**Домашнее задание №9**

a.

create or replace function fnc\_trg\_check\_permissions()

returns trigger as $$

begin

declare

some\_date timestamp := timestamp '2001-02-16 20:38:40';

current\_day\_of\_week int := extract(dow from some\_date);

current\_hour int := extract(hour from some\_date);

begin

if current\_day\_of\_week >= 1 and current\_day\_of\_week <= 5

and current\_hour >= 9 and current\_hour <= 18 then

return new;

else

raise exception 'Вы не можете выполнить изменение данных в нерабочее время';

end if;

end;

end; $$

language plpgsql;

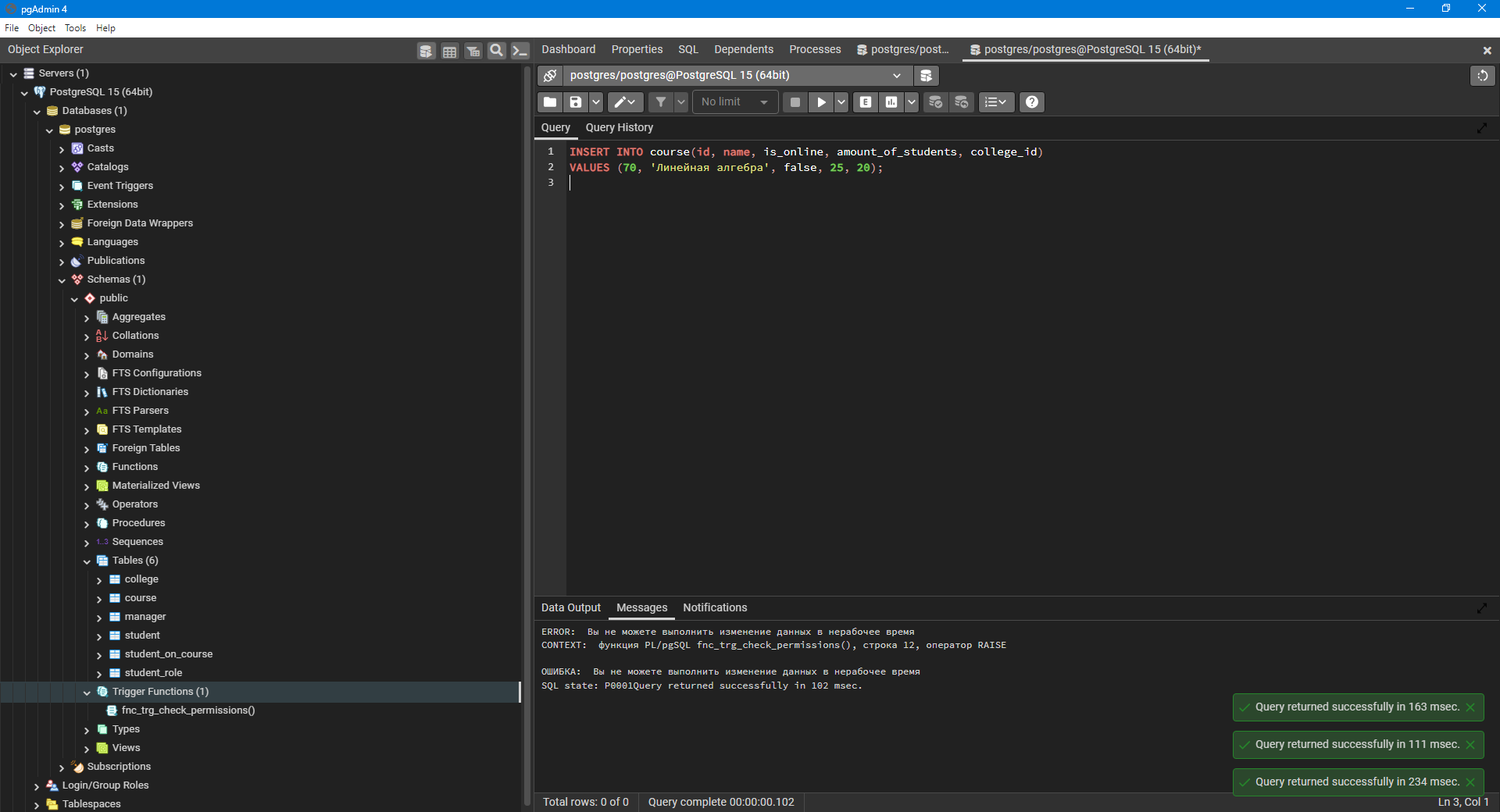
create trigger trg\_check\_permissions

before insert or update or delete on course

for each statement execute function fnc\_trg\_check\_permissions();

INSERT INTO course(id, name, is\_online, amount\_of\_students, college\_id)

VALUES (70, 'Линейная алгебра', false, 25, 20);



b.

create or replace function fnc\_trg\_check\_contact ()

returns trigger as $$

begin

if new.telegram\_contact is NULL then

new.telegram\_contact = '@default\_contact';

end if;

return new;

end; $$

language plpgsql;

create trigger trg\_check\_contact

before insert or update on student

for EACH ROW execute function fnc\_trg\_check\_contact();

