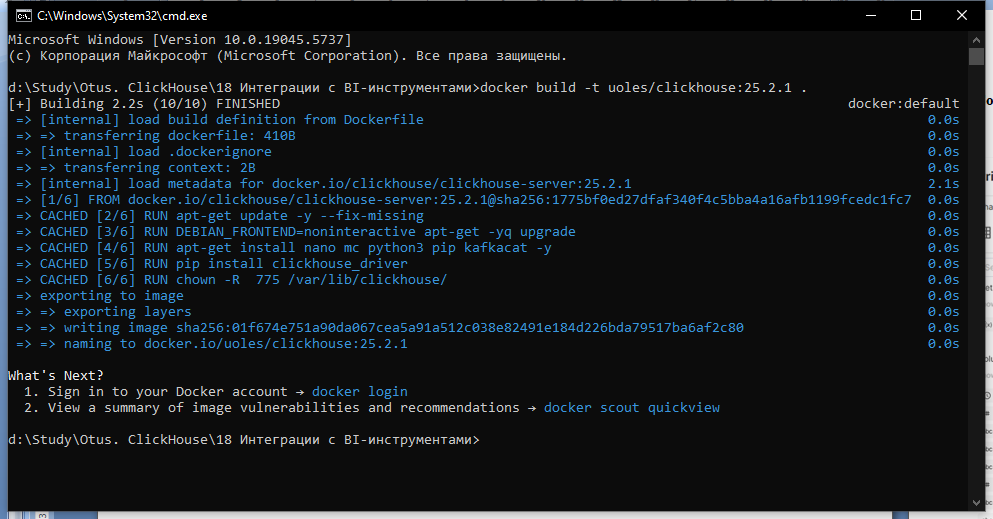
**18 ДЗ - Интеграции с BI-инструментами**

**Собираем Dockerfile для clickhouse.**

Dockerfile:  
  
FROM clickhouse/clickhouse-server:25.2.1   
MAINTAINER Maksim Kulikov [max.uoles@rambler.ru](mailto: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 .

****

**Поднимаем superset с clickhouse.**

Скачиваем репозиторий superset:

git clone <https://github.com/apache/superset.git>  
cd superset  
git checkout 2.1.0

Редактируем файл docker-compose-non-dev.yml и сохраняем как docker-compose-clickhouse.yml:  
- добавил установку clickhouse-connect в superset  
- добавил разворачивание clickhouse-server  
- добавил зависимость от clickhouse-server для superset и линк

docker-compose-clickhouse.yml:

x-superset-image: &superset-image apache/superset:${TAG:-latest-dev}

x-superset-depends-on: &superset-depends-on

- db

- redis

x-superset-only-depends-on: &superset-only-depends-on

- db

- redis

- clickhouse-server

x-superset-volumes: &superset-volumes

# /app/pythonpath\_docker will be appended to the PYTHONPATH in the final container

- ./docker:/app/docker

- superset\_home:/app/superset\_home

version: "3.7"

services:

redis:

image: redis:7

container\_name: superset\_cache

restart: unless-stopped

volumes:

- redis:/data

db:

env\_file: docker/.env-non-dev

image: postgres:14

container\_name: superset\_db

restart: unless-stopped

volumes:

- db\_home:/var/lib/postgresql/data

superset:

env\_file: docker/.env-non-dev

image: \*superset-image

container\_name: superset\_app

command: [sh, -c, "pip install clickhouse-connect && /app/docker/docker-bootstrap.sh app-gunicorn"]

user: "root"

restart: unless-stopped

ports:

- 8088:8088

depends\_on: \*superset-only-depends-on

volumes: \*superset-volumes

links:

