**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базы данных»

Отчет по лабораторной работе №5

«Использование триггеров и хранимых

процедур в PostgreSQL.»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-42Б |  | преподаватель каф. ИУ5 |
| Афонин И.И. |  | Силантьева Е. Ю. |

2025 г.

## Цель лабораторной работы

Изучить хранимые процедуры, функции и триггеры в базах данных, приобрести практические навыки создания хранимых процедур и триггеров в PostgreSQL.

## Полученное задание

1. Изучить теоретические сведения лабораторной работы.

2. Создать следующие хранимые процедуры и функции (в своей базе):

a. Процедуру для изменения данных таблицы.

b. Процедуру для вставки данных в таблицу.

c. Примеры арифметических функций из теоретической части

лабораторной работы.

d. Функцию для поиска информации по названию компании.

e. Функцию для поиска товаров по диапазону цен.

f. Функцию для поиска заказов по дате заказа и/или диапазону дат

заказа, доставки, в зависимости от введенных параметров.

g. Функции по заданию варианта.

3. Создать триггер INSERT.

4. Создать триггер DELETE.

5. Создать триггер UPDATE.

6. Создать триггер, который при удалении записи из таблицы Products сначала

удаляет все связанные с ней записи из таблицы Items, а затем удаляет саму

запись из таблицы Products.

7. Создать триггер, с использованием временной таблицы NEW.

8. Создать триггер DDL, который предотвратит удаление или изменение таблиц

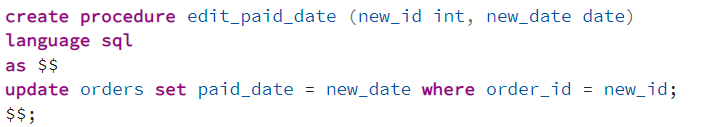
в базе данных.

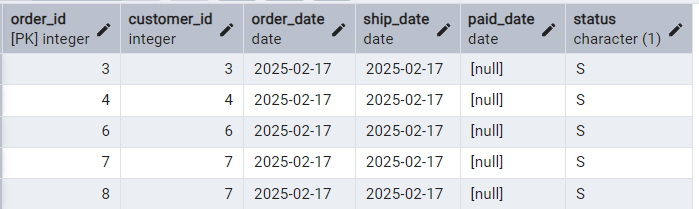
9. Подготовиться к защите лабораторной работы (уметь написать собственную

процедуру/триггер).

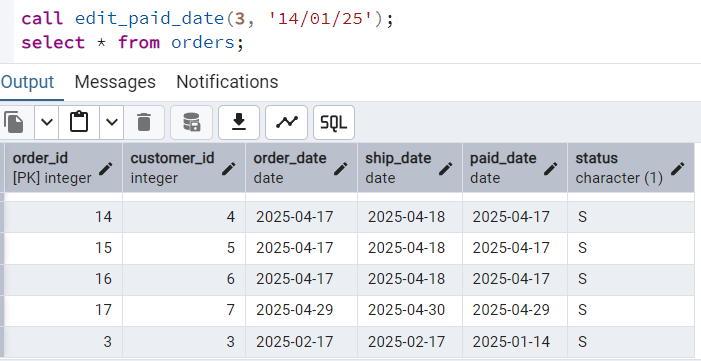
## Выполнение лабораторной работы

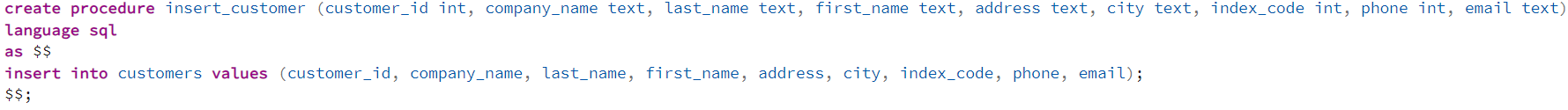
2.а) Реализуем процедуру обновления даты платежа.

