1. Write the following SQL queries. (5 points each)

a. Find the titles of courses in the Comp. Sci. department that have 3 credits.

select title

from course c

where credits = 3 and dept\_name = 'Comp. Sci.'

b. Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result.

select s.ID

from (student s join takes t using(ID))

join (instructor I join teaches t1 using(ID))

using(course\_id, sec\_id, semester, year)

where i.name = 'Einstein'

c. Find the highest salary of any instructor.

select max(salary)

from instructor

d. Find all instructors earning the highest salary (there may be more than one with the same salary).

select name

from instructor

where salary = (select max(salary)

from instructor

)

e. Find the enrollment of each section that was offered in Fall 2009. Remember just a number is meaningless unless you include what it refers to

select t.course\_id, t.sec\_id, count(ID)

from section s natural join takes t

where semester = 'Fall' and year = 2009

group by t.course\_id, t.sec\_id

f. Find the maximum enrollment, across all sections, in Fall 2009

select max(enroll)

from (select count(ID) as enroll

from section s natural join takes t

where semester = 'Fall'

and year = 2009

group by s.course\_id, s.sec\_id) as enroll

g. Find the sections that had the maximum enrollment in Fall 2009. Make sure you use a WITH clause

with A1 as (

select t.course\_id, t.sec\_id, count(ID) as enroll

from section s natural join takes t

where semester = 'Fall'

and year = 2009

group by t.course\_id, t.sec\_id),

A2 as(

select A1.course\_id, A1.sec\_id

from A1

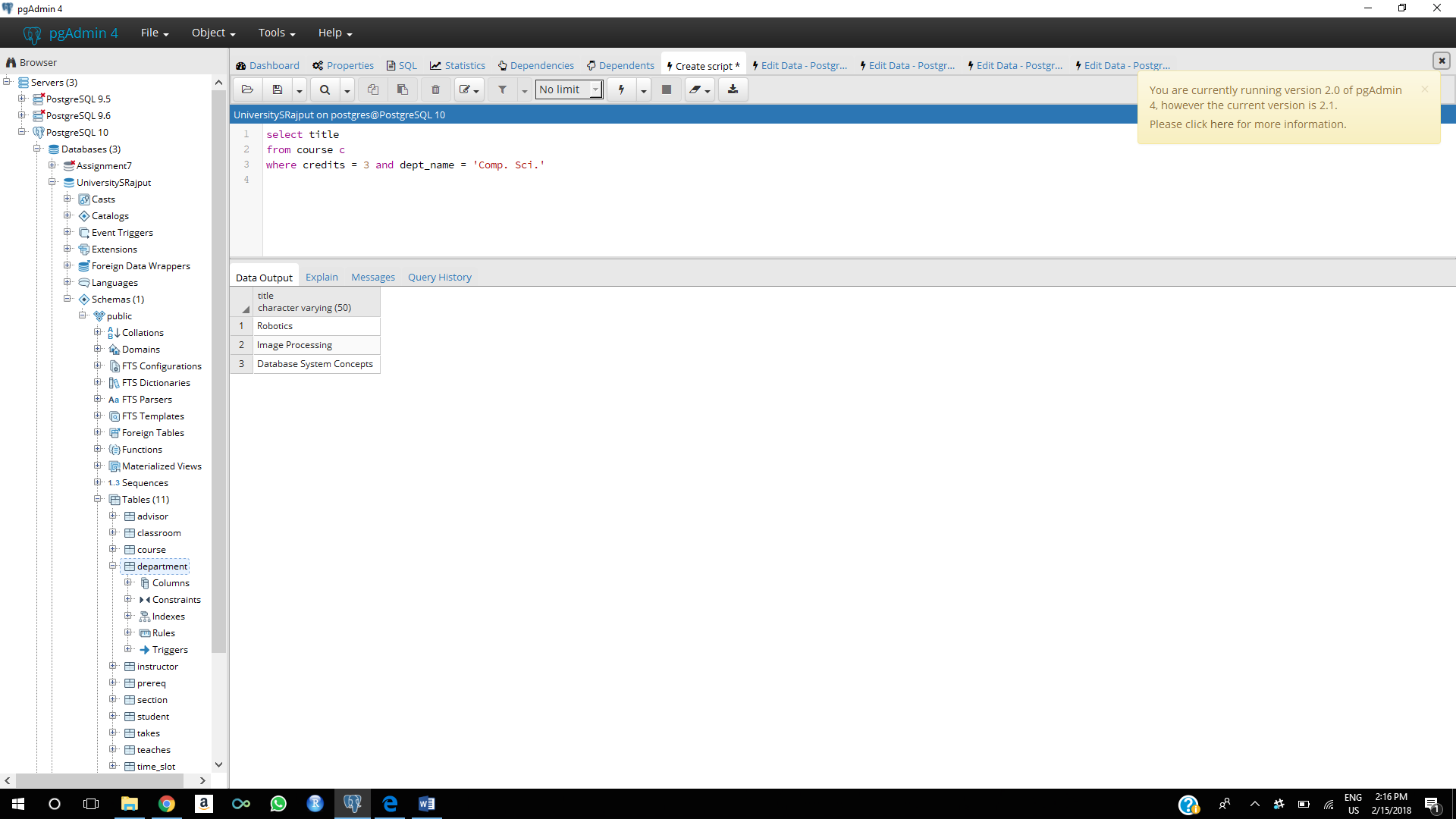
where enroll = (select max(enroll) from A1)

