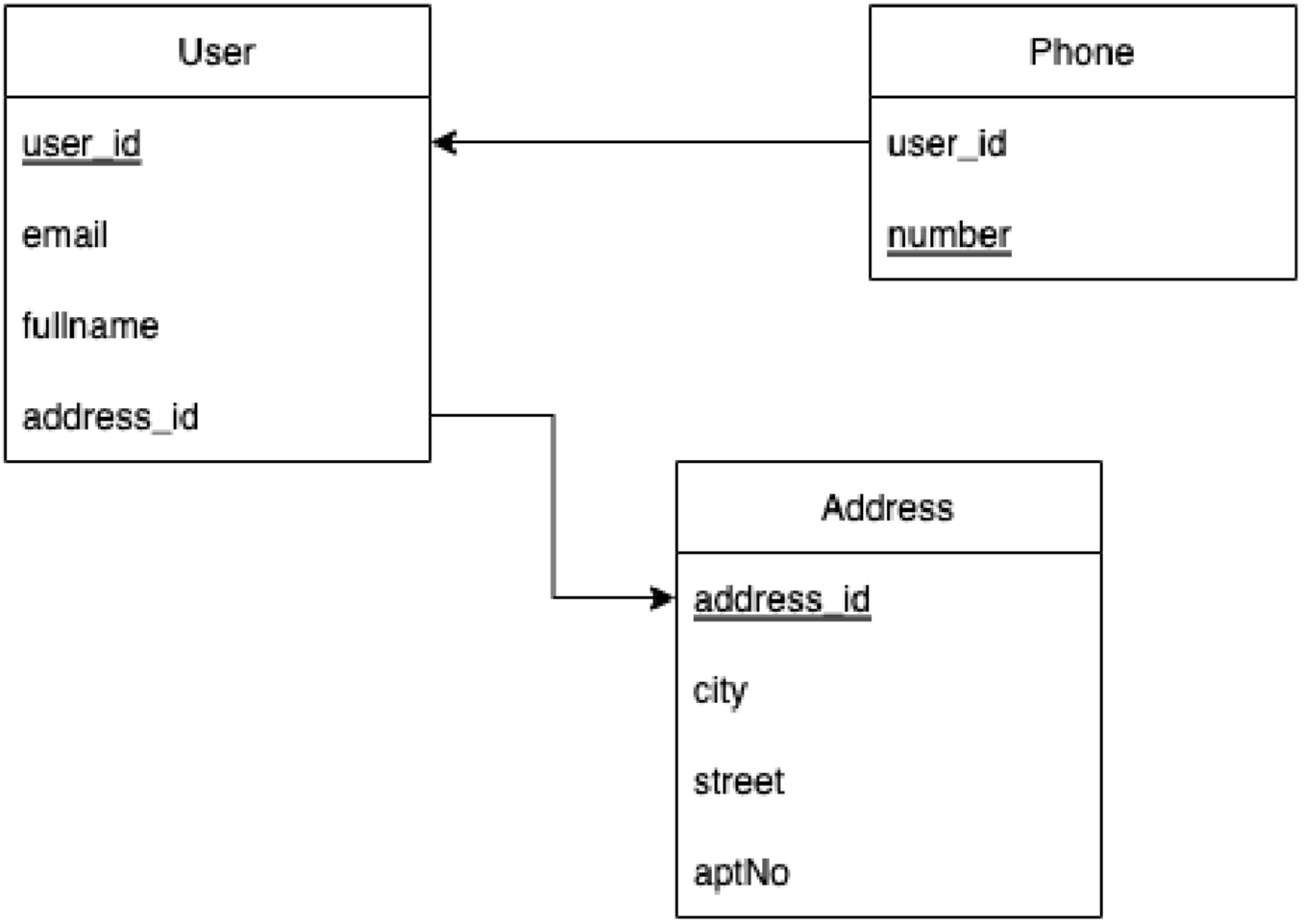


(b)



4. SQL DDL (5 pts)

Given the following relational schema:

