**1. Функции и процедуры для выполнения CRUD-операций (общая база данных).**

1.1 create or replace procedure pr\_add\_ сustomers(new\_id integer,name varchar(50))

language plpgsql as $$

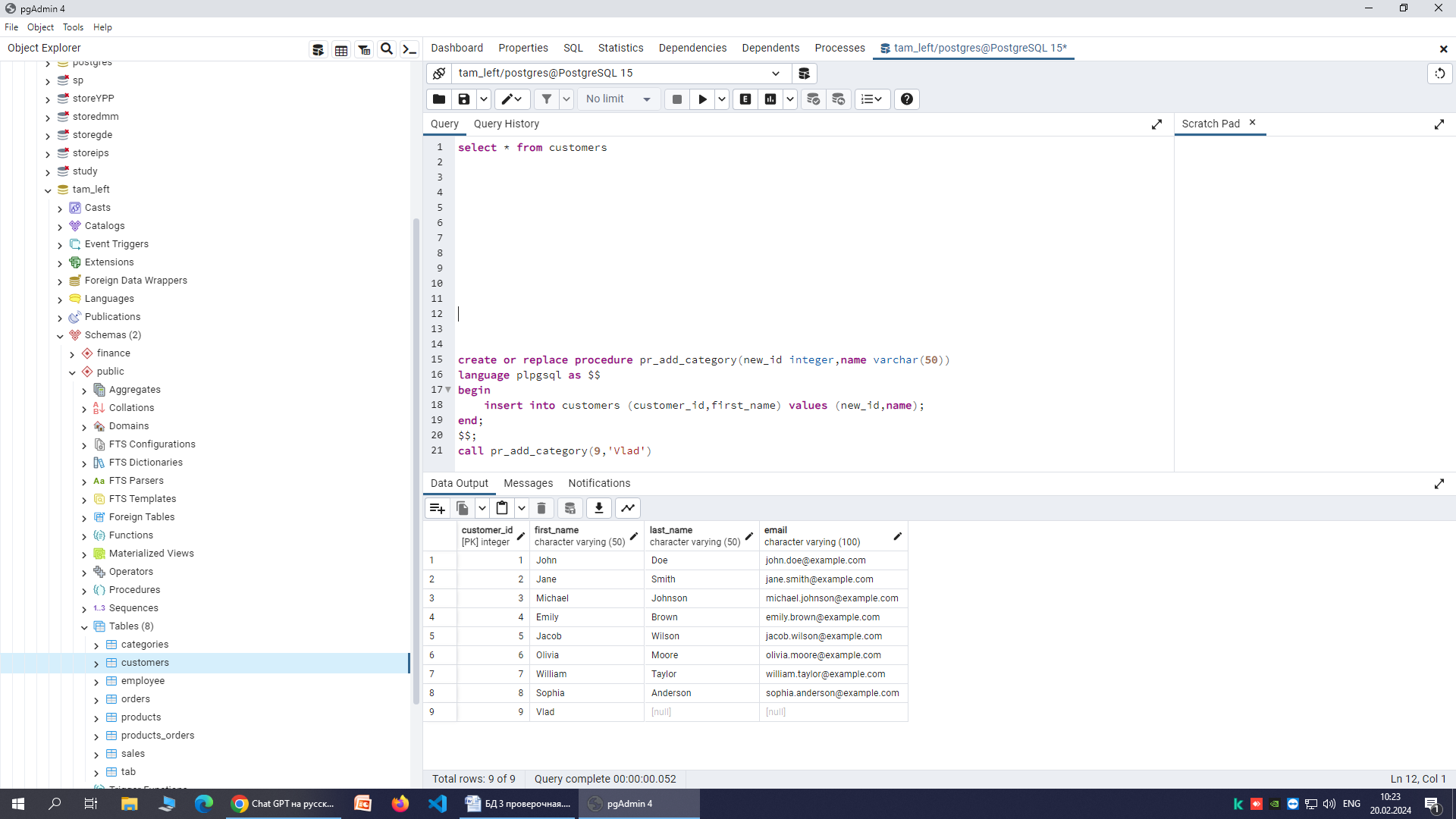
begin

insert into сustomers (customer\_id,first\_name) values (new\_id,name);

end;

$$;

call pr\_add\_category(9,'Vlad')



create or replace procedure pr\_update\_сustomers(cur\_id integer, name\_new varchar(50))