)

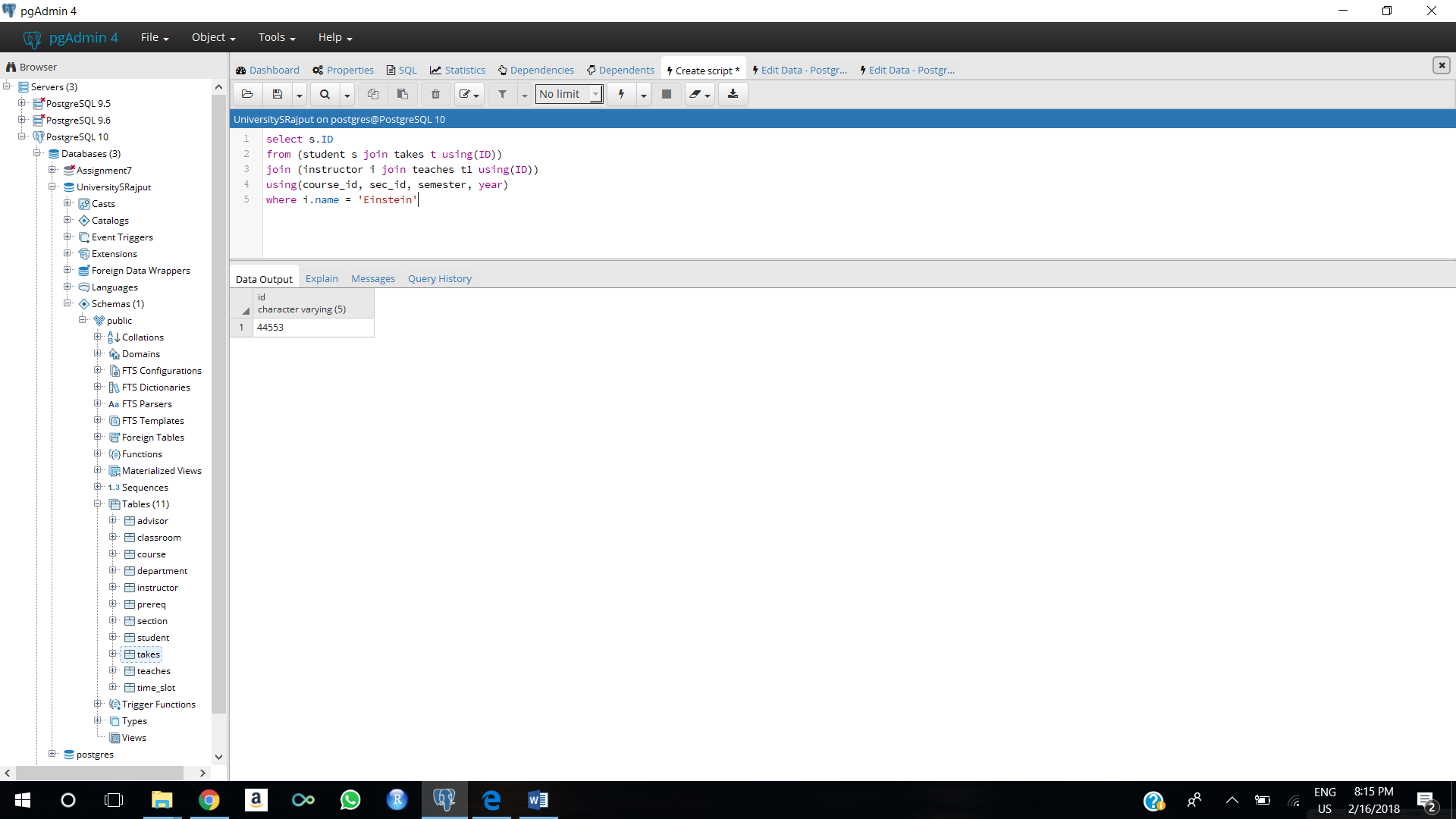
select \* from A2

2. Now run these queries on your university database and submit the screen shots of each answer. If the answer does not match what you think it should be, correct the query and try again. This is a way for you to learn from your mistakes. You can submit each one separately but make sure you label them appropriately. Otherwise paste them all into one Word document and submit that. Make sure I can see all rows and columns for the output. You might have to make the result window full screen. (2 points each)