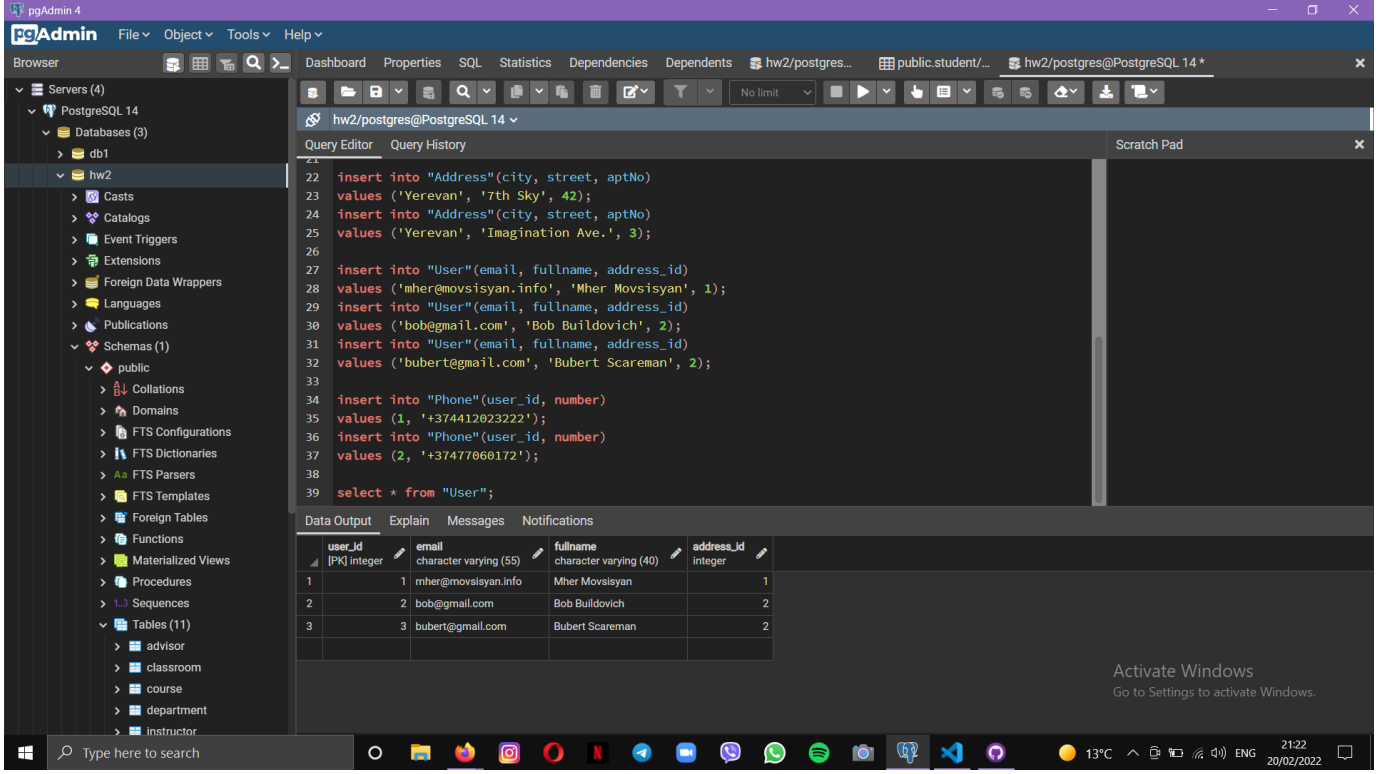


(a) Write SQL statements that create tables to implement relational schema. Make sure that primary, foreign key, and unique constraints are implemented as necessary.

Add at least three different users and two phone numbers for each user. Write the following queries and include a screenshot of the results in your answers.

In [ ]:

```
drop table if exists "User", "Phone", "Address";

create table "Address"(
    address_id serial primary key,
    city varchar(20),
    street varchar(40),
    aptNo int
);

create table "User"(
    user_id serial primary key,
    email varchar(55) unique,
    fullname varchar(40),
    address_id serial references "Address"(address_id)
);

create table "Phone"(
    user_id serial not null references "User"(user_id),
    number varchar(16) primary key
);

insert into "Address"(city, street, aptNo)
values ('Yerevan', '7th Sky', 42);
insert into "Address"(city, street, aptNo)
values ('Yerevan', 'Imagination Ave.', 3);

insert into "User"(email, fullname, address_id)
values ('mher@movsisyan.info', 'Mher Movsisyan', 1);
insert into "User"(email, fullname, address_id)
values ('bob@gmail.com', 'Bob Buildovich', 2);
insert into "User"(email, fullname, address_id)
values ('bubert@gmail.com', 'Bubert Scareman', 2);

insert into "Phone"(user_id, number)
values (1, '+374412023222');
insert into "Phone"(user_id, number)
values (2, '+37477060172');
```