language plpgsql as $$

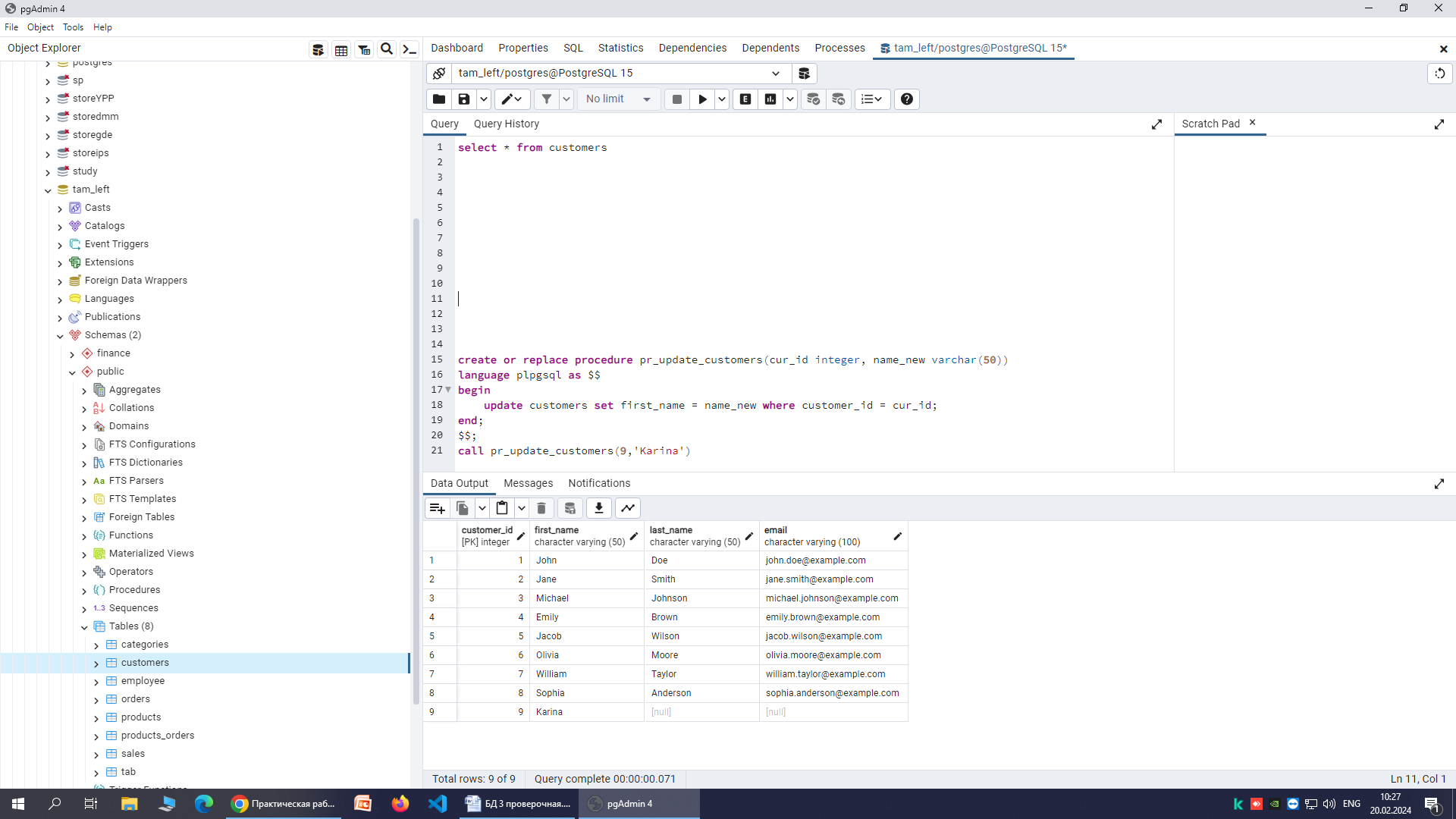
begin

update сustomers set first\_name = name\_new where customer\_id = cur\_id;

end;

$$;

call pr\_update\_сustomers(9,'Karina')



create or replace procedure pr\_delete\_сustomers(cur\_id integer)

