**Домашнее задание №8**

a.

create or replace function getStudentHierarchy(p\_manager\_id integer)

returns table ("студент" character varying, "роль" character varying,

"уровень" integer, "путь от лидера" text) as $$

begin

------------------------------------

return query WITH RECURSIVE \_manager AS

(

SELECT manager.id,

manager\_id,

student.name,

1 AS level,

array [student.name] AS path,

student\_id

FROM manager

join student on manager.student\_id = student.id

WHERE manager\_id = p\_manager\_id or (manager\_id is null and p\_manager\_id is null)

UNION

SELECT m1.id,

m1.manager\_id,

student.name,

level + 1 AS level,

path || student.name AS path,

m1.student\_id

FROM manager AS m1

join student on m1.student\_id = student.id

INNER JOIN \_manager AS m2 ON m1.manager\_id = m2.id

)

SELECT student.name as "студент",

role as "роль",

level as "уровень",

array\_to\_string (path, '->') as "путь от лидера"

FROM \_manager

join student\_role on student\_role.svyaz\_manager\_id = \_manager.id

join student on \_manager.student\_id = student.id

order by student.id desc;

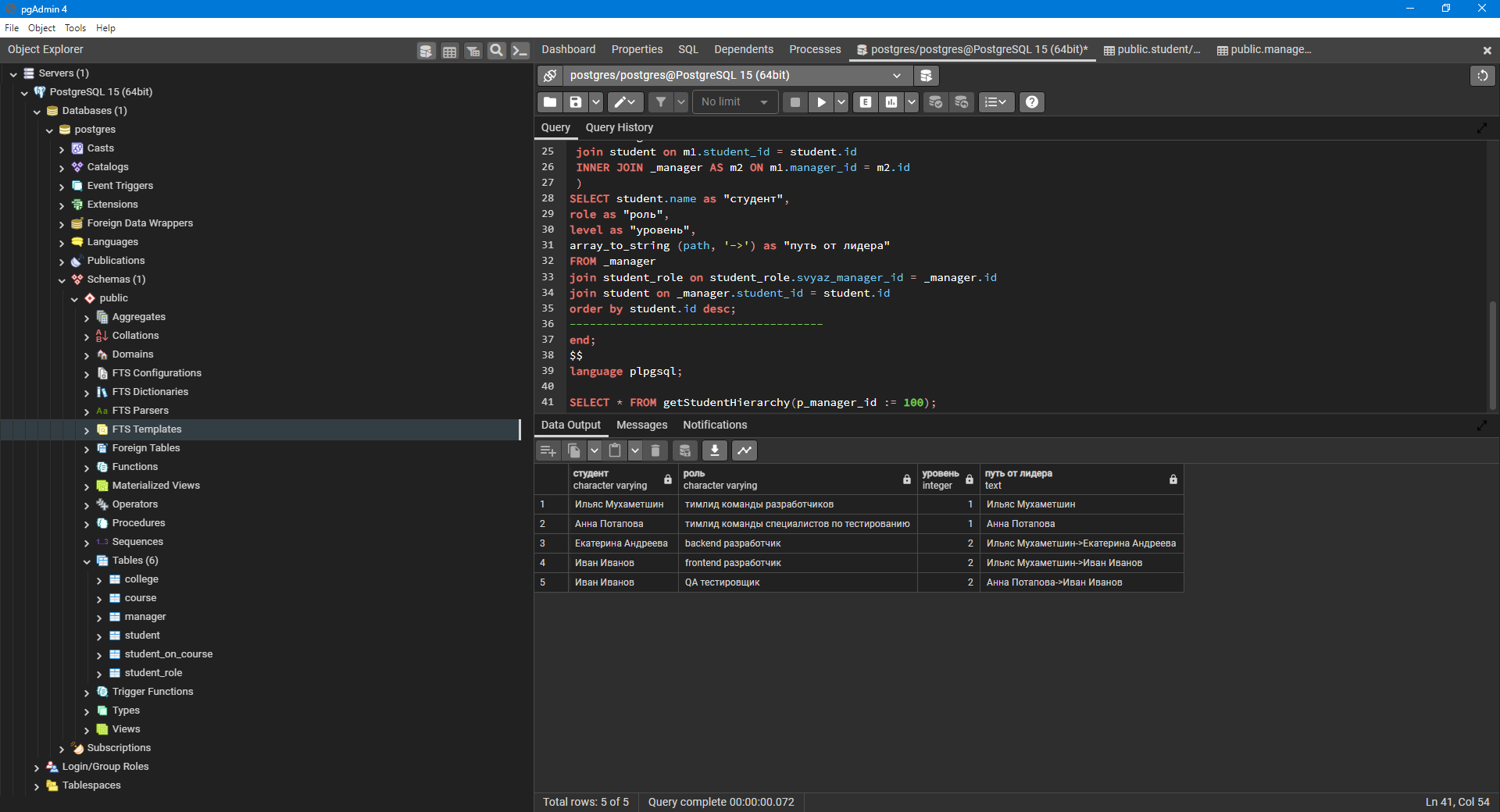
--------------------------------------

end;