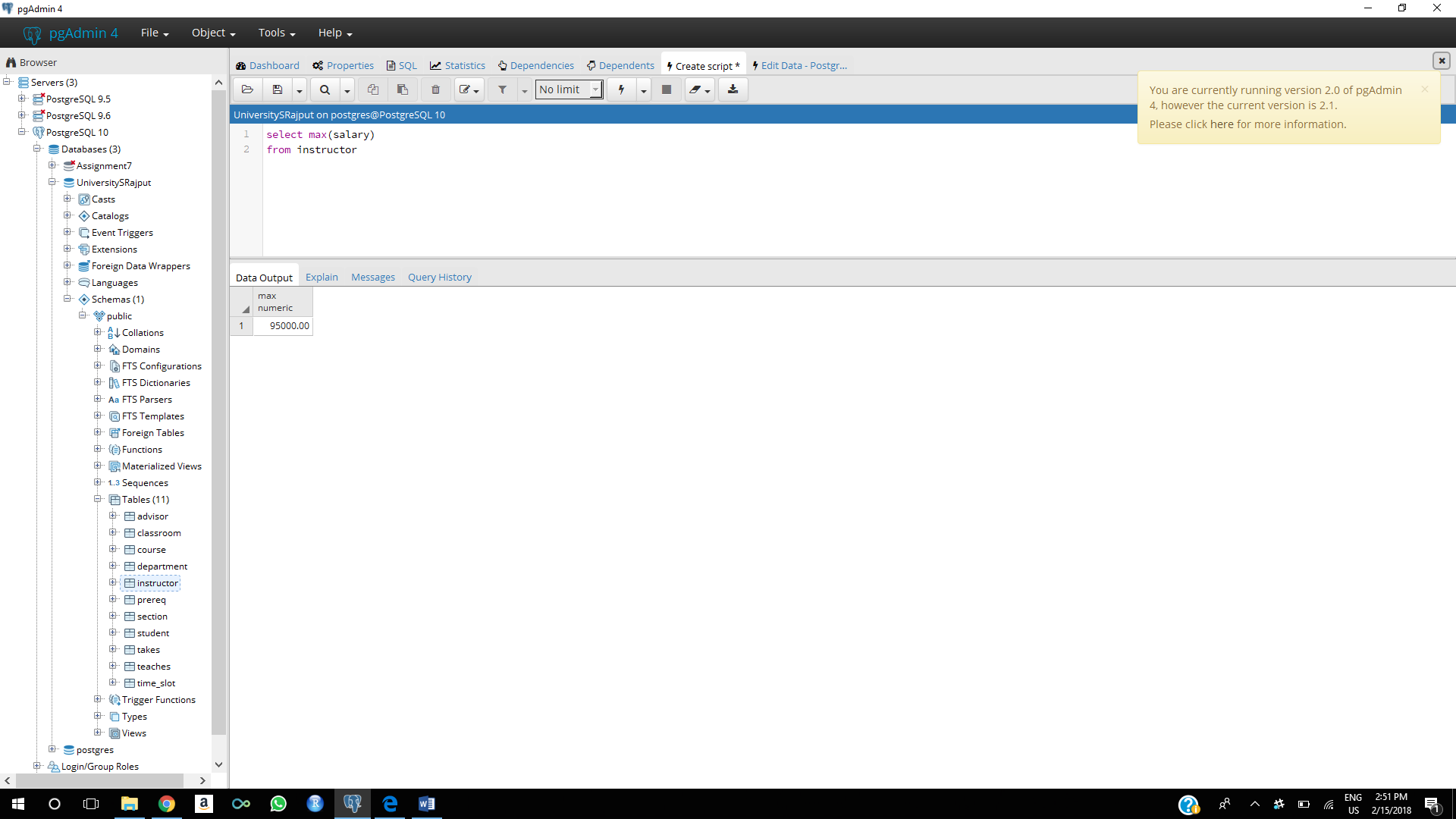
a. Find the titles of courses in the Comp. Sci. department that have 3 credits.