language plpgsql as $$

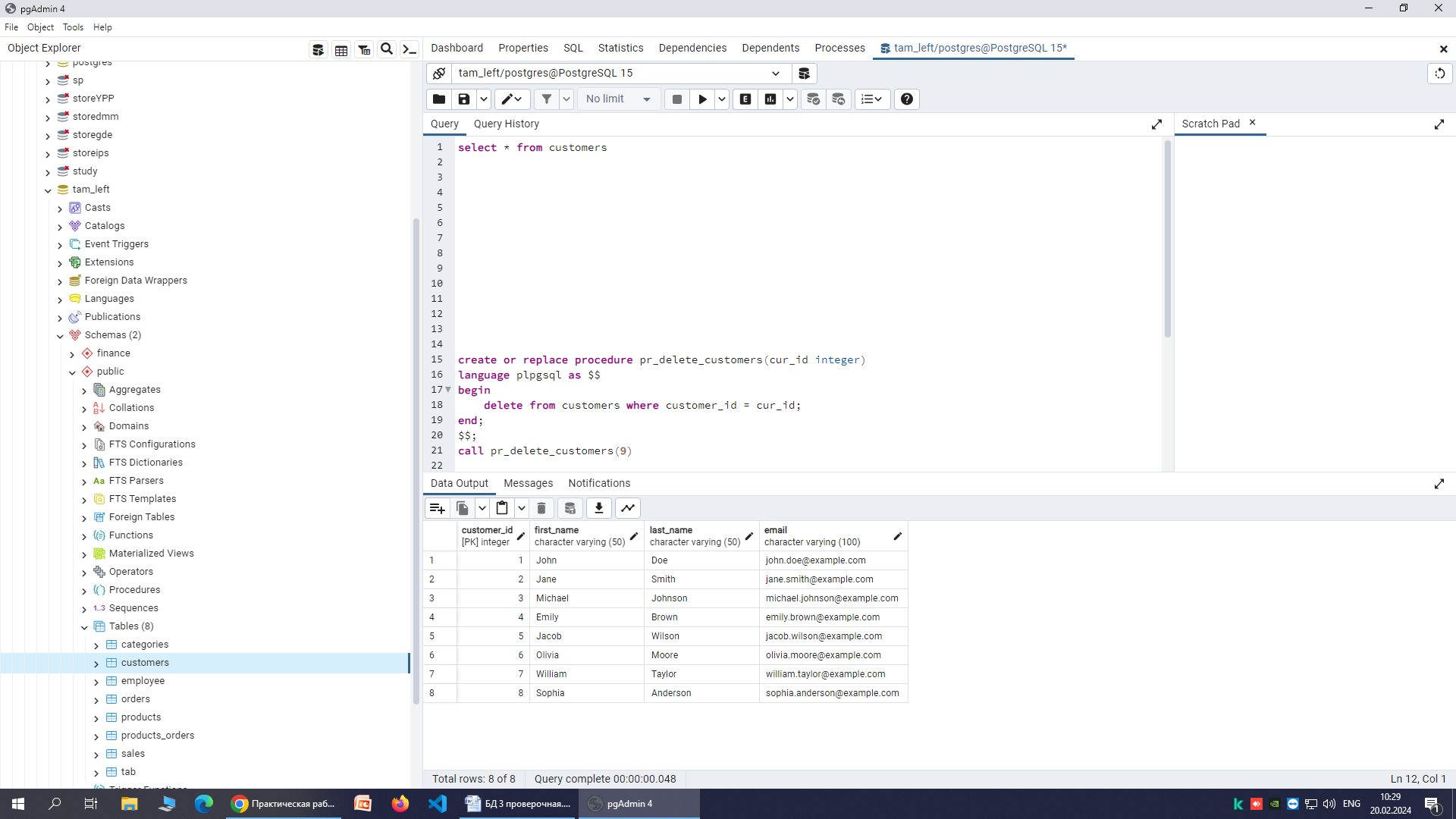
begin

delete from сustomers where customer\_id = cur\_id;

end;

$$;

call pr\_delete\_сustomers(9)



1.2 create or replace function f\_add\_products(id integer,name varchar(50))