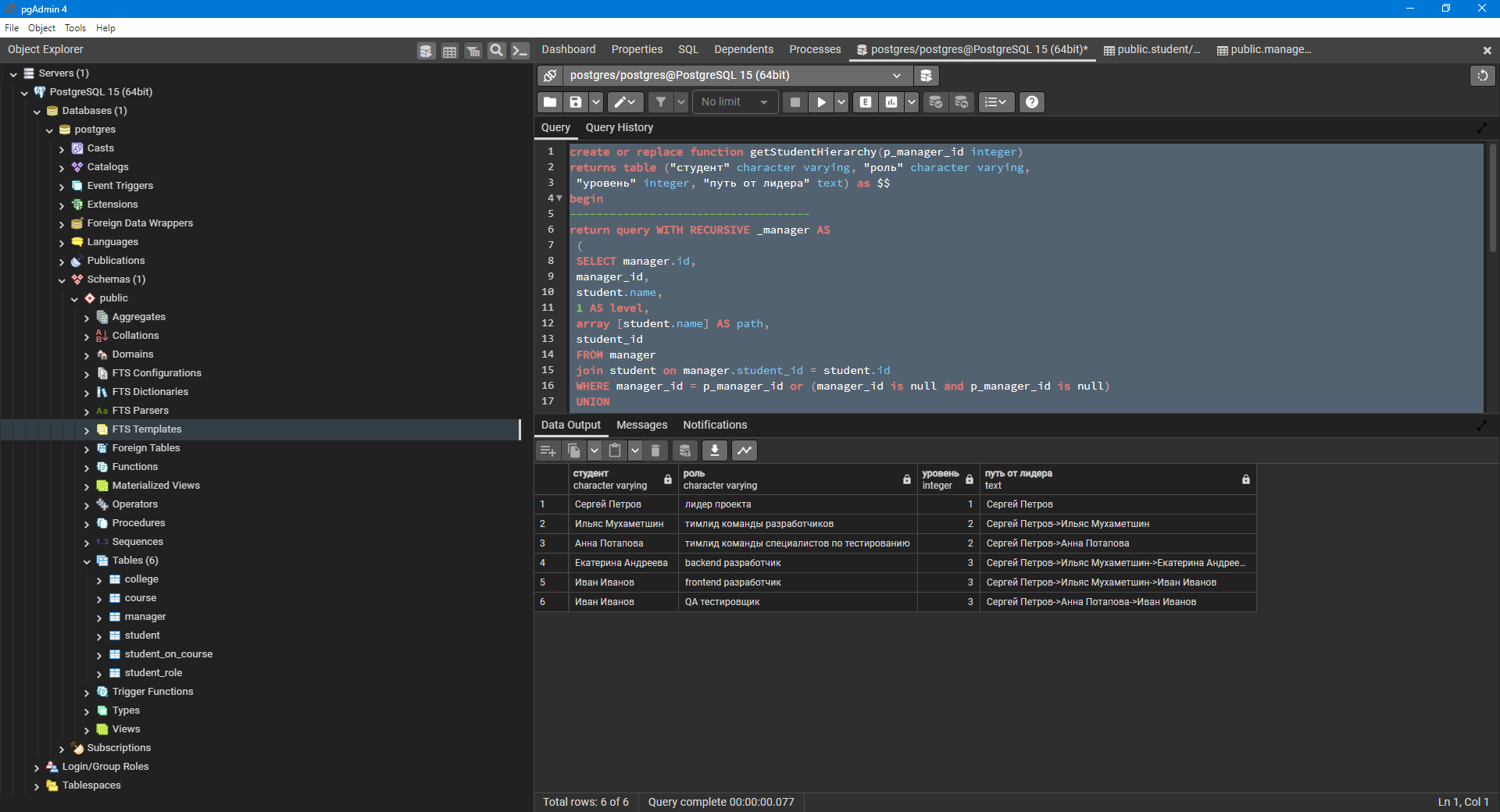
$$

language plpgsql;

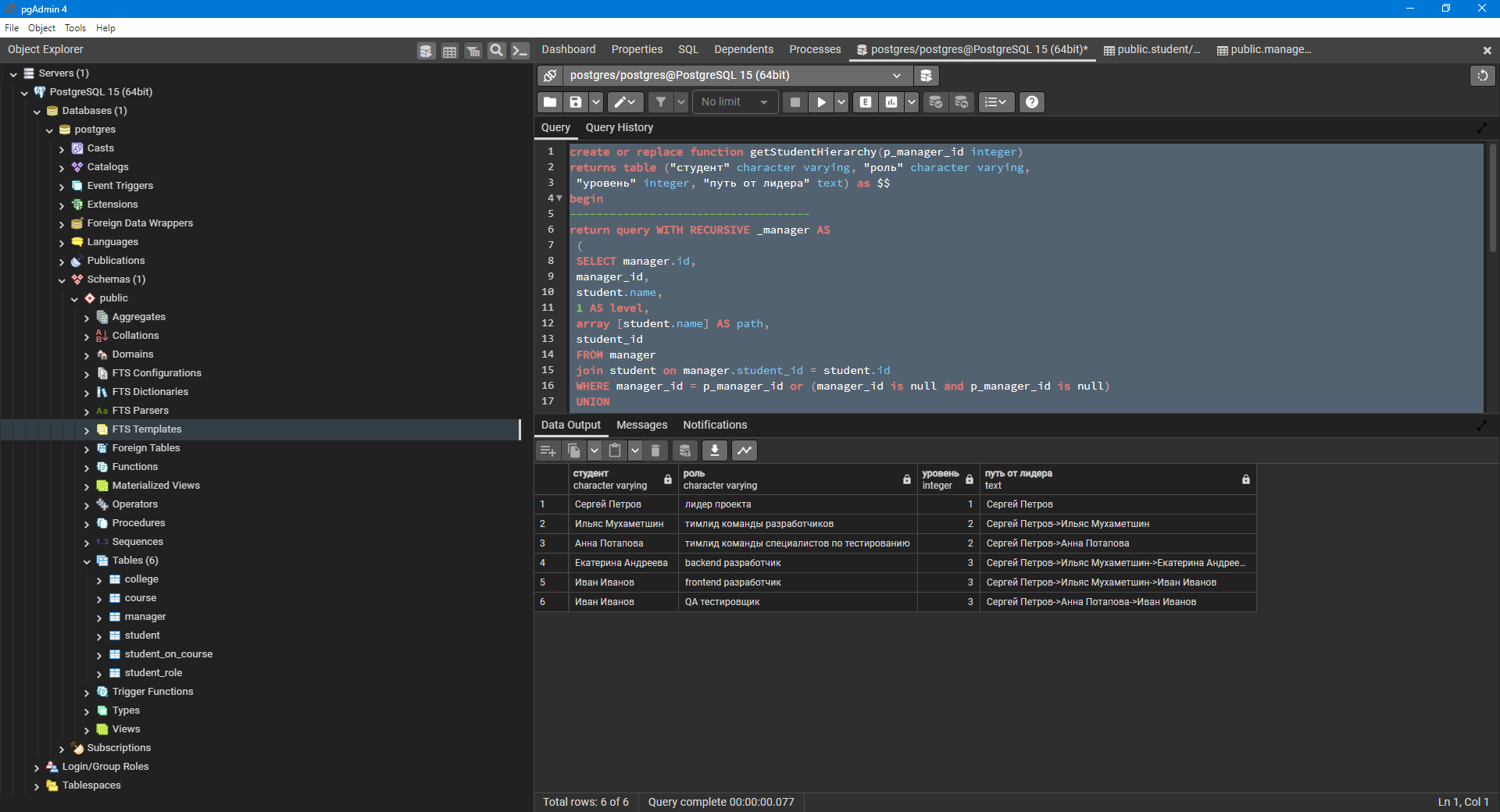
SELECT \* FROM getStudentHierarchy(p\_manager\_id := 100);



SELECT \* FROM getStudentHierarchy(p\_manager\_id := null);



SELECT \* FROM getStudentHierarchy();



b.

create or replace function getStudentHierarchyByLevel(p\_manager\_id integer default null, p\_level integer default null)

returns table ("студент" character varying, "роль" character varying,

"уровень" integer, "путь от лидера" text) as $$

begin

------------------------------------

return query WITH RECURSIVE \_manager AS

(

SELECT manager.id,

manager\_id,

student.name,

1 AS level,

array [student.name] AS path,

student\_id

FROM manager

join student on manager.student\_id = student.id

WHERE manager\_id = p\_manager\_id or (manager\_id is null and p\_manager\_id is null)

UNION

SELECT m1.id,

m1.manager\_id,

student.name,

level + 1 AS level,

path || student.name AS path,

m1.student\_id

FROM manager AS m1

join student on m1.student\_id = student.id

INNER JOIN \_manager AS m2 ON m1.manager\_id = m2.id

)