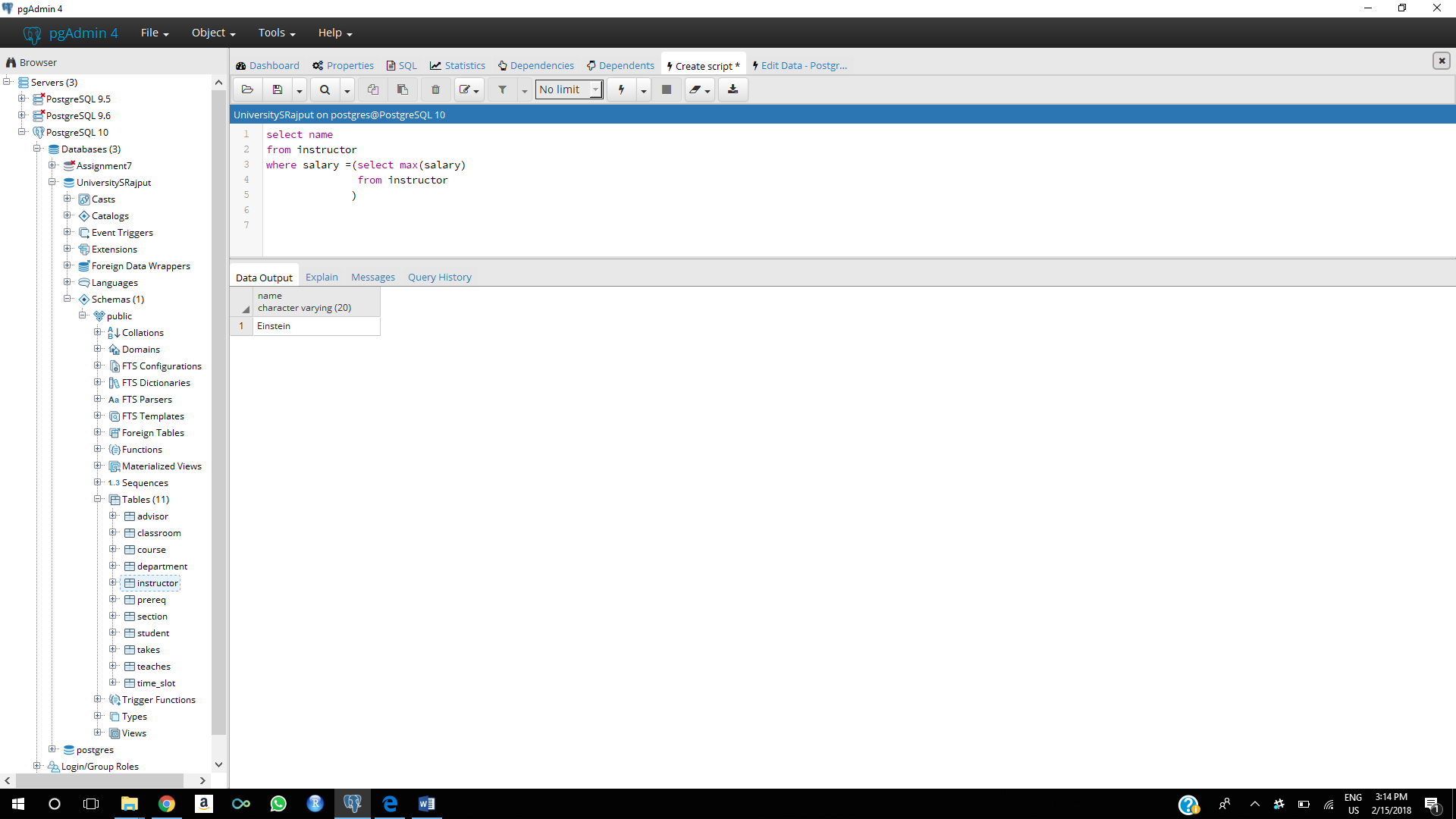
b. Find the IDs of all students who were taught by an instructor named Einstein; make sure there are no duplicates in the result.



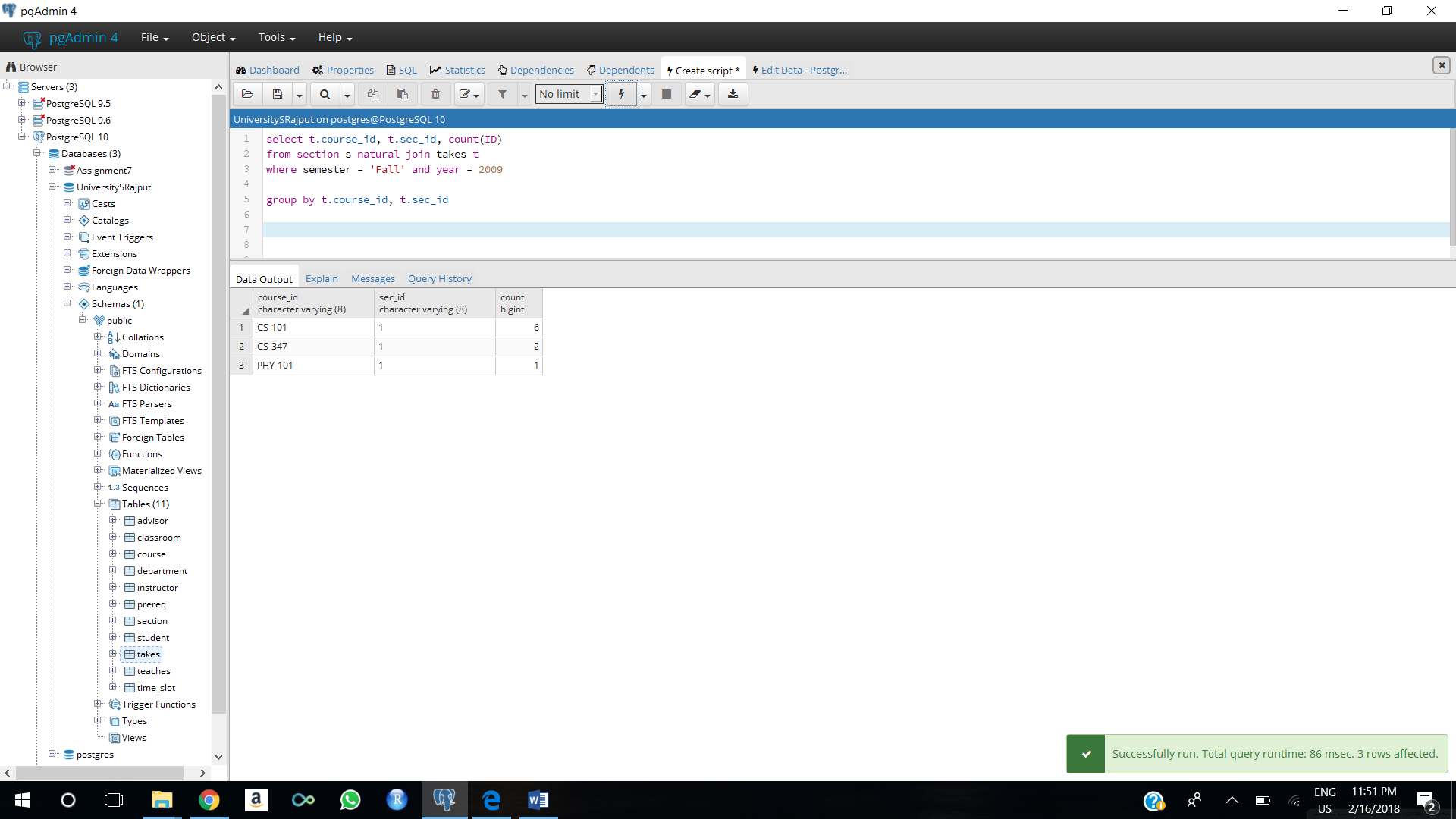
c. Find the highest salary of any instructor.