returns integer as $$

begin

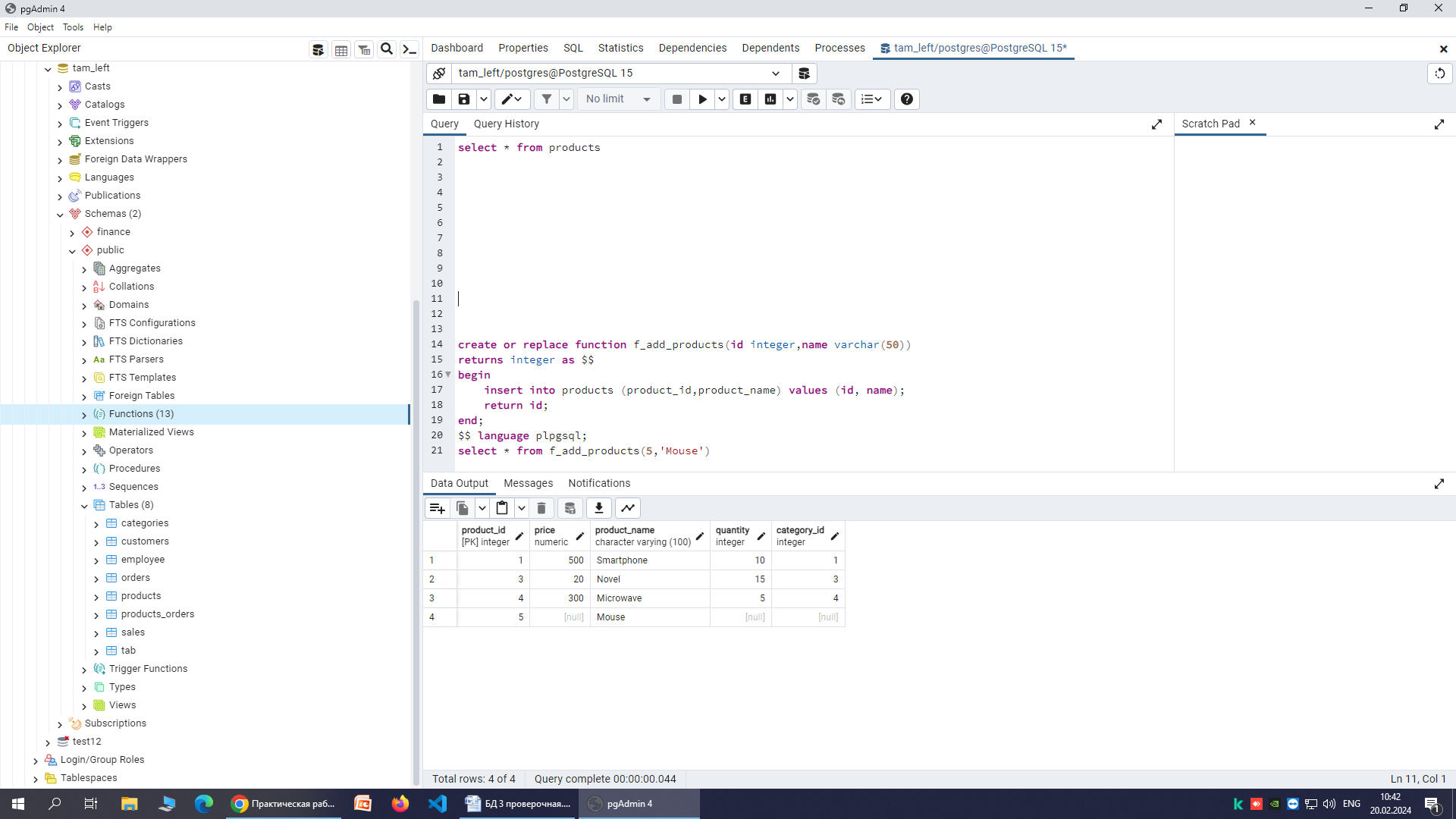
insert into products (product\_id,product\_name) values (id, name);

return id;

end;

$$ language plpgsql;

select \* from f\_add\_products(5,'Mouse')



create or replace function f\_update\_products(cur\_id integer, name\_new varchar(50))

returns integer as $$

begin

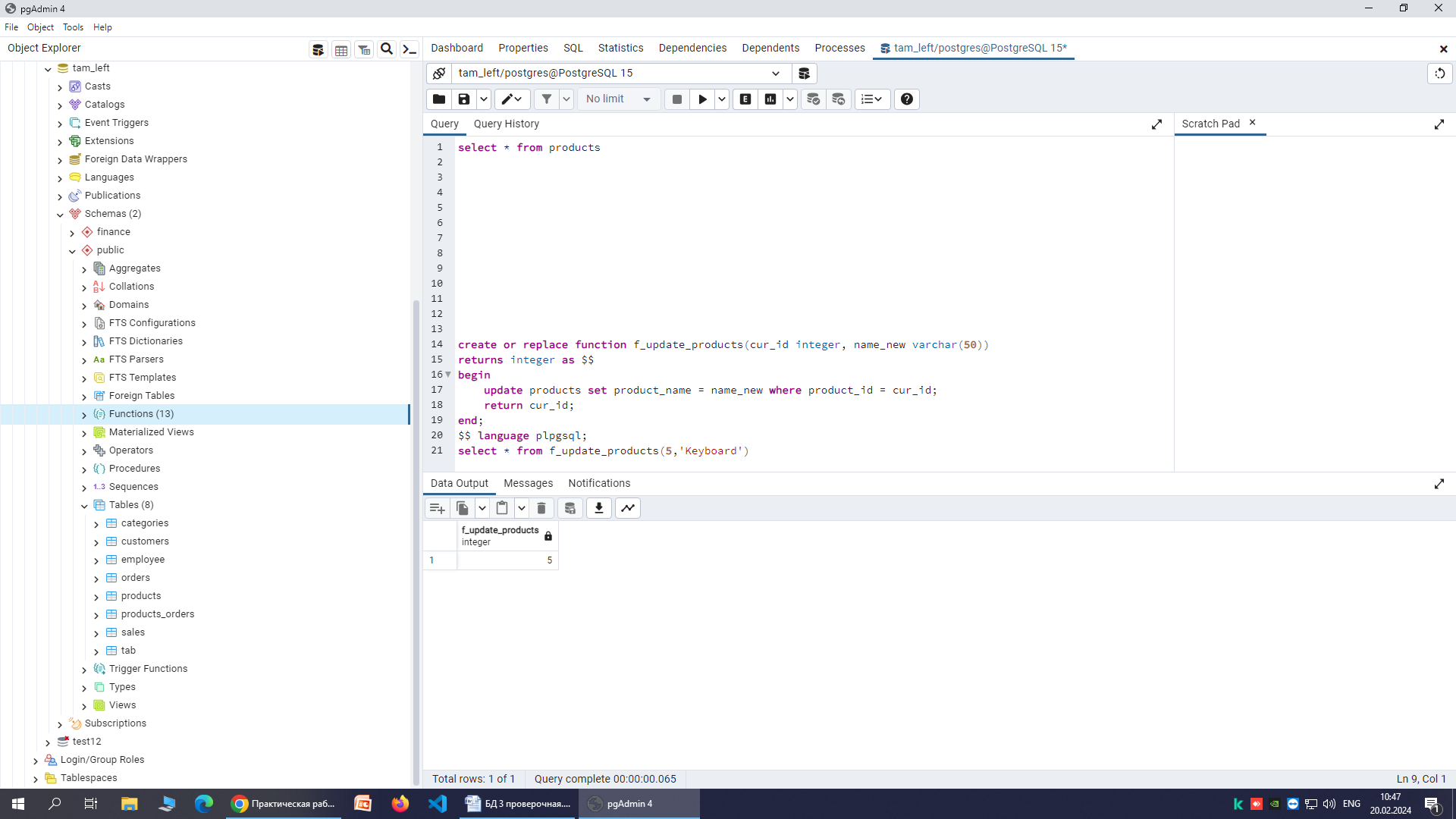
update products set product\_name = name\_new where product\_id = cur\_id;

