19 ДЗ - Интеграция ClickHouse с PostgreSQL

Собираем Dockerfile для clickhouse.

Dockerfile:

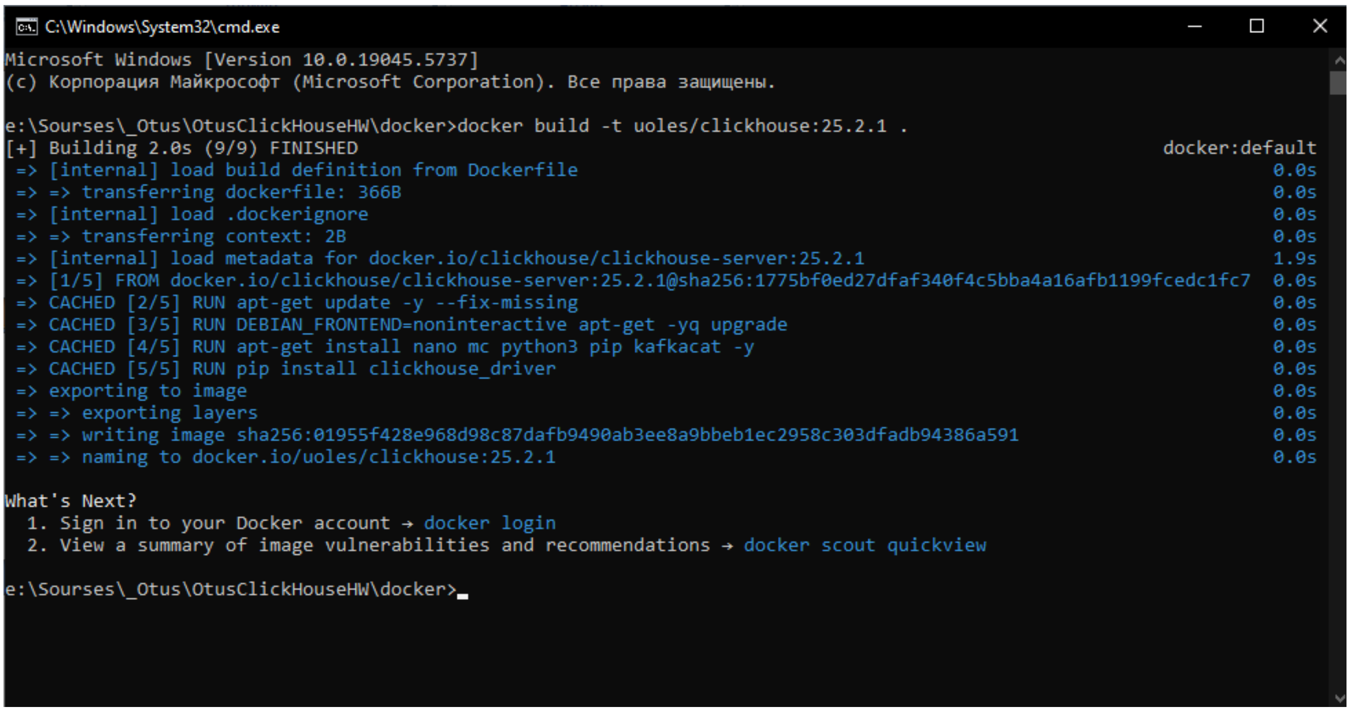
```
FROM clickhouse/clickhouse-server:25.2.1
MAINTAINER Maksim Kulikov max.uoles@rambler.ru

RUN apt-get update -y --fix-missing
RUN DEBIAN_FRONTEND=noninteractive apt-get -yq upgrade
RUN apt-get install nano mc python3 pip kafkacat –y
RUN pip install clickhouse_driver

EXPOSE 8123 9000
ENTRYPOINT ["/entrypoint.sh"]
```

Собираем образ командой:

```
docker build -t uoles/clickhouse:25.2.1 .
```



```
Microsoft Windows [Version 10.0.19045.5737]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

e:\Sourses\_Otus\OtusClickHouseHW\docker>docker build -t uoles/clickhouse:25.2.1 .
[+] Building 2.0s (9/9) FINISHED                                                                   docker:default
 => [internal] load build definition from Dockerfile                                                         0.0s
 => => transferring dockerfile: 366B                                                                         0.0s
 => [internal] load .dockerignore                                                                            0.0s
 => => transferring context: 2B                                                                              0.0s
 => [internal] load metadata for docker.io/clickhouse/clickhouse-server:25.2.1                               1.9s
 => [1/5] FROM docker.io/clickhouse/clickhouse-server:25.2.1@sha256:1775bf0ed27dfaf340f4c5bba4a16afb1199fcedc1fc7  0.0s
 => CACHED [2/5] RUN apt-get update -y --fix-missing                                                         0.0s
 => CACHED [3/5] RUN DEBIAN_FRONTEND=noninteractive apt-get -yq upgrade                                      0.0s
 => CACHED [4/5] RUN apt-get install nano mc python3 pip kafkacat -y                                         0.0s
 => CACHED [5/5] RUN pip install clickhouse_driver                                                           0.0s
 => exporting to image                                                                                       0.0s
 => => exporting layers                                                                                      0.0s
 => => writing image sha256:01955f428e968d98c87dafb9490ab3ee8a9bbeb1ec2958c303dfadb94386a591                 0.0s
 => => naming to docker.io/uoles/clickhouse:25.2.1                                                           0.0s

What's Next?
  1. Sign in to your Docker account → docker login
  2. View a summary of image vulnerabilities and recommendations → docker scout quickview

e:\Sourses\_Otus\OtusClickHouseHW\docker>_
```

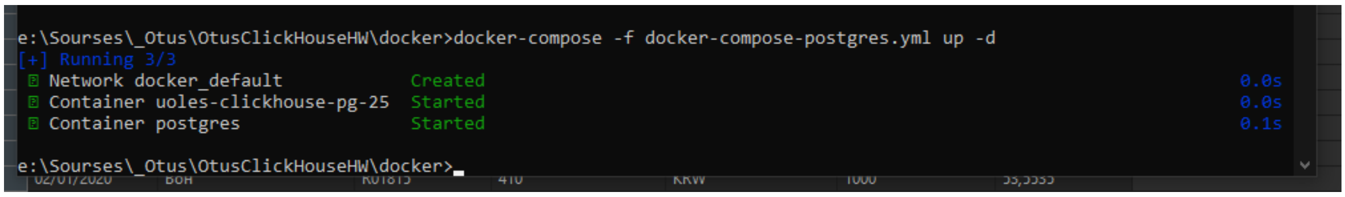
Собираем docker-compose с PostgreSQL.

Docker-compose.yml:

```
version: "3.7"
services:
  postgres:
    container_name: postgres
    image: postgres:latest
    environment:
      POSTGRES_DB: pgdb
      POSTGRES_USER: userpg
      POSTGRES_PASSWORD: passwordpg
    ports:
      - 5432:5432
    depends_on:
      - clickhouse-server
    links:
      - clickhouse-server

  clickhouse-server:
    container_name: uoles-clickhouse-pg-25
    image: uoles/clickhouse:25.2.1
    environment:
      CLICKHOUSE_DB: my_database
      CLICKHOUSE_USER: username
      CLICKHOUSE_DEFAULT_ACCESS_MANAGEMENT: 1
      CLICKHOUSE_PASSWORD: password
    ports:
      - "18123:8123"
      - "19000:9000"
    ulimits:
      nofile:
        soft: 262144
        hard: 262144
```

Собираем командой:
docker-compose -f docker-compose-postgres.yml up –d



```
e:\Sourses\_Otus\OtusClickHouseHW\docker>docker-compose -f docker-compose-postgres.yml up -d
[+] Running 3/3
 ✔ Network docker_default            Created                                                         0.0s
 ✔ Container uoles-clickhouse-pg-25  Started                                                         0.0s
 ✔ Container postgres                Started                                                         0.1s

e:\Sourses\_Otus\OtusClickHouseHW\docker>
```

Создаем и заполняем схему в PostgreSQL.

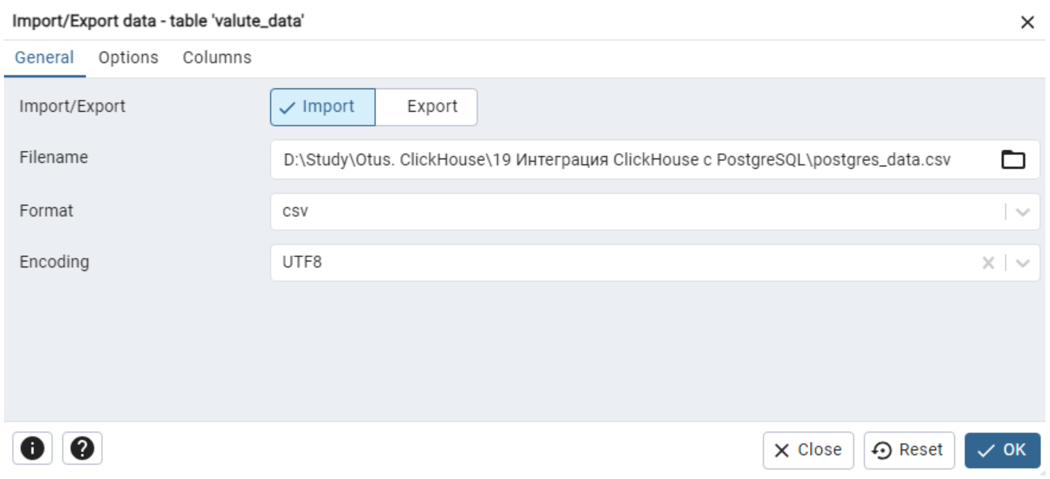
Коннектимся к базе и создаем схему, таблицу для тестовых данных.

```
CREATE SCHEMA pgclick;

CREATE TABLE IF NOT EXISTS pgclick.valute_data(
	c_date VARCHAR(36) NOT NULL,
	c_name VARCHAR(200) NOT NULL,
	c_str_id VARCHAR(20) NOT NULL,
	c_num_code VARCHAR(20) NOT NULL,
	c_char_code VARCHAR(20) NOT NULL,
	c_nominal VARCHAR(20) NOT NULL,
	c_value VARCHAR(50) NOT NULL
)

CREATE UNIQUE INDEX idx_valute_data
ON pgclick.valute_data(c_date, c_str_id);
```

Делаем импорт данных о курсах валют:



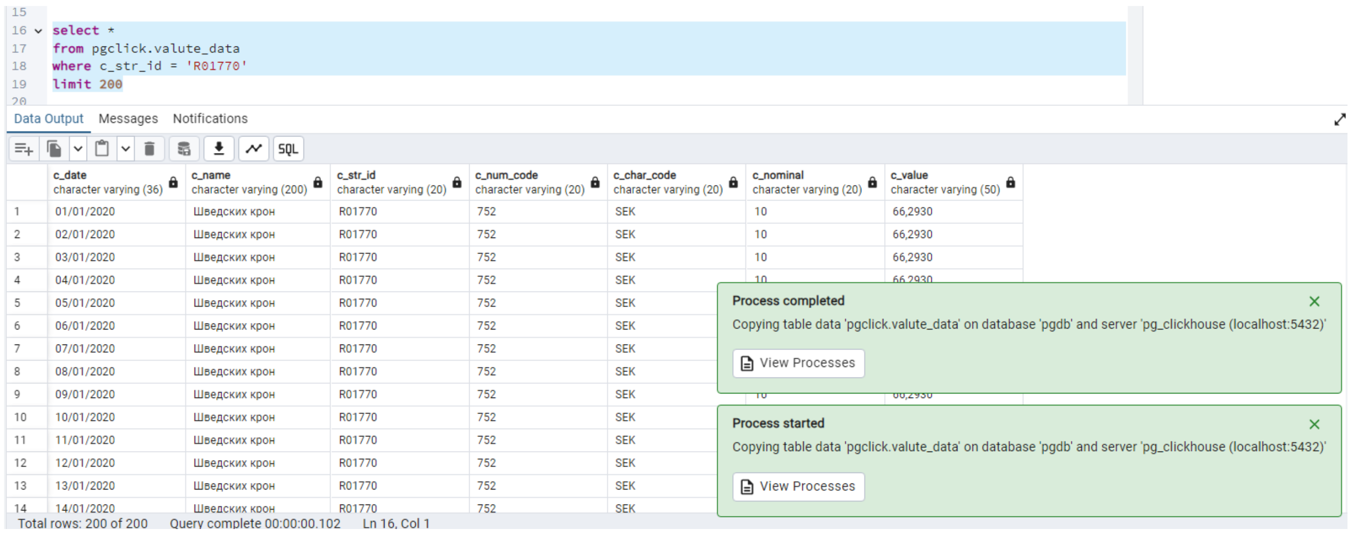
Import/Export data - table 'valute_data'

General | Options | Columns

Import/Export	[✓ Import] Export
Filename	D:\Study\Otus. ClickHouse\19 Интеграция ClickHouse с PostgreSQL\postgres_data.csv
Format	csv
Encoding	UTF8

И проверяем данные:

select *
from pgclick.valute_data
where c_str_id = 'R01770'
limit 200



```
select *
from pgclick.valute_data
where c_str_id = 'R01770'
limit 200
```

Data Output | Messages | Notifications

	c_date character varying (36)	c_name character varying (200)	c_str_id character varying (20)	c_num_code character varying (20)	c_char_code character varying (20)	c_nominal character varying (20)	c_value character varying (50)
1	01/01/2020	Шведских крон	R01770	752	SEK	10	66,2930
2	02/01/2020	Шведских крон	R01770	752	SEK	10	66,2930
3	03/01/2020	Шведских крон	R01770	752	SEK	10	66,2930
4	04/01/2020	Шведских крон	R01770	752	SEK	10	66,2930
5	05/01/2020	Шведских крон	R01770	752	SEK		
6	06/01/2020	Шведских крон	R01770	752	SEK		
7	07/01/2020	Шведских крон	R01770	752	SEK		
8	08/01/2020	Шведских крон	R01770	752	SEK		
9	09/01/2020	Шведских крон	R01770	752	SEK	10	66,2930
10	10/01/2020	Шведских крон	R01770	752	SEK		
11	11/01/2020	Шведских крон	R01770	752	SEK		
12	12/01/2020	Шведских крон	R01770	752	SEK		
13	13/01/2020	Шведских крон	R01770	752	SEK		
14	14/01/2020	Шведских крон	R01770	752	SEK		

Process completed
Copying table data 'pgclick.valute_data' on database 'pgdb' and server 'pg_clickhouse (localhost:5432)'

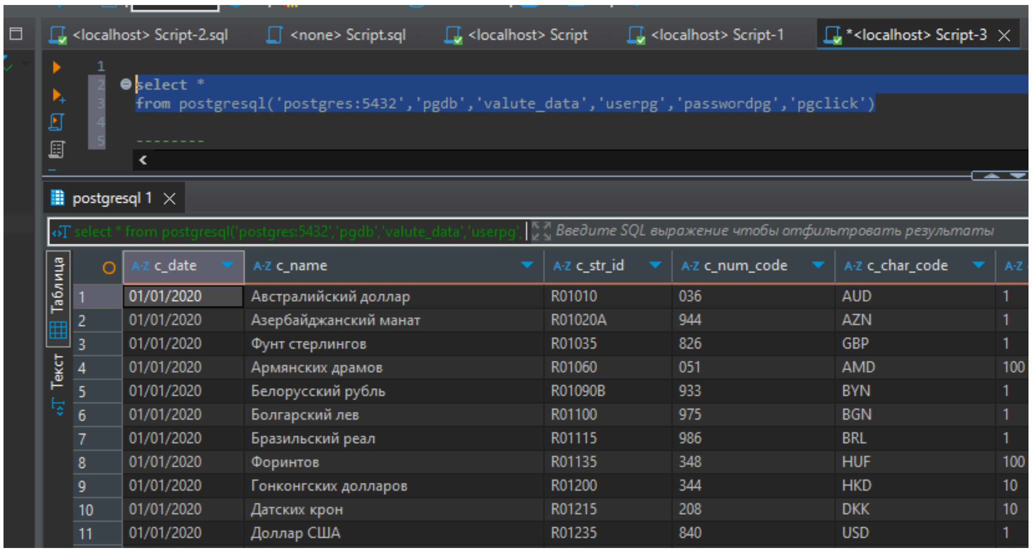
Process started
Copying table data 'pgclick.valute_data' on database 'pgdb' and server 'pg_clickhouse (localhost:5432)'

Total rows: 200 of 200   Query complete 00:00:00.102   Ln 16, Col 1

Делаем выборку данных через ClickHouse.

Выборка данных через функцию postgresql:

```
select *
from postgresql('postgres:5432','pgdb','value_data','userpg','passwordpg','pgclick')
```



<localhost> Script-2.sql | <none> Script.sql | <localhost> Script | <localhost> Script-1 | *<localhost> Script-3

```
select *
from postgresql('postgres:5432','pgdb','value_data','userpg','passwordpg','pgclick')

--------
```

postgresql 1

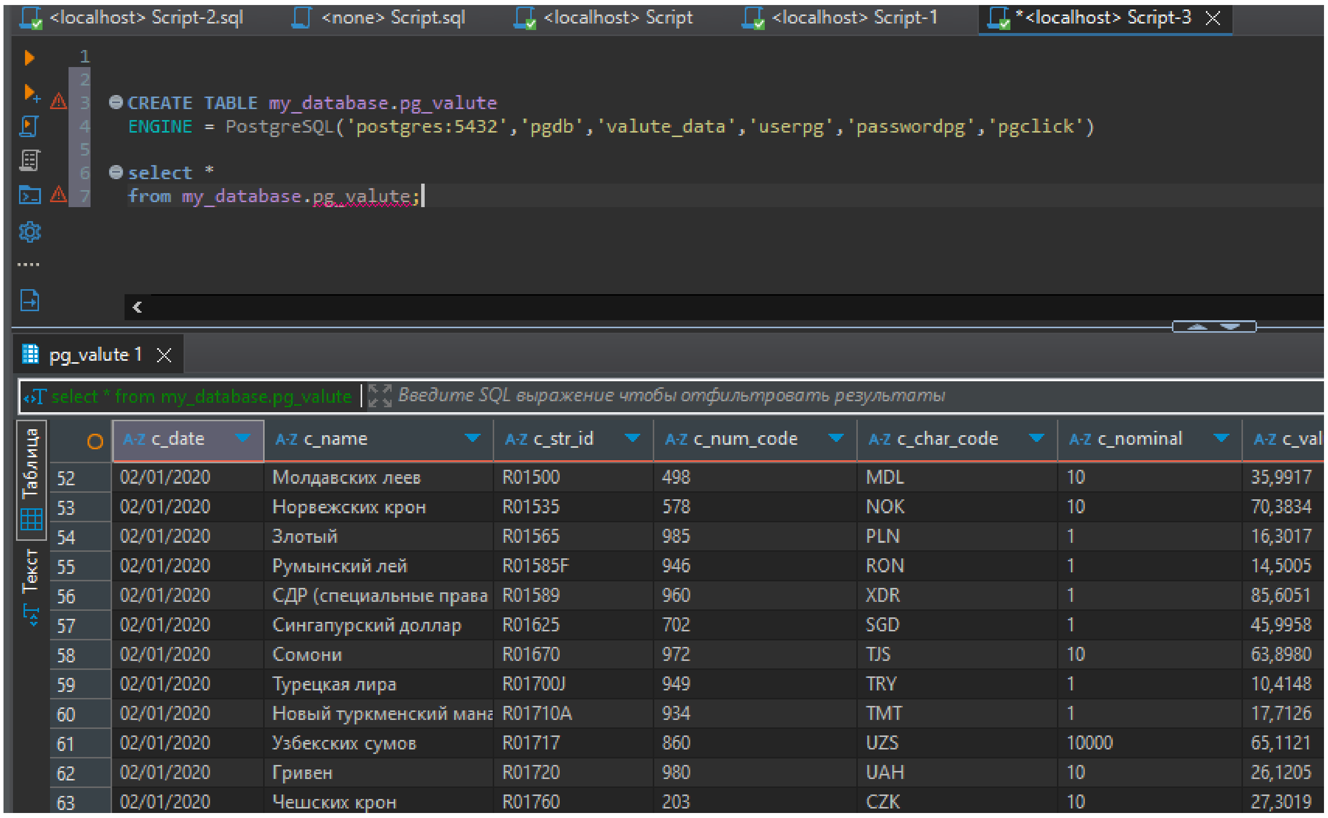
select * from postgresql('postgres:5432','pgdb','value_data','userpg' | Введите SQL выражение чтобы отфильтровать результаты

	c_date	c_name	c_str_id	c_num_code	c_char_code	
1	01/01/2020	Австралийский доллар	R01010	036	AUD	1
2	01/01/2020	Азербайджанский манат	R01020A	944	AZN	1
3	01/01/2020	Фунт стерлингов	R01035	826	GBP	1
4	01/01/2020	Армянских драмов	R01060	051	AMD	100
5	01/01/2020	Белорусский рубль	R01090B	933	BYN	1
6	01/01/2020	Болгарский лев	R01100	975	BGN	1
7	01/01/2020	Бразильский реал	R01115	986	BRL	1
8	01/01/2020	Форинтов	R01135	348	HUF	100
9	01/01/2020	Гонконгских долларов	R01200	344	HKD	10
10	01/01/2020	Датских крон	R01215	208	DKK	10
11	01/01/2020	Доллар США	R01235	840	USD	1

Выборка данных через создание таблицы:

```
CREATE TABLE my_database.pg_value
ENGINE = PostgreSQL('postgres:5432','pgdb','value_data','userpg','passwordpg','pgclick')

select *
from my_database.pg_value;
```



<localhost> Script-2.sql | <none> Script.sql | <localhost> Script | <localhost> Script-1 | *<localhost> Script-3

```
CREATE TABLE my_database.pg_value
ENGINE = PostgreSQL('postgres:5432','pgdb','value_data','userpg','passwordpg','pgclick')

select *
from my_database.pg_value;
```

pg_value 1

select * from my_database.pg_value | Введите SQL выражение чтобы отфильтровать результаты

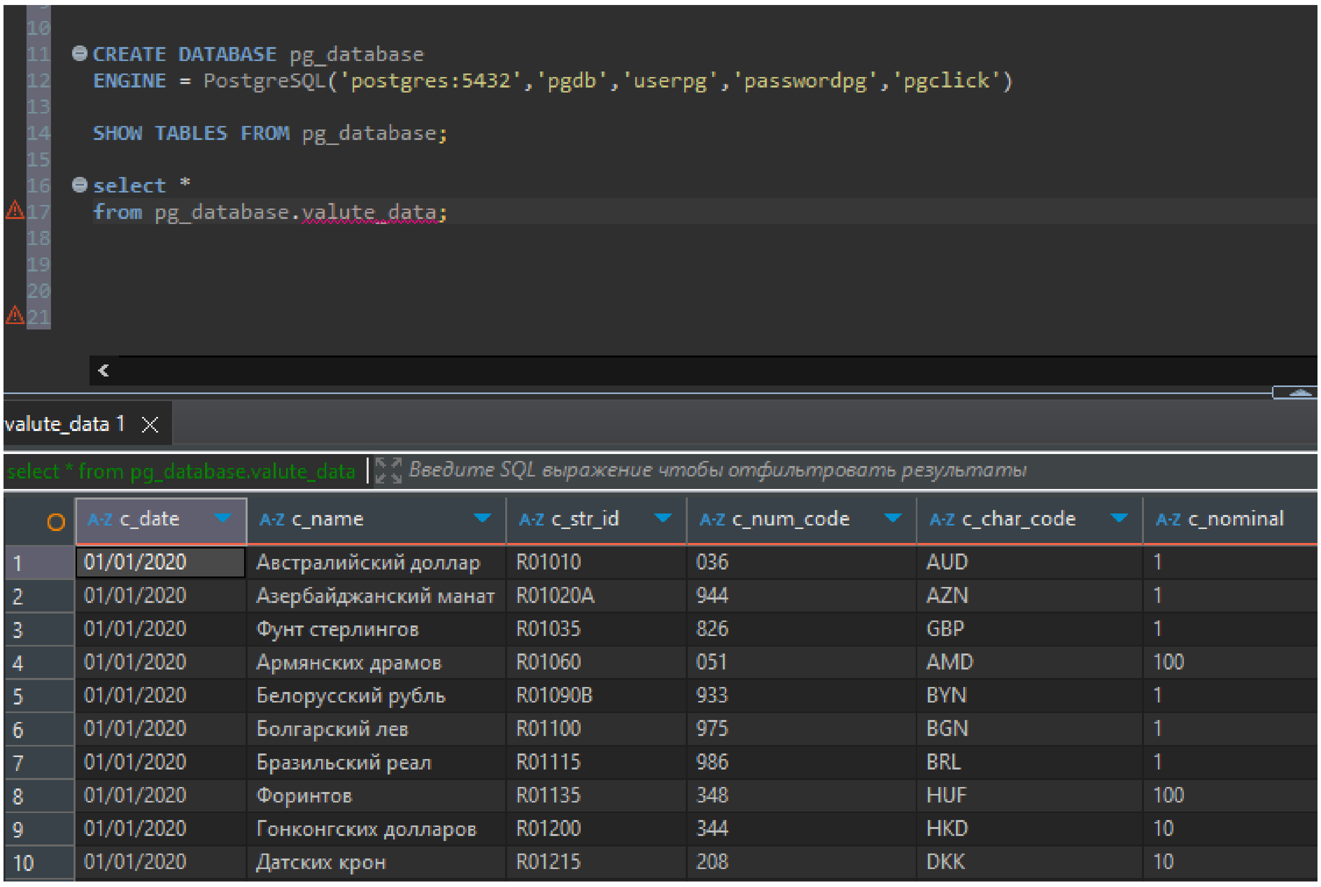
	c_date	c_name	c_str_id	c_num_code	c_char_code	c_nominal	c_val
52	02/01/2020	Молдавских леев	R01500	498	MDL	10	35,9917
53	02/01/2020	Норвежских крон	R01535	578	NOK	10	70,3834
54	02/01/2020	Злотый	R01565	985	PLN	1	16,3017
55	02/01/2020	Румынский лей	R01585F	946	RON	1	14,5005
56	02/01/2020	СДР (специальные права	R01589	960	XDR	1	85,6051
57	02/01/2020	Сингапурский доллар	R01625	702	SGD	1	45,9958
58	02/01/2020	Сомони	R01670	972	TJS	10	63,8980
59	02/01/2020	Турецкая лира	R01700J	949	TRY	1	10,4148
60	02/01/2020	Новый туркменский мана	R01710A	934	TMT	1	17,7126
61	02/01/2020	Узбекских сумов	R01717	860	UZS	10000	65,1121
62	02/01/2020	Гривен	R01720	980	UAH	10	26,1205
63	02/01/2020	Чешских крон	R01760	203	CZK	10	27,3019

Выборка данных через создание базы данных в ClickHouse:

```
CREATE DATABASE pg_database
ENGINE = PostgreSQL('postgres:5432','pgdb','userpg','passwordpg','pgclick')

SHOW TABLES FROM pg_database;

select *
from pg_database.valute_data;
```



```
CREATE DATABASE pg_database
ENGINE = PostgreSQL('postgres:5432','pgdb','userpg','passwordpg','pgclick')

SHOW TABLES FROM pg_database;

select *
from pg_database.valute_data;
```

valute_data 1

select * from pg_database.valute_data

Введите SQL выражение чтобы отфильтровать результаты

	c_date	c_name	c_str_id	c_num_code	c_char_code	c_nominal
1	01/01/2020	Австралийский доллар	R01010	036	AUD	1
2	01/01/2020	Азербайджанский манат	R01020A	944	AZN	1
3	01/01/2020	Фунт стерлингов	R01035	826	GBP	1
4	01/01/2020	Армянских драмов	R01060	051	AMD	100
5	01/01/2020	Белорусский рубль	R01090B	933	BYN	1
6	01/01/2020	Болгарский лев	R01100	975	BGN	1
7	01/01/2020	Бразильский реал	R01115	986	BRL	1
8	01/01/2020	Форинтов	R01135	348	HUF	100
9	01/01/2020	Гонконгских долларов	R01200	344	HKD	10
10	01/01/2020	Датских крон	R01215	208	DKK	10