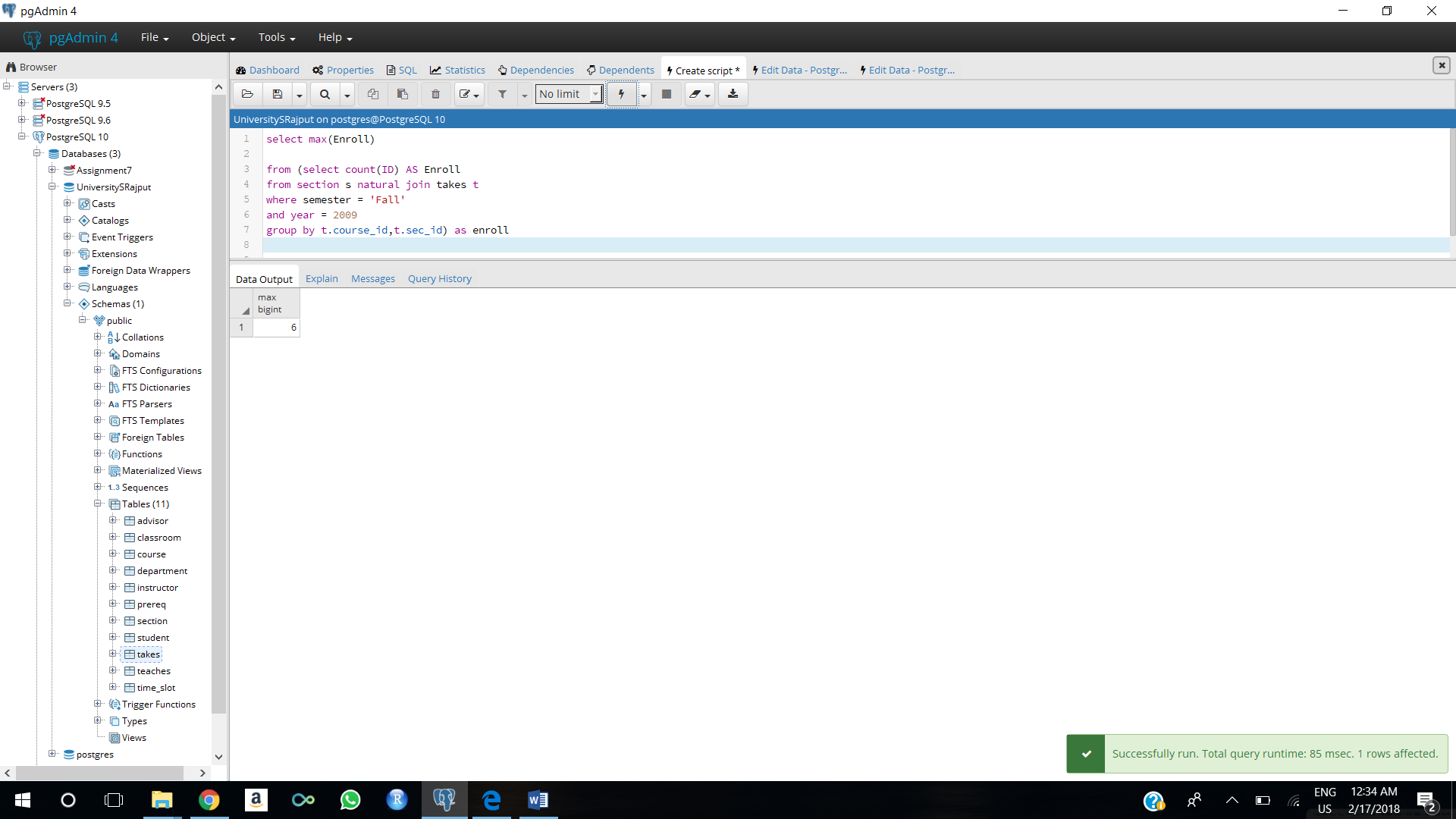


d. Find all instructors earning the highest salary (there may be more than one with the same salary).