return product\_id;

end;

$$ language plpgsql;

select \* from f\_update\_products(5,'Keyboard')



create or replace function f\_delete\_products(cur\_id integer)

returns integer as $$

begin

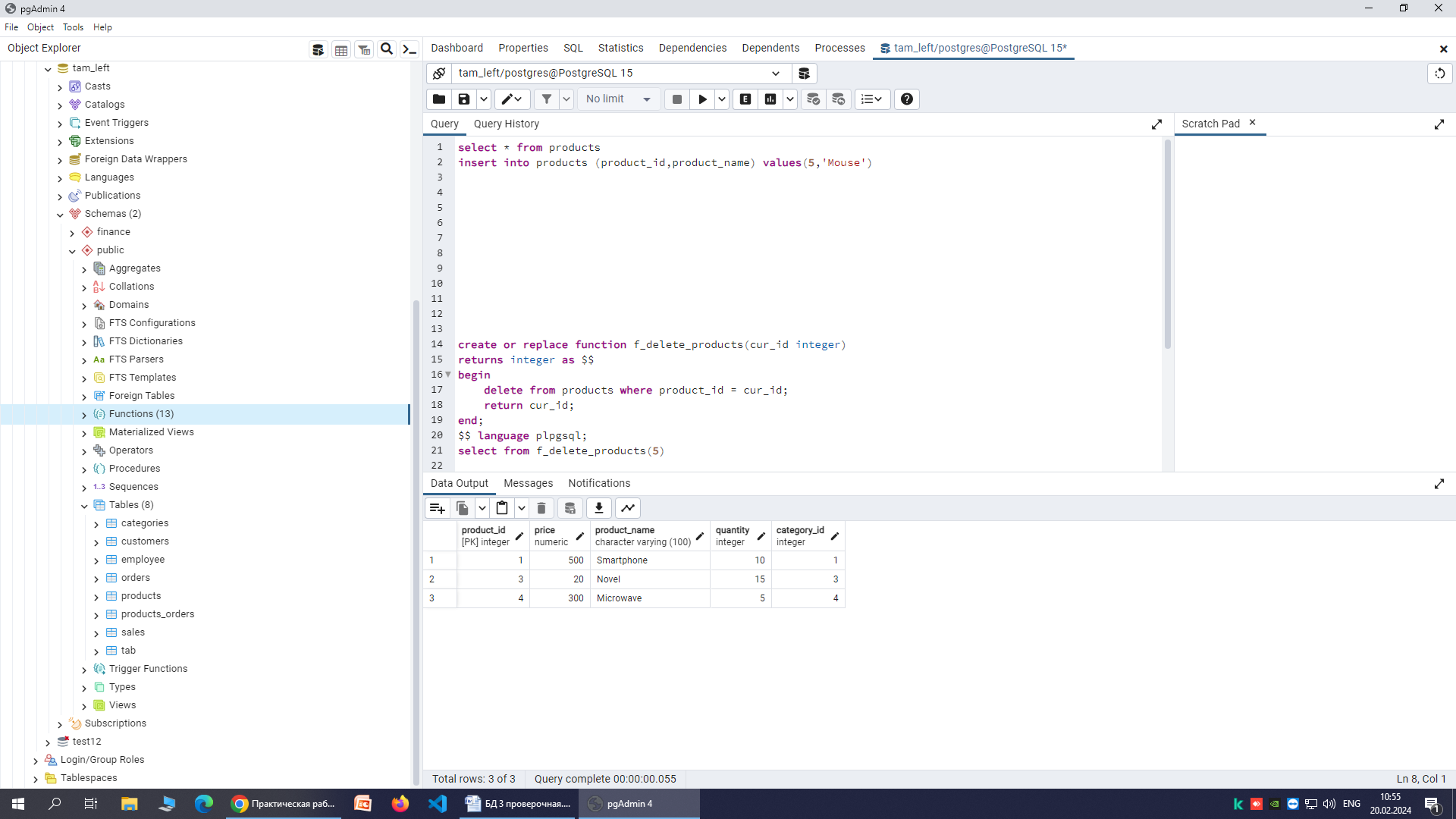
delete from products where product\_id = cur\_id;