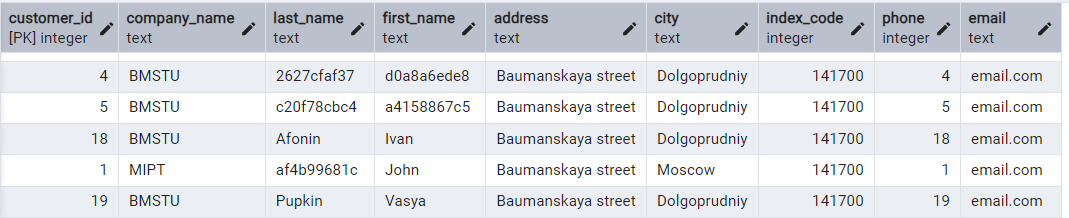




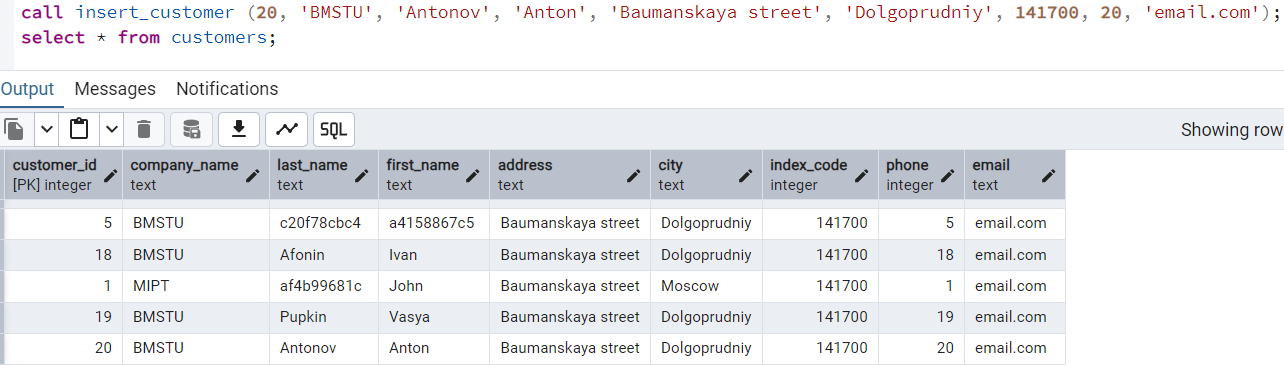
Выполним процедуру:

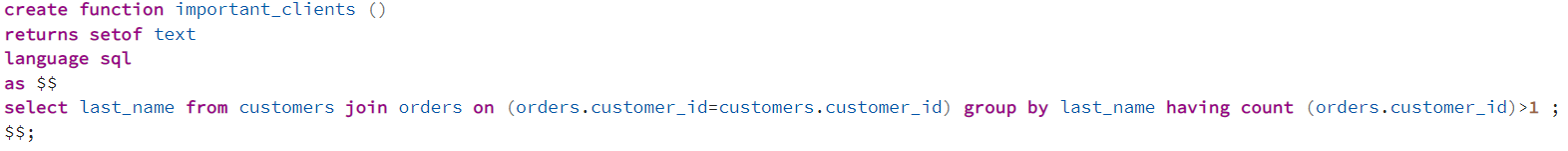


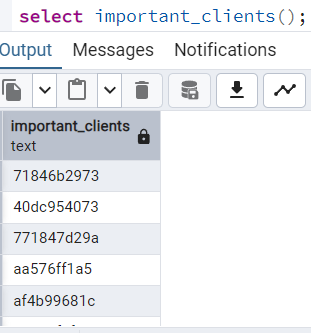
2.b) Создадим процедуру для вставки информации о новом покупателе. 



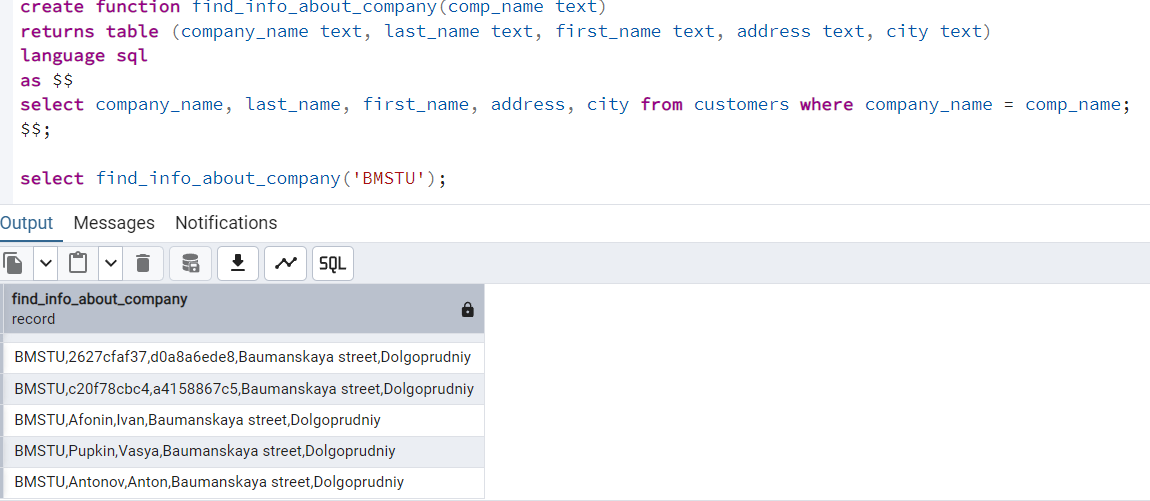
Выполним процедуру:



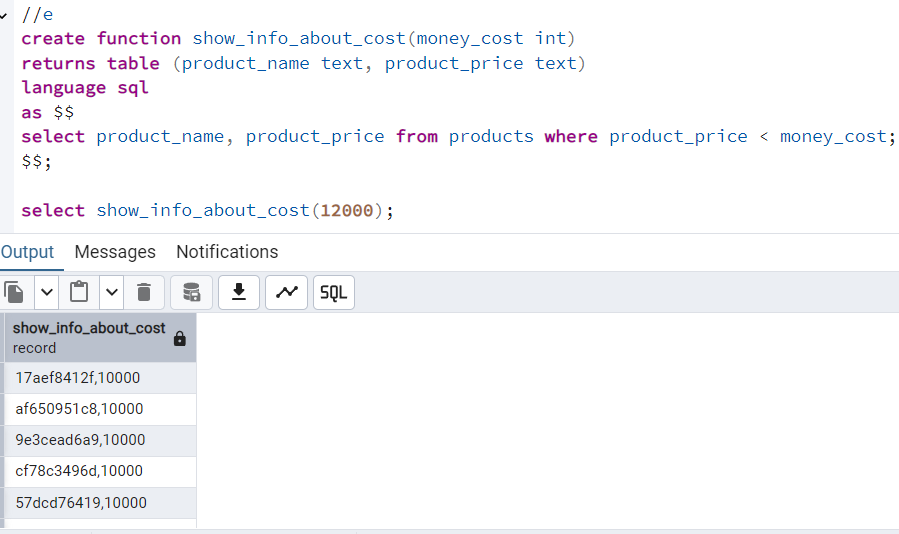
2.с) Создадим функцию c арифметической функцией COUNT.  




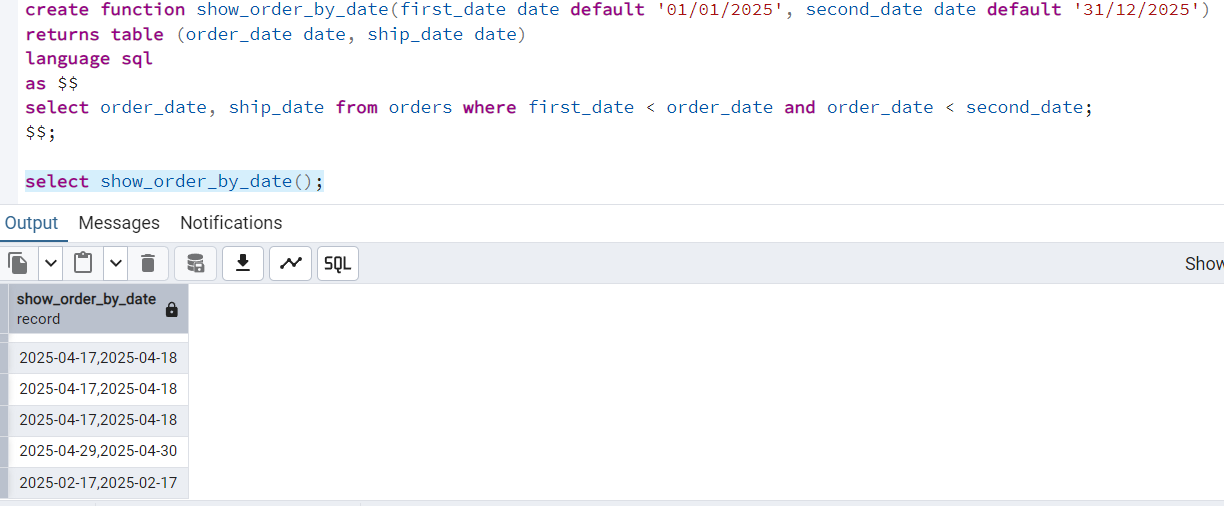
2.d) Создадим функцию для поиска информации по названию компании.



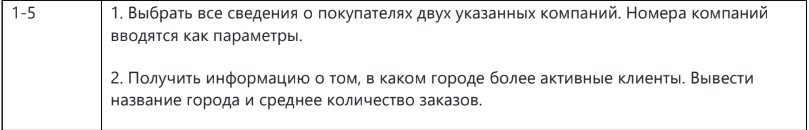
2.e) Создадим функцию для поиска товаров по диапазону цен.

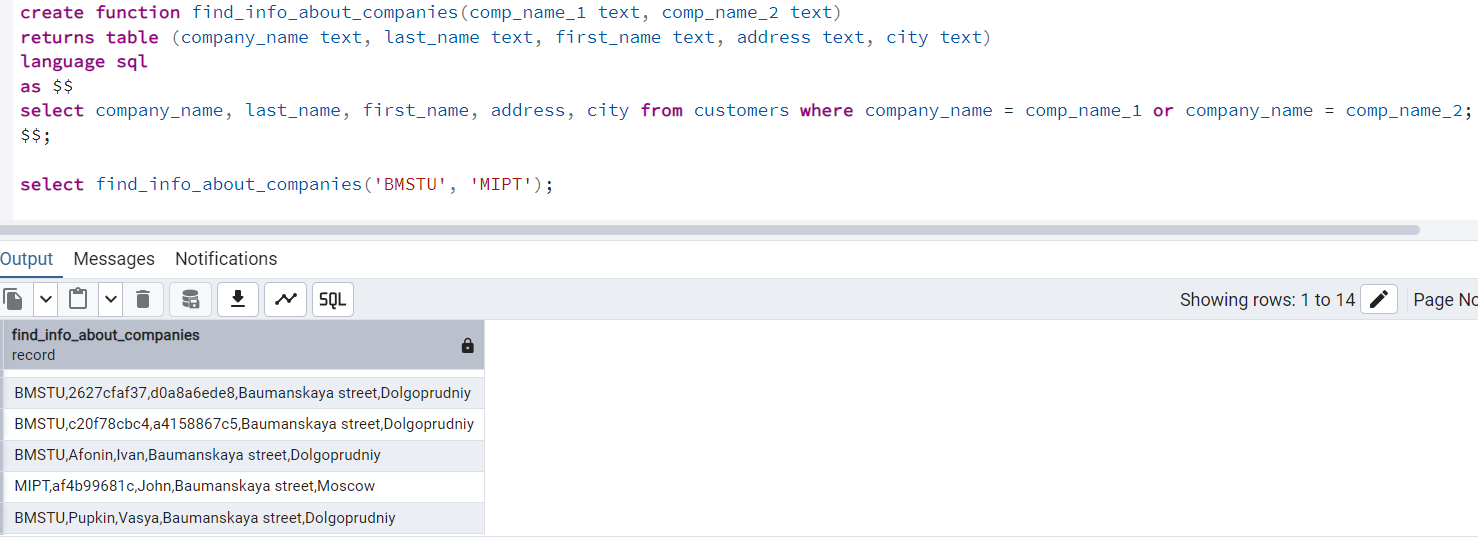


2.f) Создадим функцию, для поиска заказов по дате заказа и/или диапазону дат заказа, доставки, в зависимости от введенных параметров.



2.g) Создадим функции по заданию варианта.





create function find\_info\_about\_companies(comp\_name\_1 text, comp\_name\_2 text)

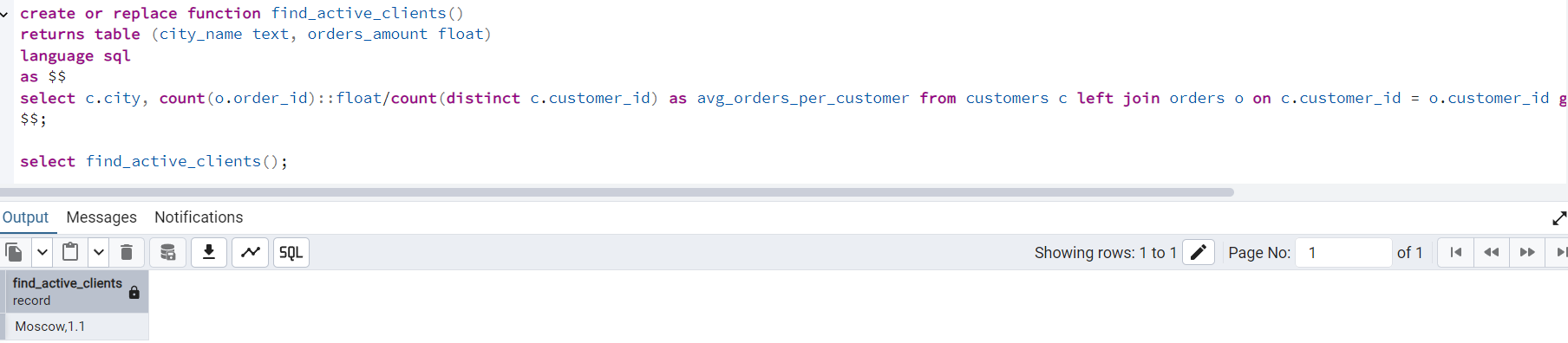
returns table (company\_name text, last\_name text, first\_name text, address text, city text)

language sql

as $$

select company\_name, last\_name, first\_name, address, city from customers where company\_name = comp\_name\_1 or company\_name = comp\_name\_2;

$$;



create or replace function find\_active\_clients()

returns table (city\_name text, orders\_amount float)

language sql

as $$

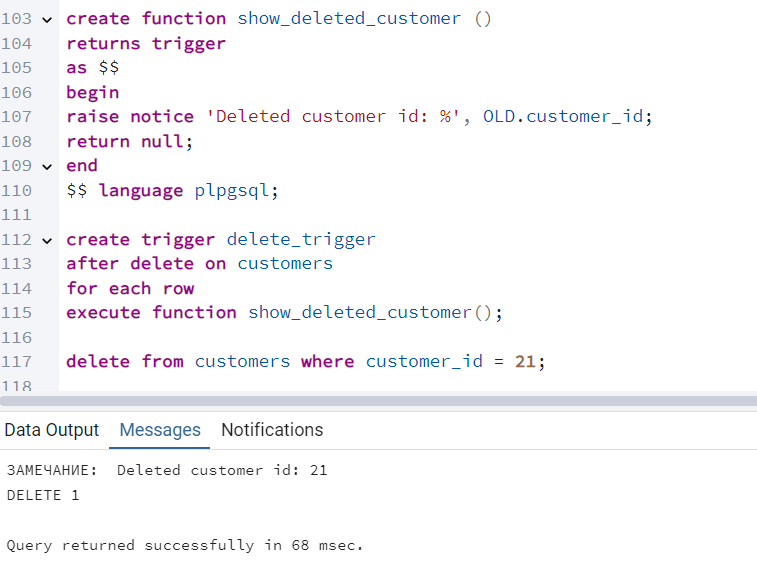
select c.city, count(o.order\_id)::float/count(distinct c.customer\_id) as avg\_orders\_per\_customer from customers c left join orders o on c.customer\_id = o.customer\_id group by c.city order by c.city desc limit 1;

$$;

3) Данный INSERT триггер автоматически при создании нового покупателя выводит в консоль его id.



4) Данный DELETE триггер автоматически при удалении существующего покупателя выводит в консоль его id.



5) Данный UPDATE триггер автоматически выводит в консоль при редактировании имени пользователя его новое имя.



6) Создадим триггер, который при удалении записи из таблицы products сначала удаляет все связанные с ней записи из таблицы items, а затем удаляет саму запись из таблицы products.

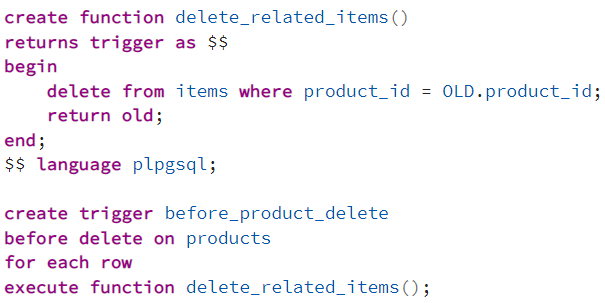


Таблица products:

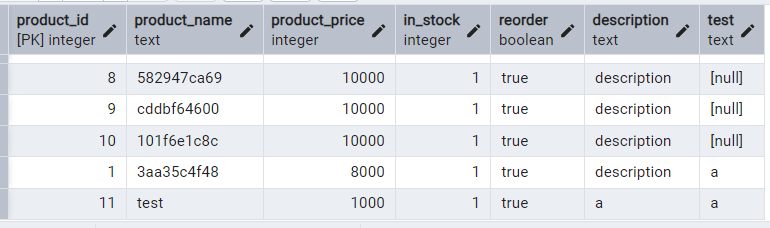
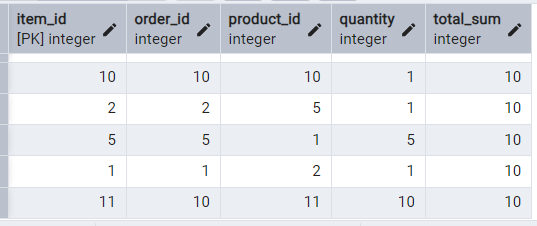


Таблица items:



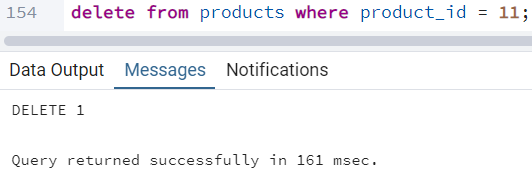


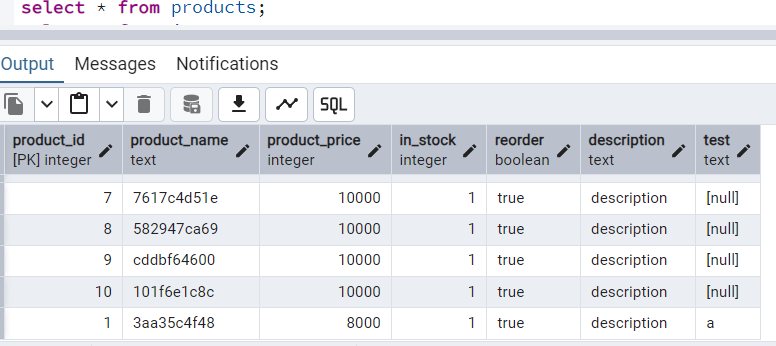
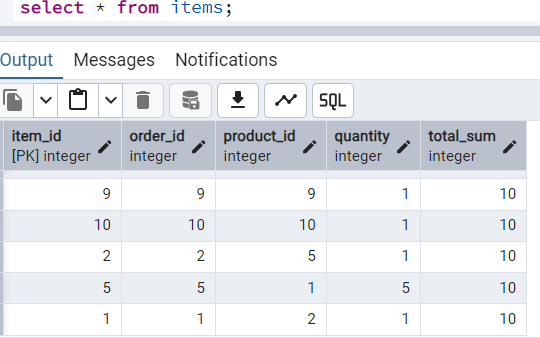
Таблица products:

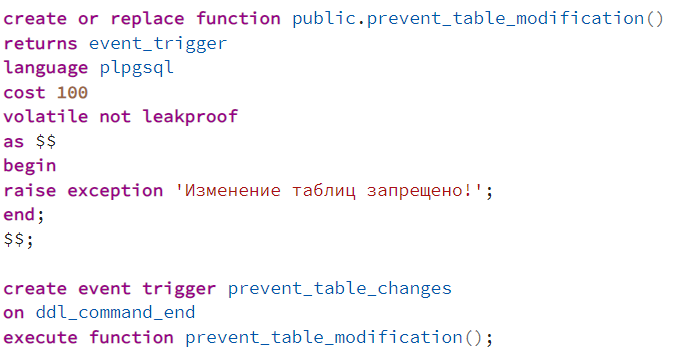
Таблица items:



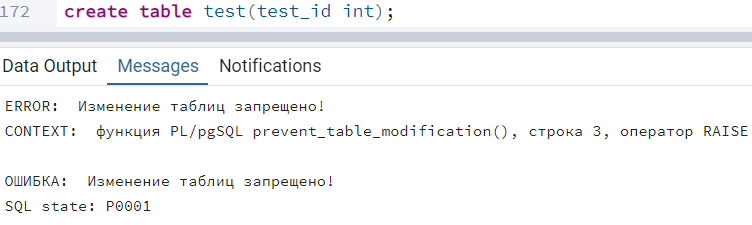
7) Таблица NEW используется в продемонстрированных триггерах.

8) Запретим пользователю изменять таблицы.

Создадим DDL триггер.



Попробуем создать таблицу test.



## Выводы по лабораторной работе

Были изучены хранимые процедуры, функции и триггеры в базах данных, приобретены практические навыки создания хранимых процедур и триггеров в PostgreSQL.