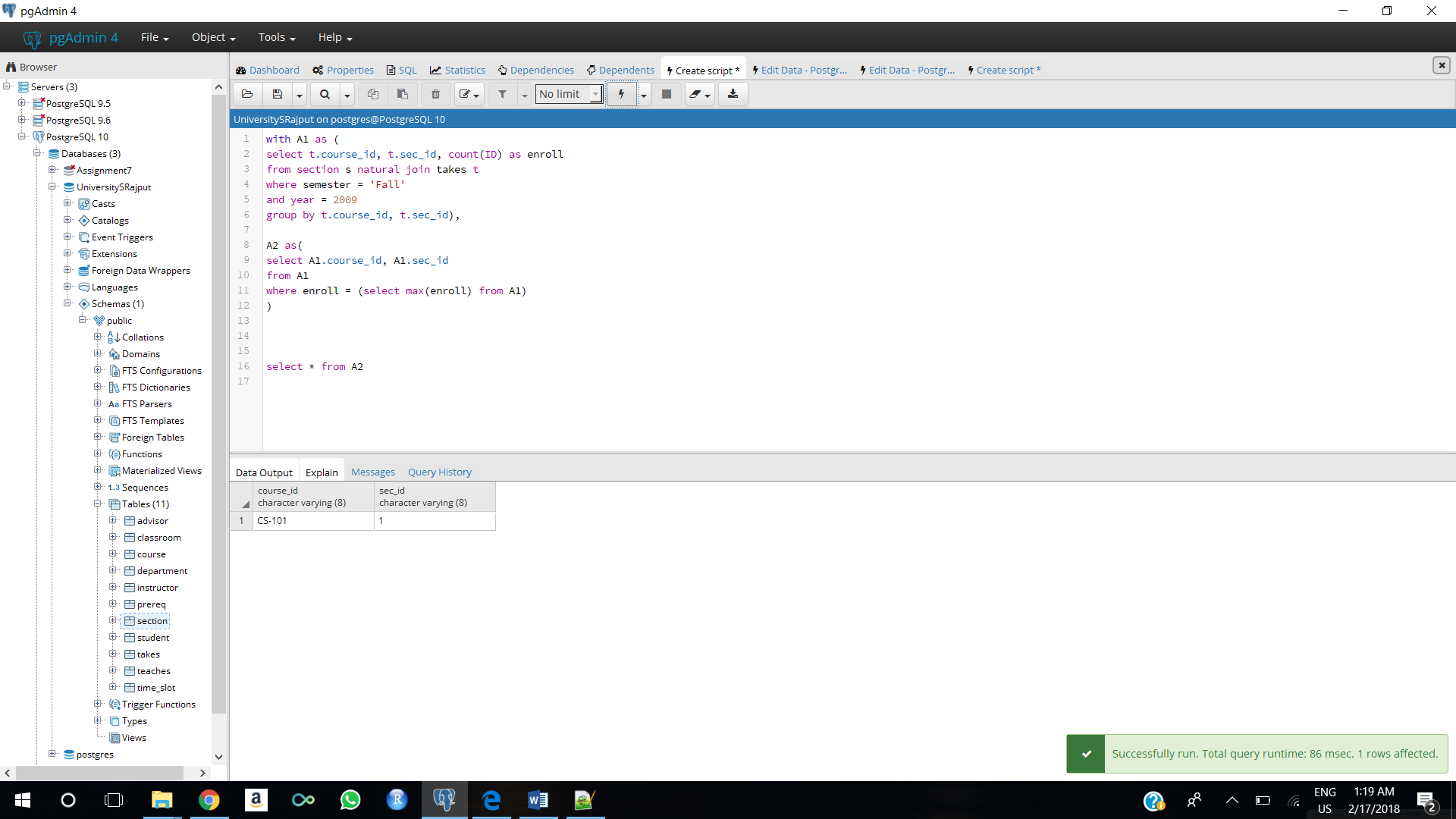


e. Find the enrollment of each section that was offered in Fall 2009. Remember just a number is meaningless unless you include what it refers to



f. Find the maximum enrollment, across all sections, in Fall 2009. 

g. Find the sections that had the maximum enrollment in Fall 2009. Make sure you use a WITH clause



3. Write the statements to insert 3 records into the student relation. Test them on our database. If they do not work, tell me why. If they work, tell me what happened. (5 points each – 3 points for insertion statement and 2 points for explanation of what happened)

a. Insert a student with your last name into the Finance department with an id of 00005 and 0 credits.

insert into student(id,name,dept\_name,tot\_cred)

values(00005,'Rajput','Finance',0);

Query worked properly. Record inserted

b. Insert a student with my last name into the Comp. Sci. department with an id of 12345 and null credits.

insert into student(id,name,dept\_name,tot\_cred)

values (12345,'Schwartz','Comp. Sci.',null);

The primary key already exists, it shows error of duplicate key value violates unique constraint "student\_pkey". We need to change the primary\_key value to add the record.