- clickhouse-server

superset-init:

image: \*superset-image

container\_name: superset\_init

command: ["/app/docker/docker-init.sh"]

env\_file: docker/.env-non-dev

depends\_on: \*superset-depends-on

user: "root"

volumes: \*superset-volumes

healthcheck:

disable: true

superset-worker:

image: \*superset-image

container\_name: superset\_worker

command: ["/app/docker/docker-bootstrap.sh", "worker"]

env\_file: docker/.env-non-dev

restart: unless-stopped

depends\_on: \*superset-depends-on

user: "root"

volumes: \*superset-volumes

healthcheck:

test: ["CMD-SHELL", "celery inspect ping -A superset.tasks.celery\_app:app -d celery@$$HOSTNAME"]

superset-worker-beat:

image: \*superset-image

container\_name: superset\_worker\_beat

command: ["/app/docker/docker-bootstrap.sh", "beat"]

env\_file: docker/.env-non-dev

restart: unless-stopped

depends\_on: \*superset-depends-on

user: "root"

volumes: \*superset-volumes

healthcheck:

disable: true

clickhouse-server:

container\_name: uoles-clickhouse-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

volumes:

superset\_home:

external: false

db\_home:

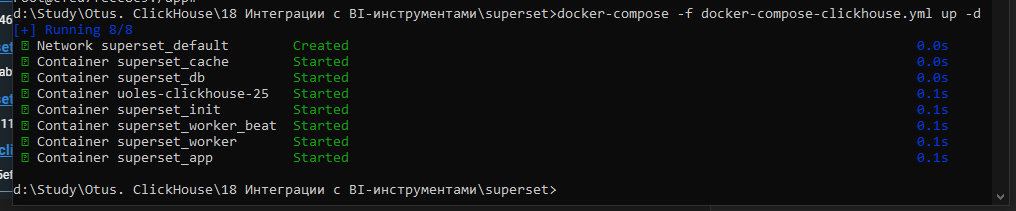
external: false

redis:

external: false

Поднимаем приложения командой:

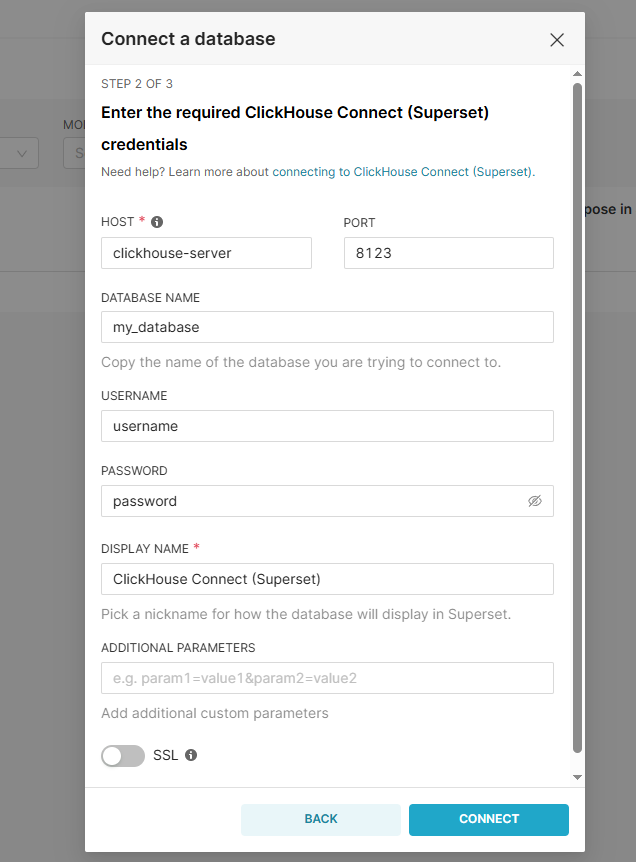
docker-compose -f docker-compose-clickhouse.yml up -d