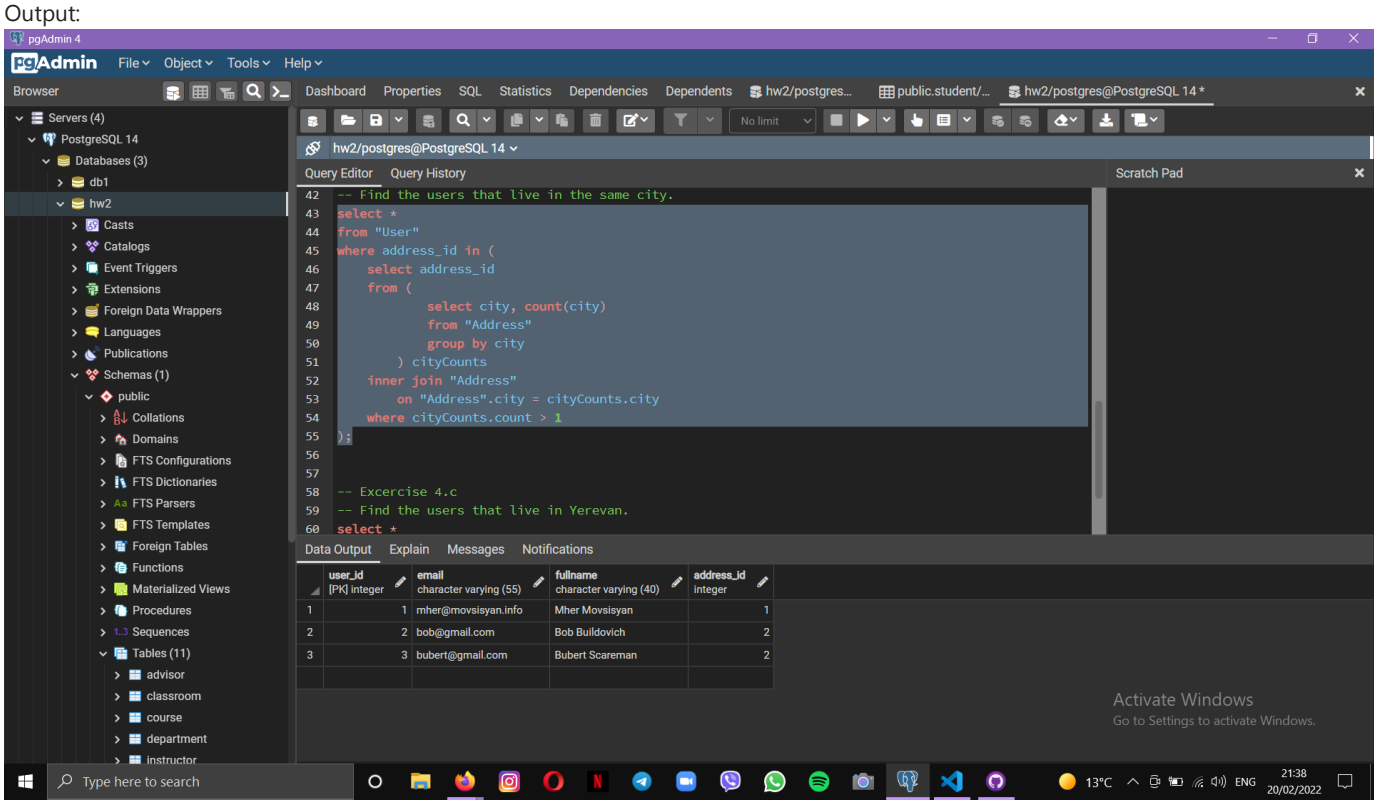


(b) Find the users that live in the same city.

In [ ]:

```
-- Exercise 4.b
-- Find the users that live in the same city.
select *
from "User"
where address_id in (
    select address_id
    from (
            select city, count(city)
            from "Address"
            group by city
        ) cityCounts
    inner join "Address"
        on "Address".city = cityCounts.city
    where cityCounts.count > 1
);
```

Output:



(c) Find the users that live in the city of Yerevan.

In [ ]:

```
-- Excercise 4.c
-- Find the users that live in Yerevan.
select *
from "User"
where "User".address_id in (
    select "Address".address_id
    from "Address"
    where "Address".city = 'Yerevan'
);
```

Output: