ubuntusql@ubuntusql-VirtualBox:~$ docker run -d -ti --name pg -p 5432:5432 -v /home/ubuntusql/sql/pg2:/var/lib/postgresql/data -e POSTGRES\_PASSWORD=mysecretpassword postgres:13

**Задача 1**

Используя docker поднимите инстанс PostgreSQL (версию 13). Данные БД сохраните в volume.

Подключитесь к БД PostgreSQL используя psql.

Воспользуйтесь командой \? для вывода подсказки по имеющимся в psql управляющим командам.

**Найдите и приведите** управляющие команды для:

* вывода списка БД

\l

* подключения к БД

\c[onnect] {[DBNAME|- USER|- HOST|- PORT|-] | conninfo}

* вывода списка таблиц

**\dt**

* вывода описания содержимого таблиц

\d[S+] NAME

* выхода из psql

\ q

## Задача 2

Используя psql создайте БД test\_database.

Изучите [бэкап БД](https://github.com/netology-code/virt-homeworks/tree/master/06-db-04-postgresql/test_data).

Восстановите бэкап БД в test\_database.

Перейдите в управляющую консоль psql внутри контейнера.

Подключитесь к восстановленной БД и проведите операцию ANALYZE для сбора статистики по таблице.

Используя таблицу [pg\_stats](https://postgrespro.ru/docs/postgresql/12/view-pg-stats), найдите столбец таблицы orders с наибольшим средним значением размера элементов в байтах.

**Приведите в ответе** команду, которую вы использовали для вычисления и полученный результат.

Подключитесь к восстановленной БД и проведите операцию ANALYZE для сбора статистики по таблице.

create DATABASE test\_database; # создать БД

analyze verbose orders; # проанализировать и вывести verbose(наглядно) таблицу orders

Используя таблицу [pg\_stats](https://postgrespro.ru/docs/postgresql/12/view-pg-stats), найдите столбец таблицы orders с наибольшим средним значением размера элементов в байтах.

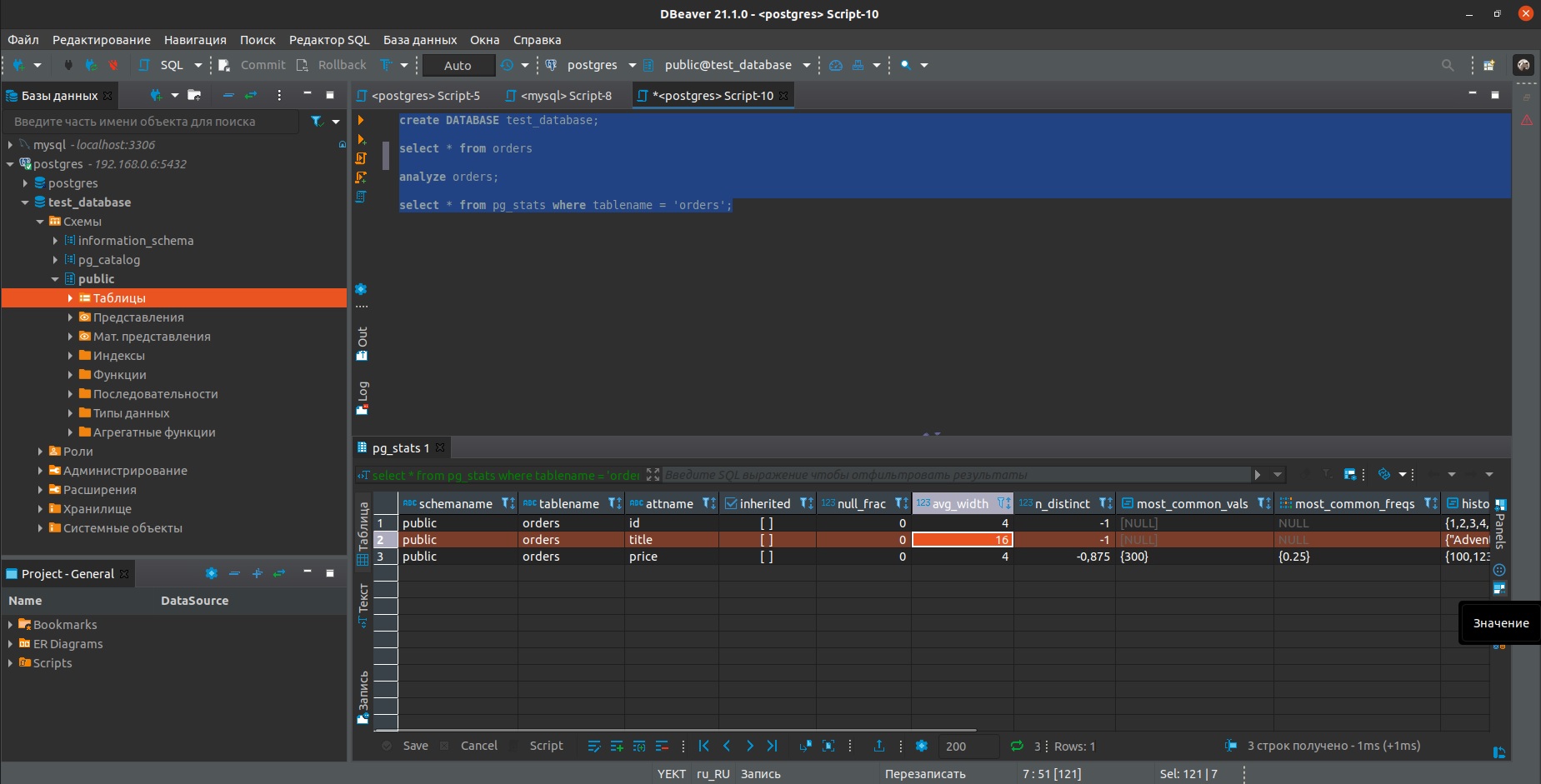
select \* from pg\_stats where tablename = 'orders'; #

```

*"orders": scanned 1 of 1 pages, containing 8 live rows and 8 dead rows; 8 rows in sample, 8 estimated total rows*

```

см. скрин “avg-width Домашнее задание к занятию 6.4. PostgreSQL.jpg”



**Задача 3**

Архитектор и администратор БД выяснили, что ваша таблица orders разрослась до невиданных размеров и поиск по ней занимает долгое время. Вам, как успешному выпускнику курсов DevOps в нетологии предложили провести разбиение таблицы на 2 (шардировать на orders\_1 - price>499 и orders\_2 - price<=499).

Предложите SQL-транзакцию для проведения данной операции.

Можно ли было изначально исключить "ручное" разбиение при проектировании таблицы orders?

CREATE TABLE public.orders (

id integer NOT NULL,

title character varying(80) NOT NULL,

price integer DEFAULT 0

) partition by range(price);

# создаем табличку public.orders с колонками id integer; title character; price integer

INSERT INTO public.orders VALUES (1,'War and peace',100);

INSERT INTO public.orders (id,title,price)

VALUES (2,'My little database',500);

INSERT INTO public.orders (id,title,price)

VALUES (3,'Adventure psql time',300);

INSERT INTO public.orders (id,title,price)

VALUES (4,'Server gravity falls',300);

INSERT INTO public.orders (id,title,price)

VALUES (5,'Log gossips',123);

INSERT INTO public.orders (id,title,price)

VALUES (6,'WAL never lies',900);

INSERT INTO public.orders (id,title,price)

VALUES (7,'Me and my bash-pet',499);

INSERT INTO public.orders (id,title,price)

VALUES (8,'Dbiezdmin',501);

# заполняем табличку  
  
select \* from orders; # вывести табличку orders

CREATE TABLE orders\_1

PARTITION OF public.orders

FOR VALUES FROM ('1') TO ('499');

CREATE TABLE orders\_2

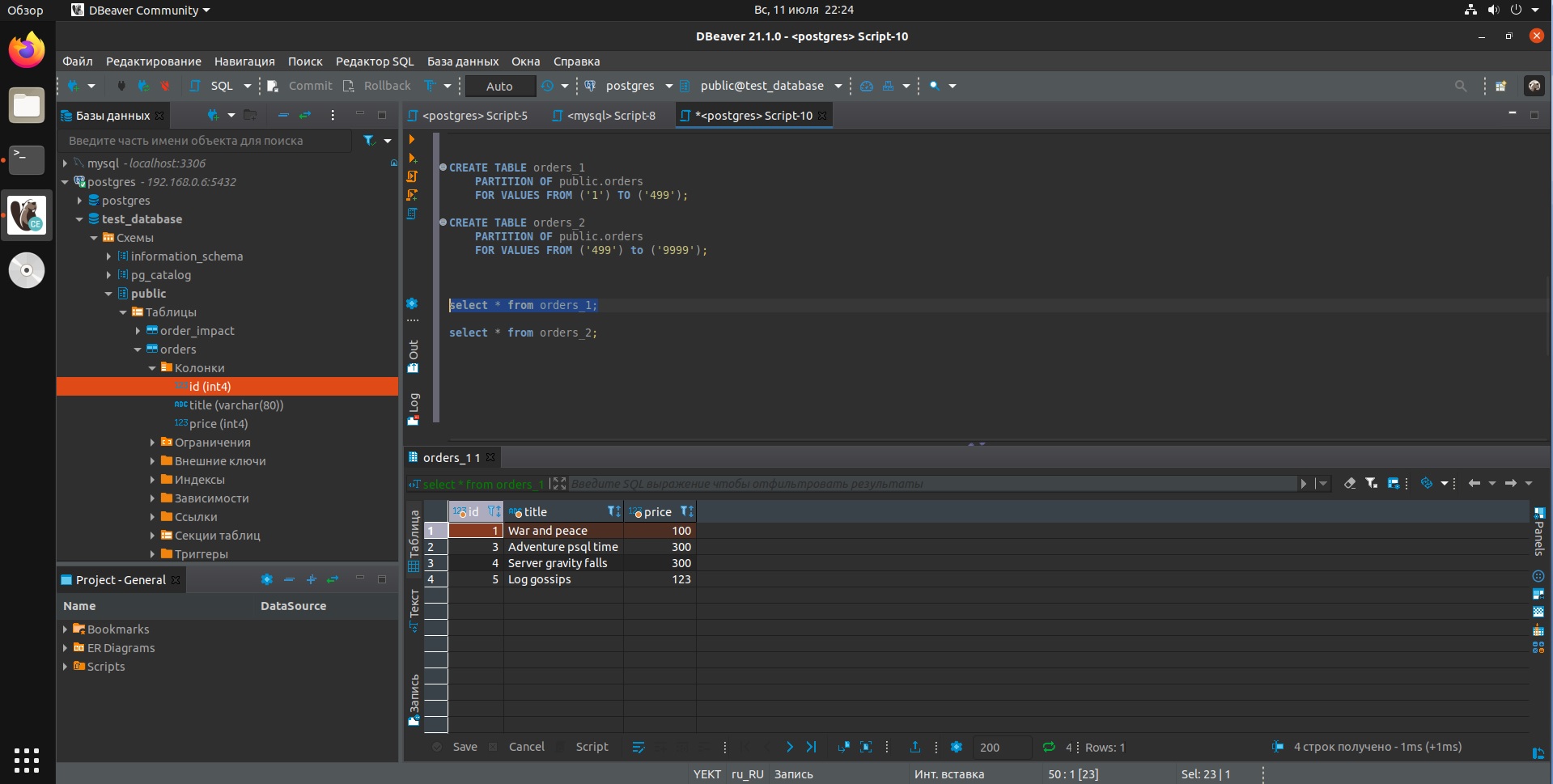
PARTITION OF public.orders

FOR VALUES FROM ('499') to ('9999');

# создали секции

select \* from orders\_1; # смотрит секции

см. скрин “select from orders\_1 Домашнее задание к занятию 6.4. PostgreSQL.jpg”



select \* from orders\_2; # смотрит секции

см. скрин “select from orders\_2 Домашнее задание к занятию 6.4. PostgreSQL.jpg”

Ручное шардирование можно было не использовать, если при создании указать PARTITION BY RANGE (price);

**Задача 4**

Используя утилиту pg\_dump создайте бекап БД test\_database.

Как бы вы доработали бэкап-файл, чтобы добавить уникальность значения столбца title для таблиц test\_database?

ubuntusql@ubuntusql-VirtualBox:~$ docker exec -ti 7faa260a3529 /bin/bash

su – postgres # заходим под пользователя postgres

postgres@7faa260a3529:~$ pg\_dump test\_database > test\_database.sql # создать dump sql

Как бы вы доработали бэкап-файл, чтобы добавить уникальность значения столбца title для таблиц test\_database?

CREATE TABLE public.orders (

id integer NOT NULL,

title character varying(80) NOT NULL,

price integer DEFAULT 0 UNIQUE

) partition by range(price);  
  
Добавил бы к столбцу UNIQUE как вариант или можно через index.