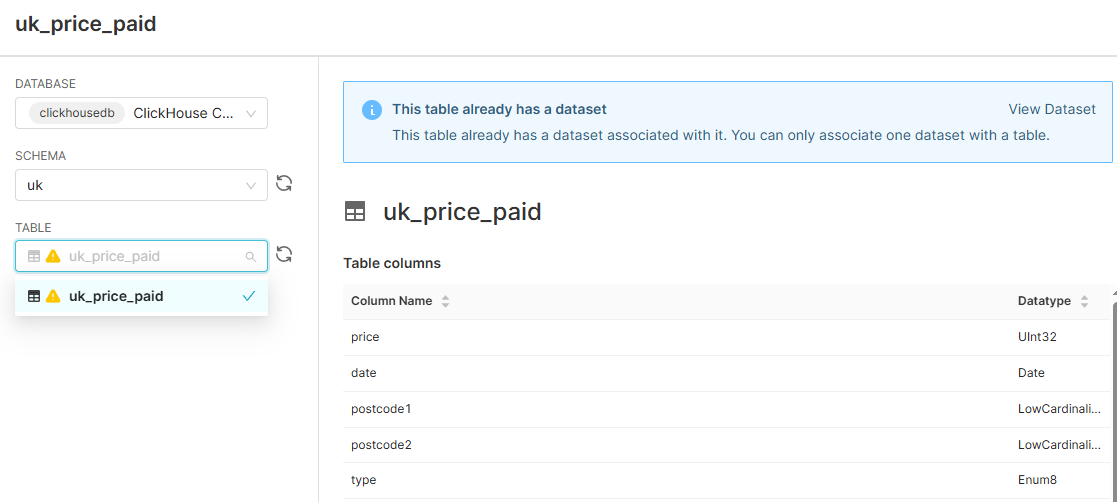


Проверяем в логах установку clickhouse-connect:

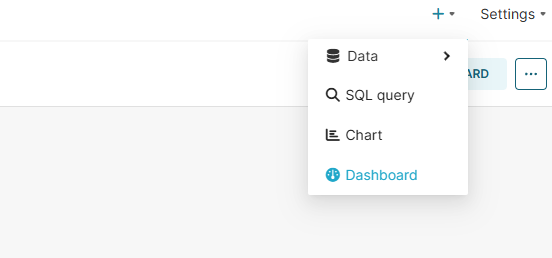


**Создаем dataset в superset.**

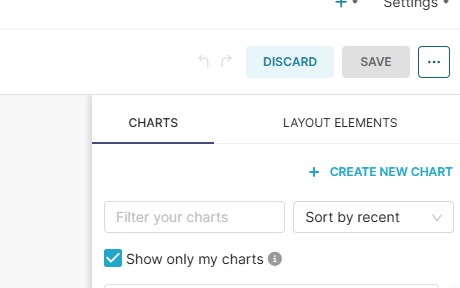


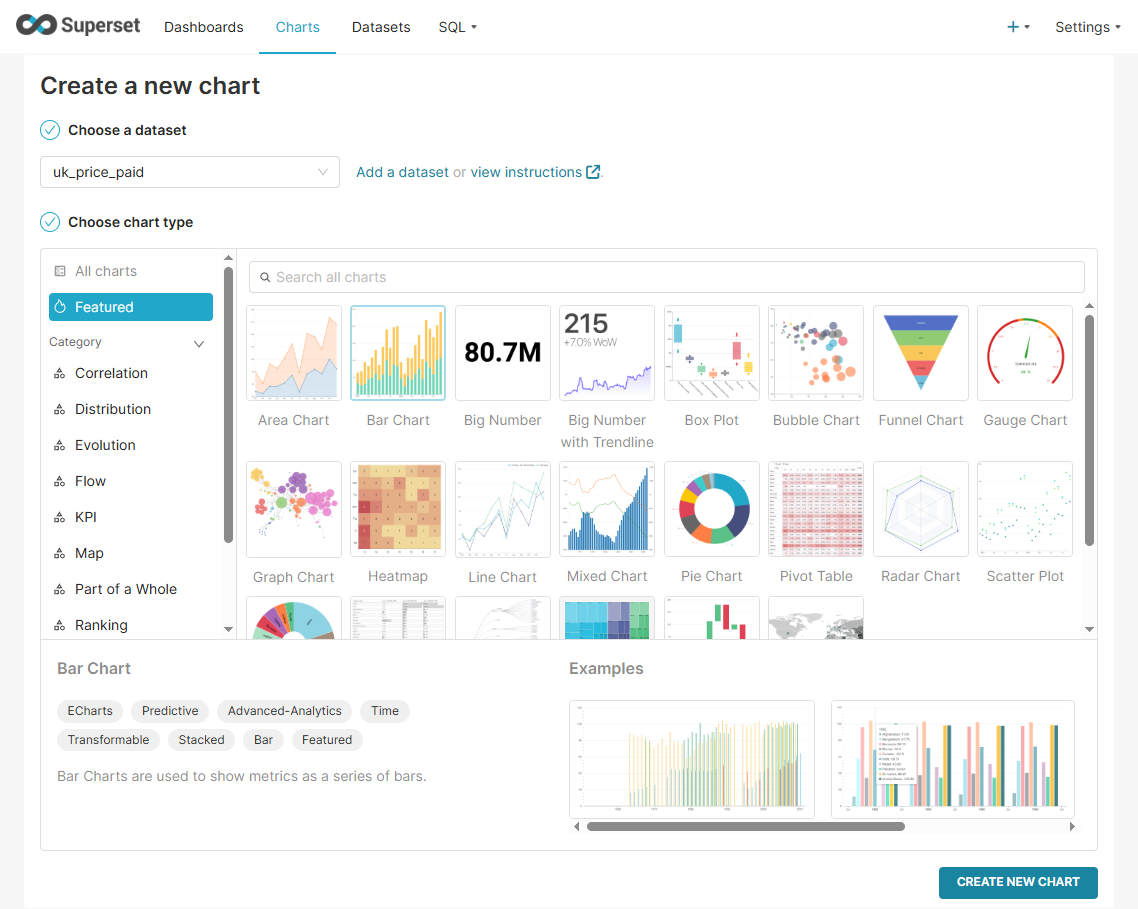


Далее создаем дашборд для графиков:



Жмем CREATE NEW CHART для создания нового графика



****

Выбираем график и созданный ранее датасет.

**Добавление данных в clickhouse.**

Я использовал стандартный набор данных – стоимость жилья в Англии:

<https://clickhouse.com/docs/getting-started/example-datasets/uk-price-paid>

Загрузил ежемесячные изменения:  
<https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads>

Создание схемы и таблицы:

CREATE DATABASE uk;

CREATE TABLE uk.uk\_price\_paid

(

price UInt32,

date Date,

postcode1 LowCardinality(String),

postcode2 LowCardinality(String),

type Enum8('terraced' = 1, 'semi-detached' = 2, 'detached' = 3, 'flat' = 4, 'other' = 0),

is\_new UInt8,

duration Enum8('freehold' = 1, 'leasehold' = 2, 'unknown' = 0),

addr1 String,

addr2 String,

street LowCardinality(String),

locality LowCardinality(String),

town LowCardinality(String),

district LowCardinality(String),

county LowCardinality(String)

)

ENGINE = MergeTree

ORDER BY (postcode1, postcode2, addr1, addr2);

INSERT INTO uk.uk\_price\_paid

SELECT

toUInt32(price\_string) AS price,

parseDateTimeBestEffortUS(time) AS date,

splitByChar(' ', postcode)[1] AS postcode1,

splitByChar(' ', postcode)[2] AS postcode2,

transform(a, ['T', 'S', 'D', 'F', 'O'], ['terraced', 'semi-detached', 'detached', 'flat', 'other']) AS type,

b = 'Y' AS is\_new,

transform(c, ['F', 'L', 'U'], ['freehold', 'leasehold', 'unknown']) AS duration,

addr1,

addr2,

street,

locality,

town,

district,

county

FROM url(

'http://prod.publicdata.landregistry.gov.uk.s3-website-eu-west-1.amazonaws.com/pp-monthly-update-new-version.csv',

'CSV',

'uuid\_string String,

price\_string String,

time String,

postcode String,

a String,

b String,

c String,

addr1 String,

addr2 String,

street String,

locality String,

town String,

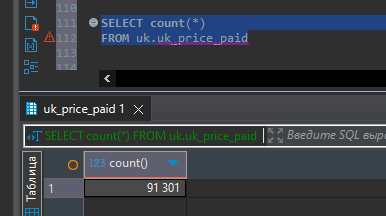
district String,

county String,

d String,

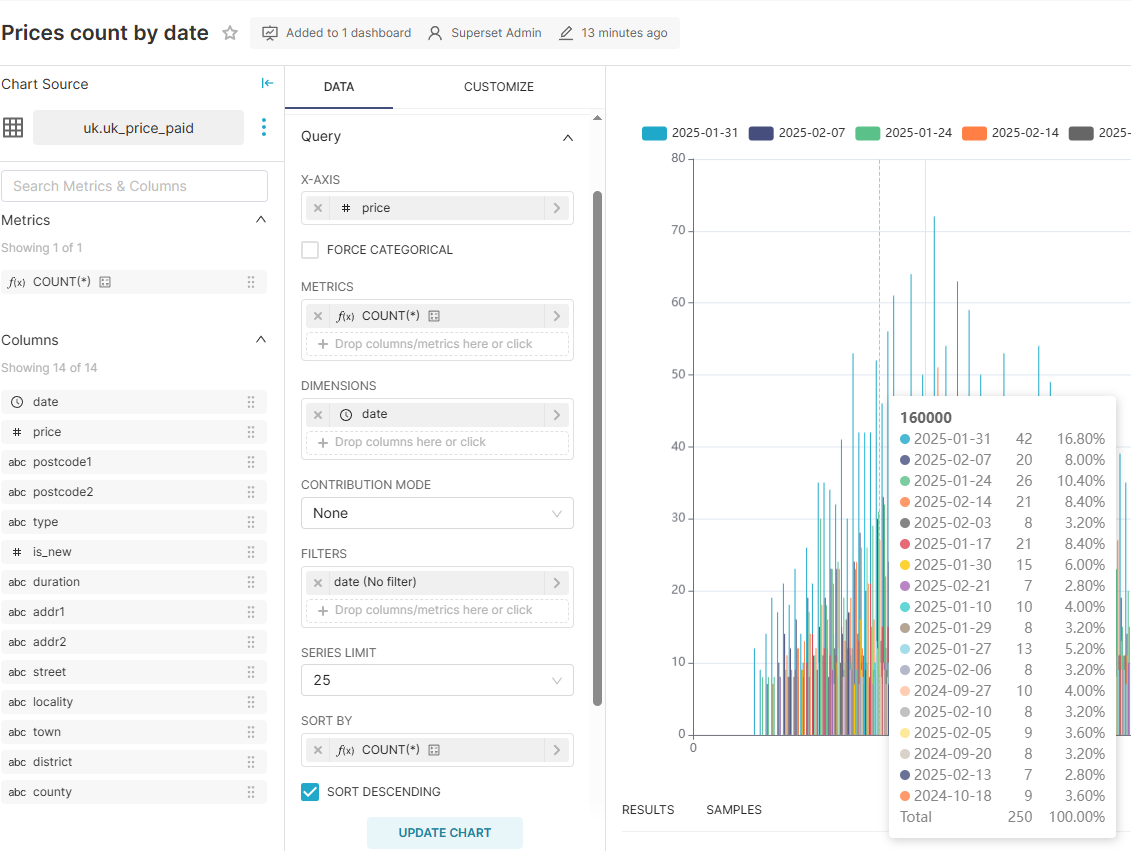
e String'

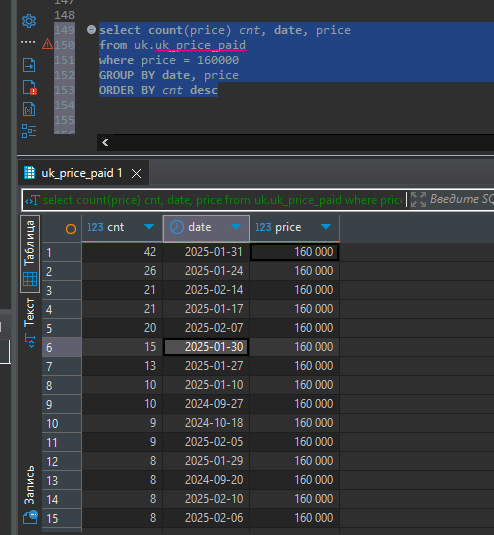
) SETTINGS max\_http\_get\_redirects=10;