return cur\_id;

end;

$$ language plpgsql;

select from f\_delete\_products(5)



create or replace function f\_select\_category(cur\_id integer)

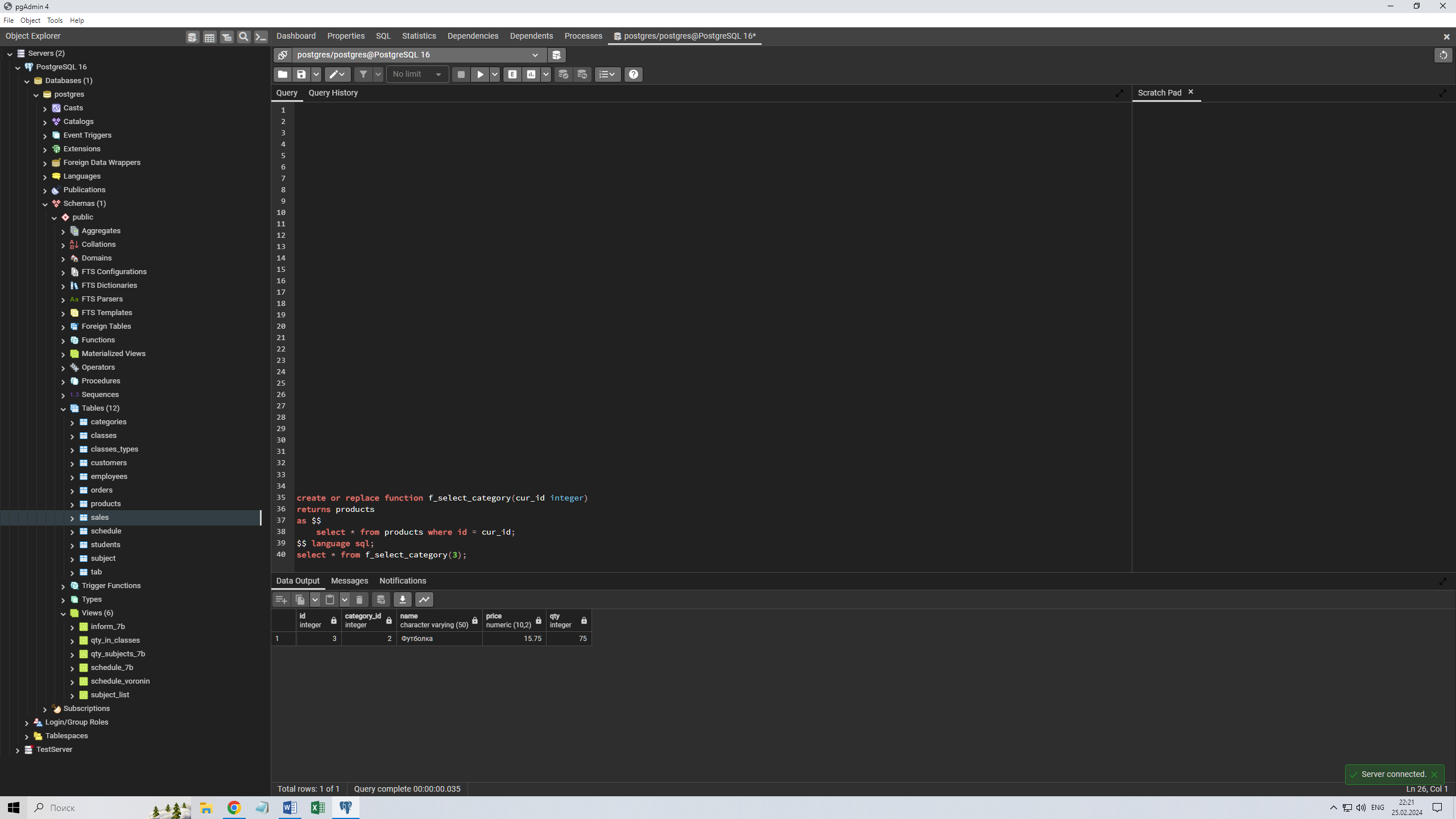
returns products

as $$

select \* from products where id = cur\_id;

$$ language sql;

select \* from f\_select\_category(3);



**2. Одиночные курсоры**

2.1 create or replace function get\_cursor (curs1 refcursor)

returns refcursor as $$

begin

open $1 for select \* from categories;

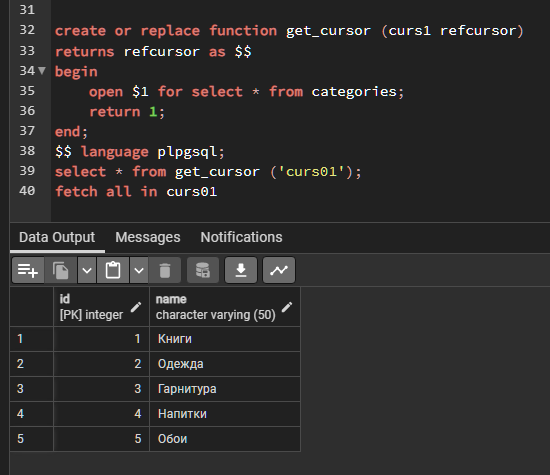
return $1;

end;

$$ language plpgsql;

select \* from get\_cursor ('curs01');

fetch all in curs01



2.2 create or replace function difcursor()

returns varchar(50) as $$

declare

curs1 cursor for select name from products where category\_id=3;

nm varchar(50);

begin

open curs1;

RAISE NOTICE 'Товары с категорией 3:';

loop

fetch curs1 into nm;

EXIT WHEN NOT FOUND;

RAISE NOTICE '%', nm;

return next nm;

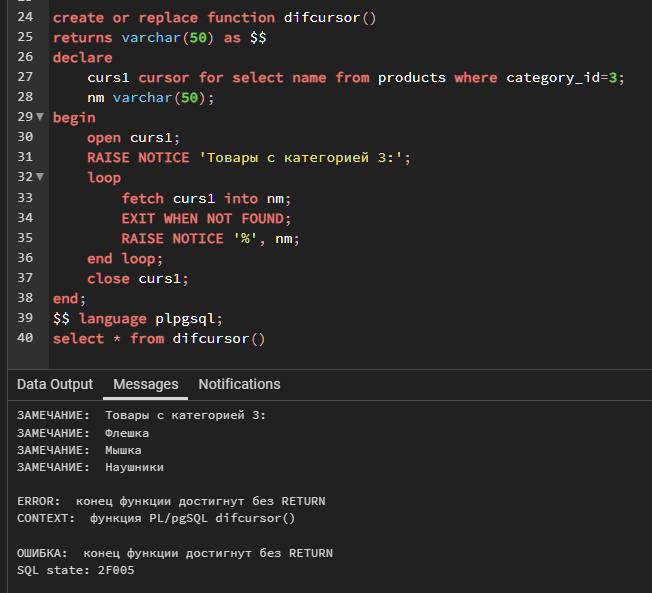
end loop;

close curs1;

end;

$$ language plpgsql;

select \* from difcursor()



2.3 create or replace function get\_cursor3()

returns numeric(15,2) as $$

declare

curs1 cursor for select dis\_amount from orders where datee='2023-10-01';

pay numeric(15,2):=0;

summ numeric(15,2):=0;

begin

open curs1;

loop

fetch curs1 into pay;

EXIT WHEN NOT FOUND;

summ:=summ+pay;

end loop;