UPDATE student

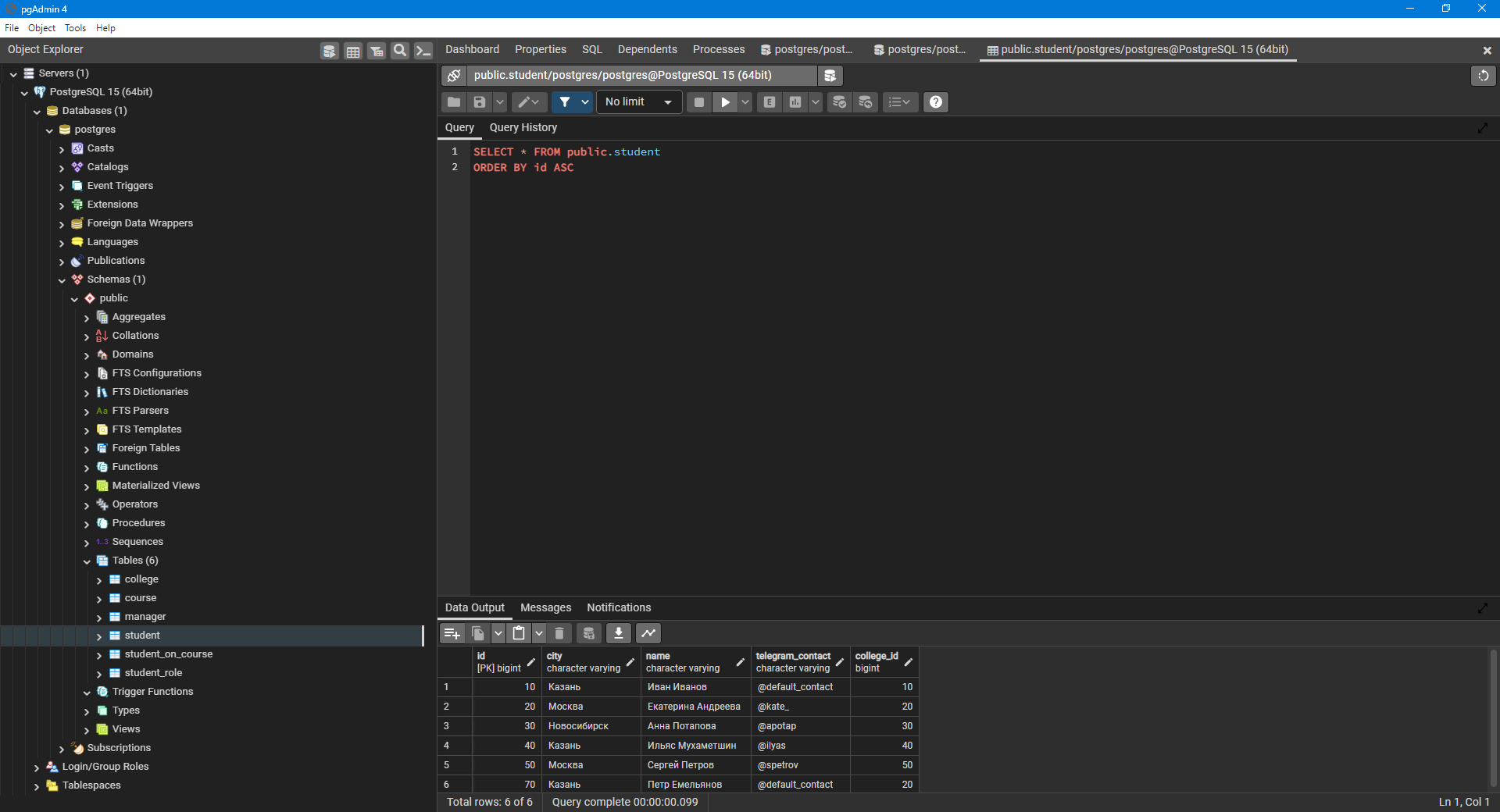
SET telegram\_contact = NULL

WHERE id = 10;

INSERT INTO student (id, city, name, telegram\_contact,

college\_id)

VALUES (70, 'Казань', 'Петр Емельянов', NULL, 20);



c.

CREATE TABLE college\_audit

(

audit\_id serial primary key,

"DS DATE" timestamp,

CHANGE\_TYPE character varying,

ROW\_ID INTEGER,

NAME character varying,

SIZE bigint

)

create or replace function fnc\_trg\_audit\_college()

returns trigger as $$

begin

if (TG\_OP = 'INSERT') then

insert into college\_audit ("DS DATE", CHANGE\_TYPE, ROW\_ID, NAME, SIZE)

values (current\_timestamp, 'insert', new.id, new.name, new.size);

elsif (TG\_OP = 'DELETE') then

insert into college\_audit ("DS DATE", CHANGE\_TYPE, ROW\_ID, NAME, SIZE)

values (current\_timestamp, 'delete', old.id, old.name, old.size);

elsif (TG\_OP = 'UPDATE') then

insert into college\_audit ("DS DATE", CHANGE\_TYPE, ROW\_ID, NAME, SIZE)

values (current\_timestamp, 'update', new.id, new.name, new.size);

insert into college\_audit ("DS DATE", CHANGE\_TYPE, ROW\_ID, NAME, SIZE)

values (current\_timestamp, 'old\_value\_update', old.id, old.name, old.size);

end if;

return new;

end; $$

language plpgsql;

create trigger trg\_audit\_college

after insert or update or delete on college

for EACH ROW execute function fnc\_trg\_audit\_college();

INSERT INTO college(id, name,size)

VALUES (60,'КАИ',12000);

UPDATE college

SET size = 12500

WHERE id = 60;

UPDATE college

SET size = 12800

WHERE id = 60;

DELETE FROM college

WHERE id = 60;