SELECT student.name as "студент",

role as "роль",

level as "уровень",

array\_to\_string (path, '->') as "путь от лидера"

FROM \_manager

join student\_role on student\_role.svyaz\_manager\_id = \_manager.id

join student on \_manager.student\_id = student.id

where level = p\_level

order by student.id desc;

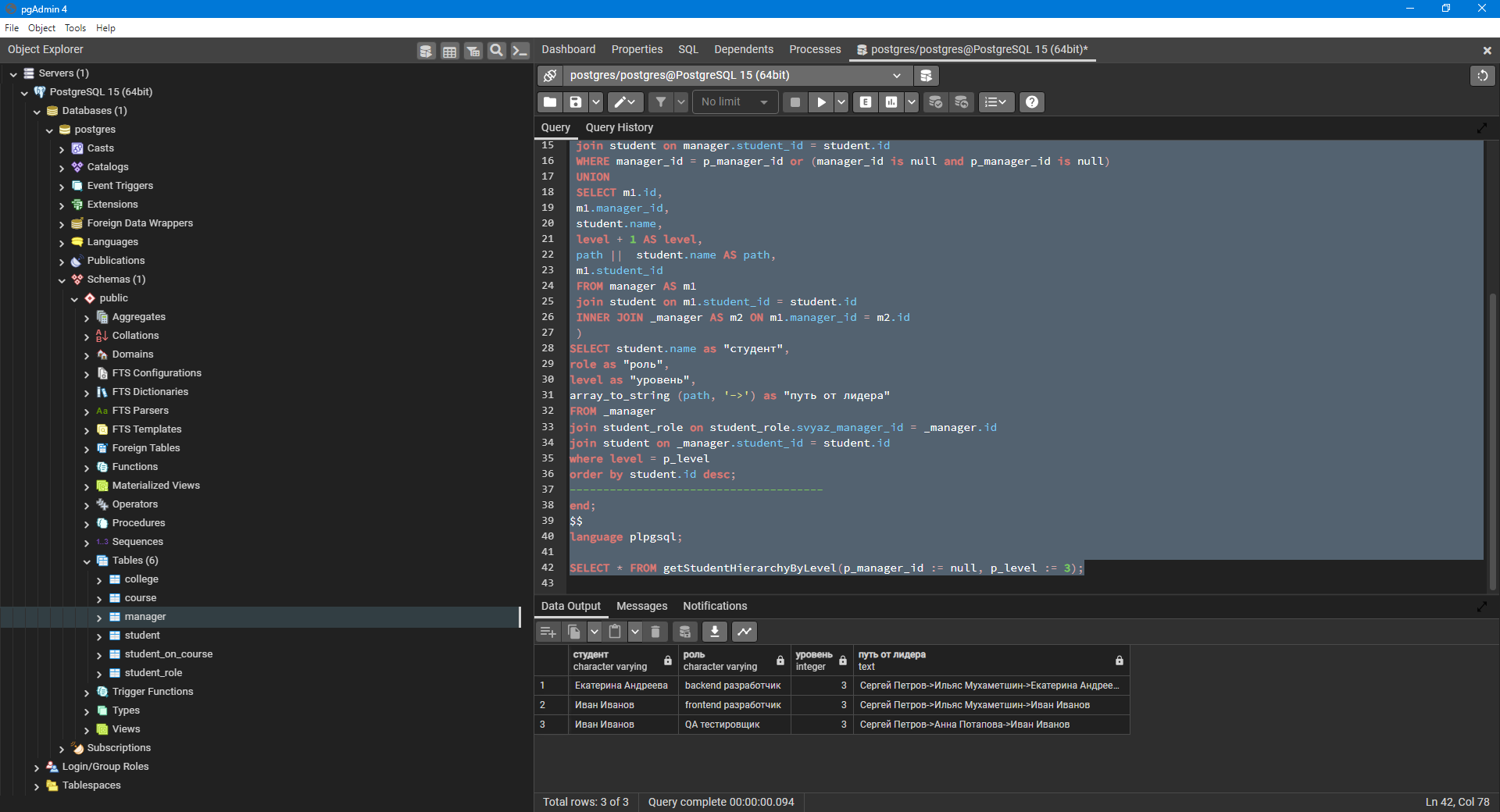
--------------------------------------

end;

$$

language plpgsql;

SELECT \* FROM getStudentHierarchyByLevel(p\_manager\_id := null, p\_level := 3);



SELECT \* FROM getStudentHierarchyByLevel(p\_manager\_id := 100, p\_level := 1);