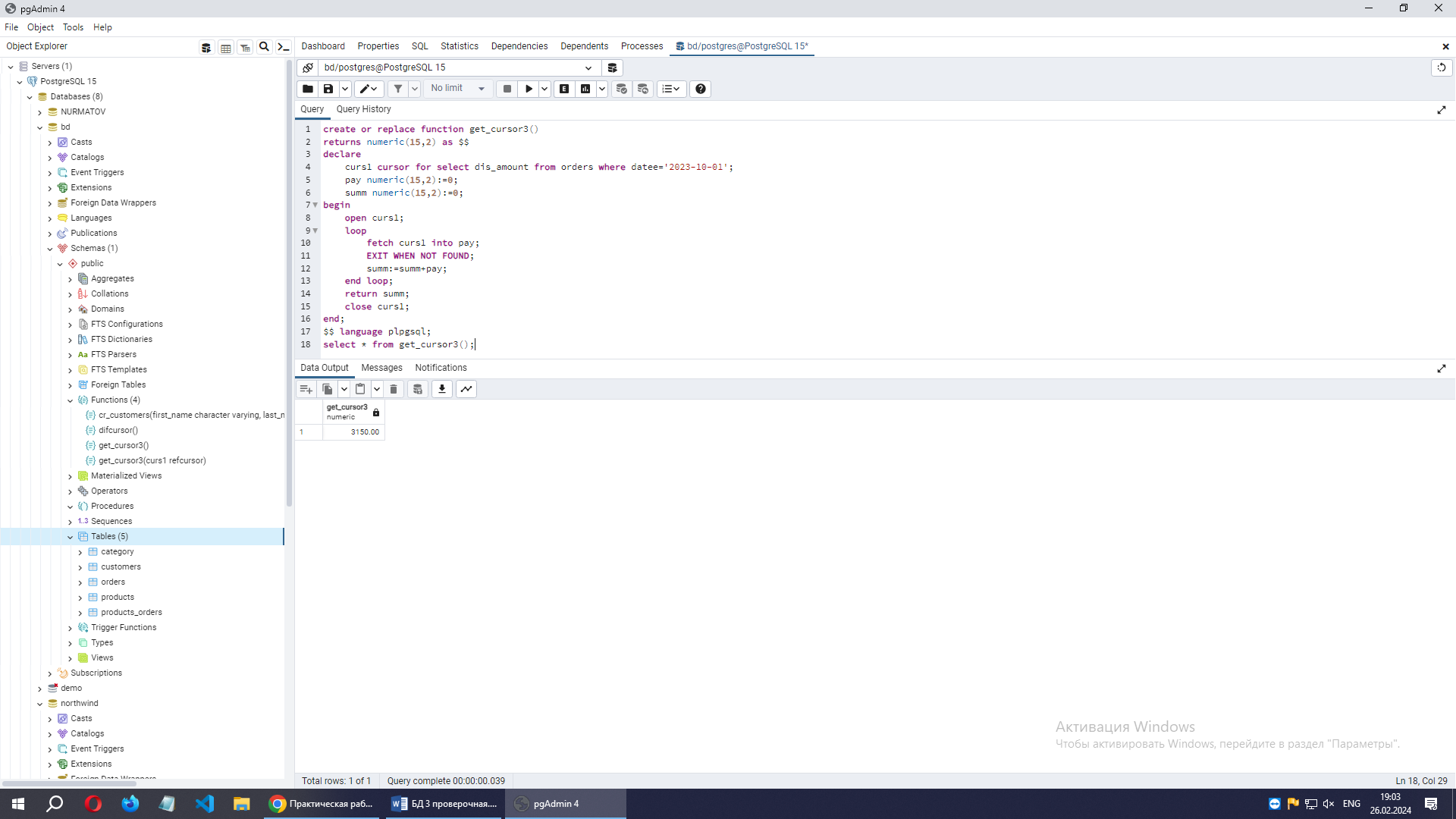
return summ;

close curs1;

end;

$$ language plpgsql;

select \* from get\_cursor3();



2.4

create or replace function get\_cursor(refcursor,refcursor)

returns SETOF refcursor as $$

begin

open $1 for SELECT \* FROM clients WHERE phone\_number LIKE '\_982%';

RETURN NEXT $1;

open $2 for SELECT \* FROM clients WHERE phone\_number LIKE '\_982%';

RETURN NEXT $2;

return;

end;

$$ language plpgsql;

begin;

select \* from get\_cursor('a','b');

fetch all in curs01

commit;

do

declare

tmpa record;

tmpb record;

begin

fetch from a into tmpa

while found loop

raise notice '%', tmpa

end;

open $1 for SELECT \* FROM clients WHERE phone\_number LIKE '\_982%';

while $1 IS NOT NULL LOOP

FETCH $1 INTO id, phone, fname, mail;

RETURN NEXT $1;

END LOOP;

RETURN;

CLOSE $1;

CREATE TABLE сustomers (

customer\_ID INT PRIMARY KEY,

first\_Name VARCHAR(50),

last\_Name VARCHAR(50),

email VARCHAR(100)

)

INSERT INTO сustomers VALUES

(1, 'John', 'Doe', 'john.doe@example.com'),

(2, 'Jane', 'Smith', 'jane.smith@example.com'),

(3, 'Michael', 'Johnson', 'michael.johnson@example.com'),

(4, 'Emily', 'Brown', 'emily.brown@example.com'),

(5, 'Jacob', 'Wilson', 'jacob.wilson@example.com'),

(6, 'Olivia', 'Moore', 'olivia.moore@example.com'),

(7, 'William', 'Taylor', 'william.taylor@example.com'),

(8, 'Sophia', 'Anderson', 'sophia.anderson@example.com');