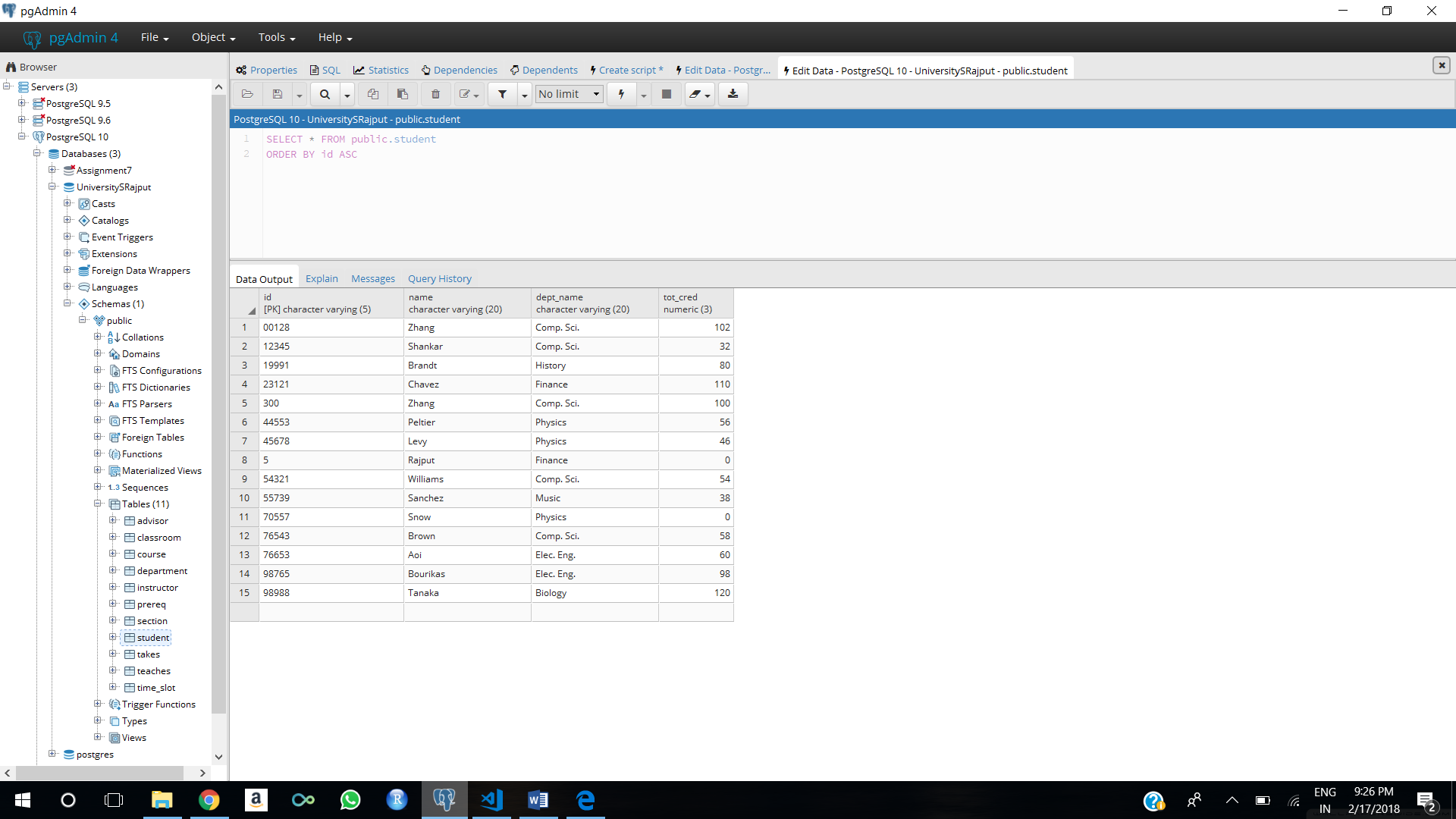
c. Insert a student with the last name of Zhang into the Comp. Sci. department with an id of 00300 and 100 total credits

insert into student(id,name,dept\_name,tot\_cred)

values(00300,'Zhang','Comp. Sci.',100);

Query worked properly. Record inserted

4. Show me a screen shot of the contents of the student relation. Make sure it shows all of the content and is large enough for me to read without a magnifying glass. (5 points)