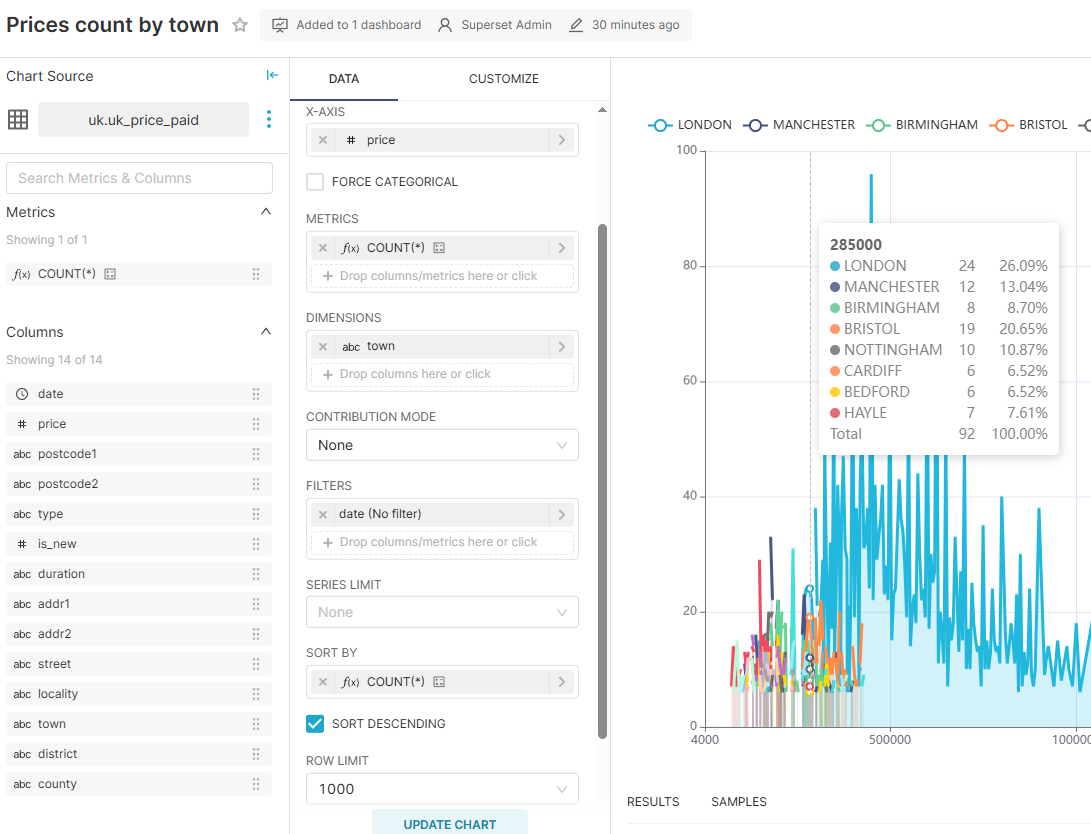
**Создание графиков и их проверка запросом.**

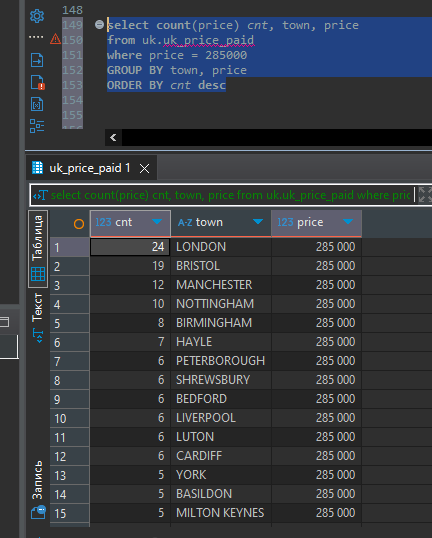
1. Bar – выборка кол-ва цен жилья относительно даты.



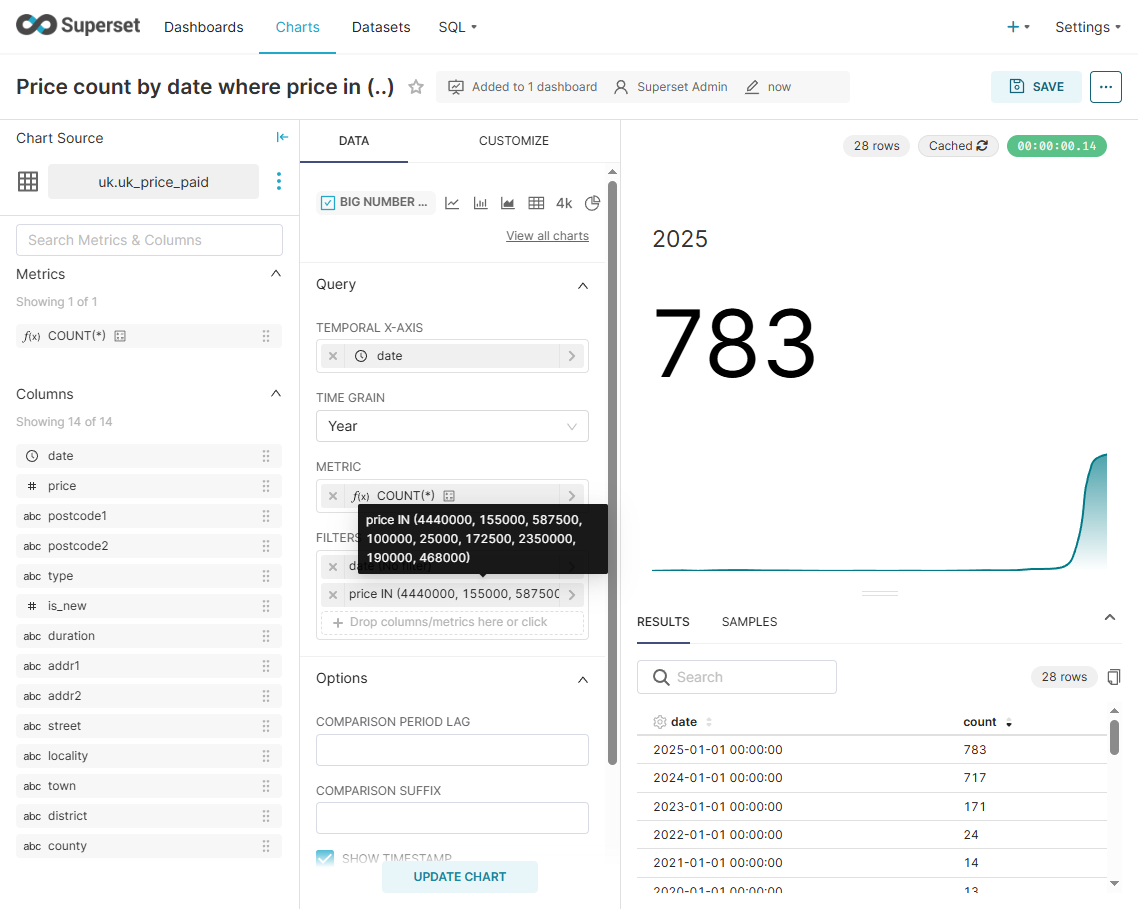


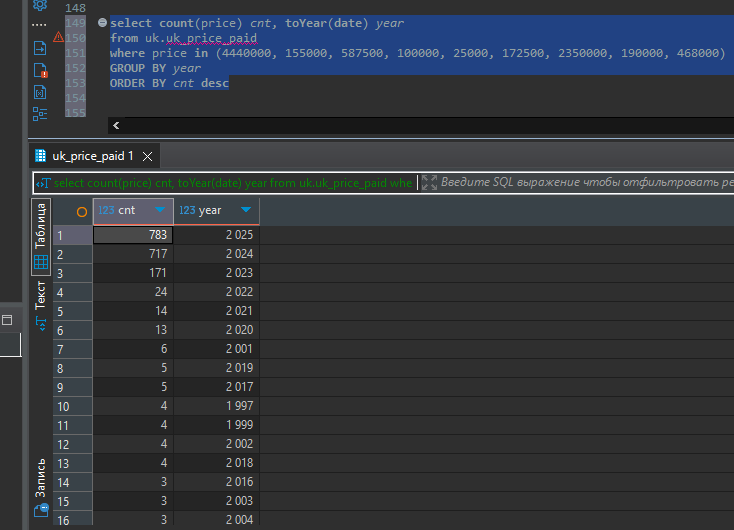
1. Line – выборка кол-ва цен жилья относительно города.



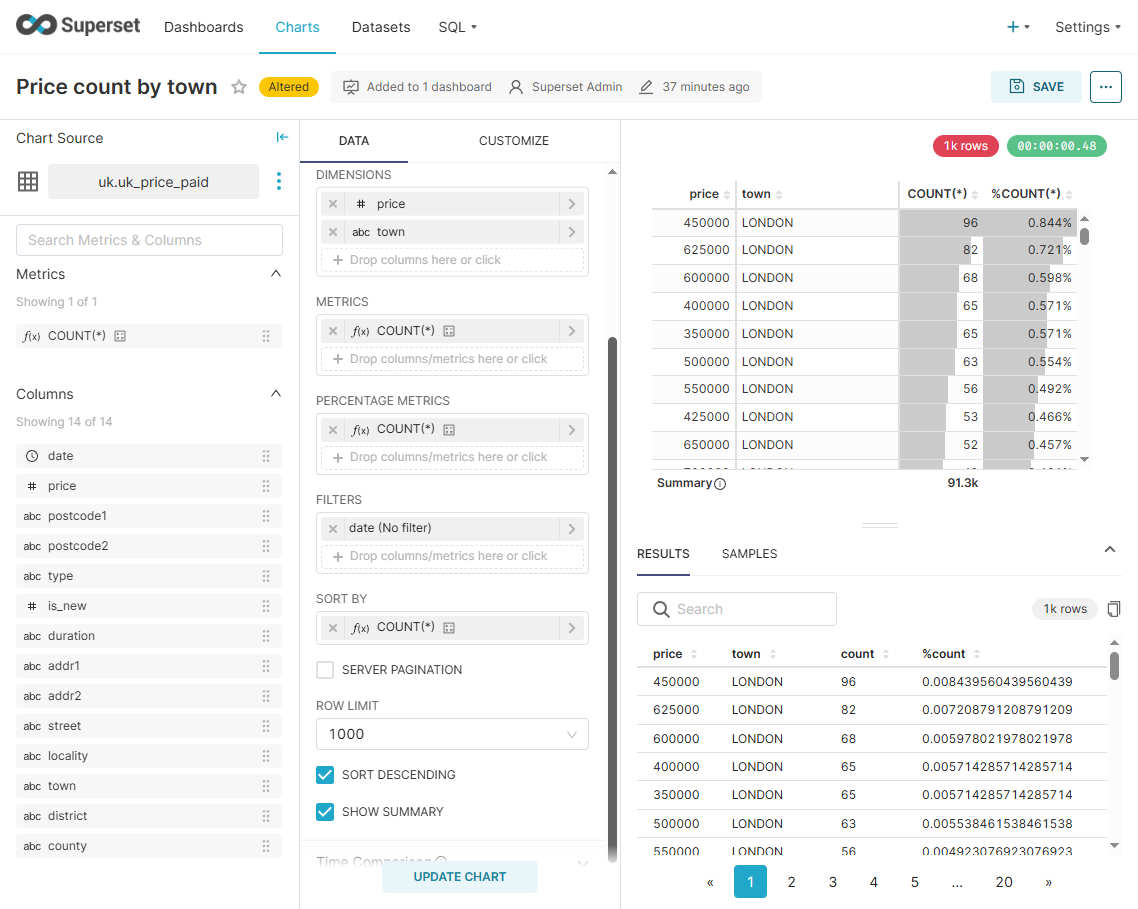


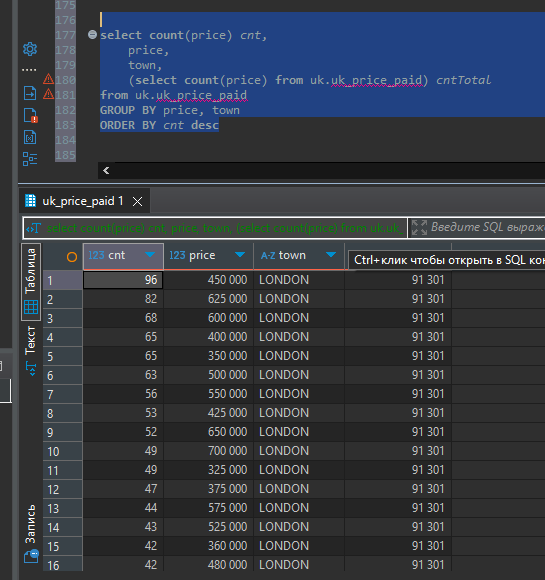
1. Bug number with trend line – выборка кол-ва жилья по годам, цена которого находится в массиве значений.





1. Table – выборка кол-ва цен жилья относительно города.





1. Sunburst chart – выборка кол-ва цен жилья относительно города.