5. Write but do NOT execute the following queries. (3 points each)

a. Delete the contents of the prereq table.

delete from prereq;

b. Delete student 00300 from the student table.

delete from student

where id = ‘00300’;

6. Find any department whose name contains all of the letters ‘psy ’in this order but be careful because you want to include departments that might start with those letters and the first one would be capitalized. See if you can write it with only one WHERE predicate. If not, I will accept two. (10 points)

select dept\_name

from department

where dept\_name

like'%psy%' or dept\_name like'%Psy%'

7. Find the names and major departments of all straight A students. This does not include students with A- as even one grade. We do not give A+. A course that has not yet had a grade assigned should be ignored as the grade does not affect the GPA. (10 points)

select s.name, s.dept\_name

from student s natural join takes t

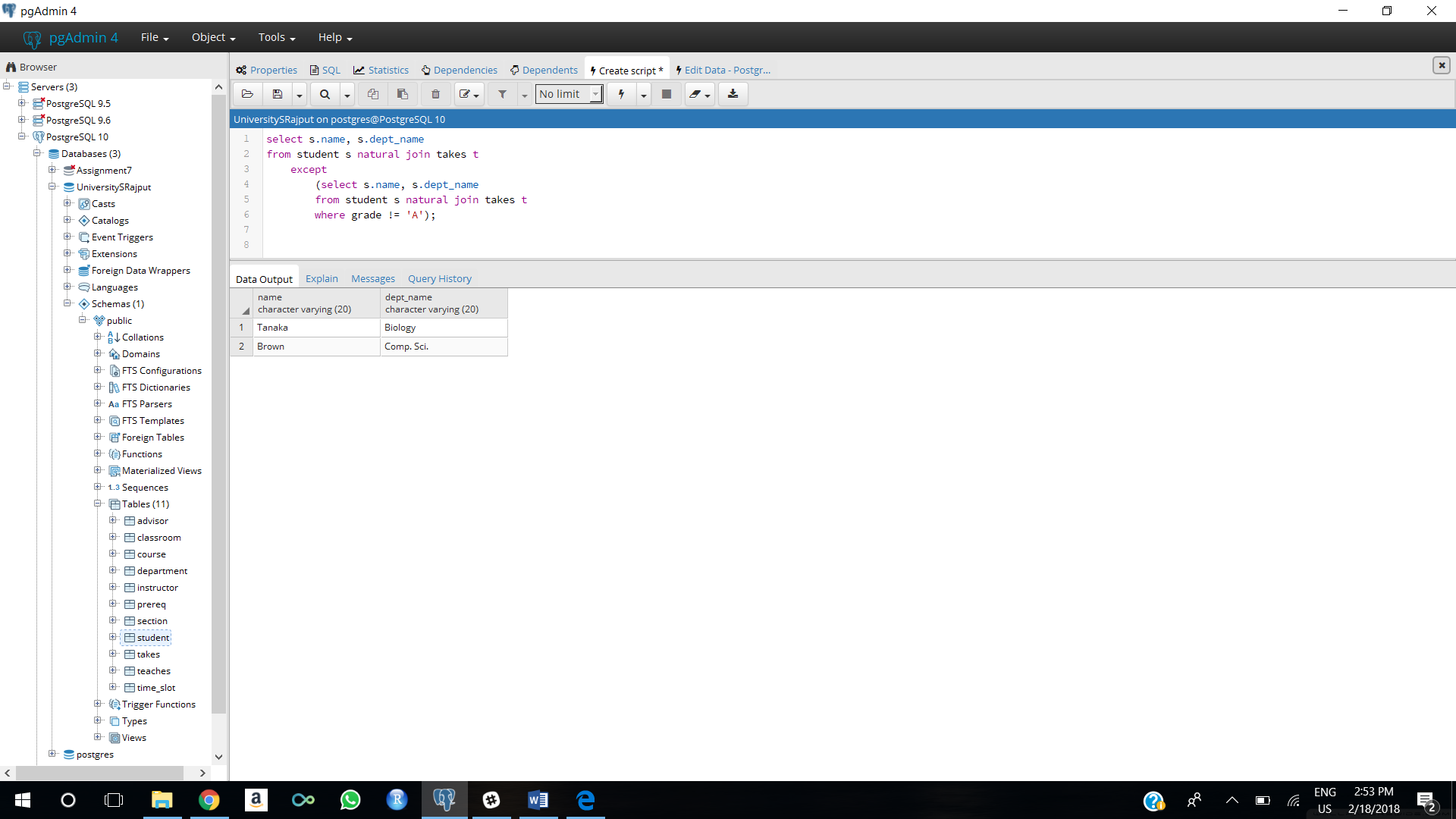
except

(select s.name, s.dept\_name

from student s natural join takes t

where grade != 'A');

8.Run the statement from 7 in our database and explain the results. (5 points)



This statement selects student names and department name who have A grade in all course. If any course is not graded, then but other in courses student has A grade he will be selected and if any student who didn’t have A grade will not be selected.