

Департамент образования и науки города Москвы  
Государственное автономное образовательное учреждение высшего  
образования города Москвы  
«Московский городской педагогический университет» Институт  
цифрового образования  
Департамент информатики, управления и технологий

**ДИСЦИПЛИНА:**

«Проектный практикум по разработке ETL-решений»

**Практическая работа № 6 Тема:**

«Оркестровка конвейера данных».

Выполнила: Нестратова А.М., АДЭУ-201

Преподаватель: Босенко Т.М.

Москва

2024

## Постановка задачи

6.1.1. Развернуть ВМ [ubuntu\\_mgpu.ova](#) в [VirtualBox](#).

6.1.2. Клонировать на ПК задание **Бизнес-кейс «StockSense»** в домашний каталог ВМ.

`git clone` <https://github.com/BosenkoTM/workshop-on-ETL.git>

6.1.3. Запустить контейнер с кейсом, изучить основные элементы **DAG** в **Apache Airflow**.

- Создать **DAG** согласно алгоритму, который предоставит преподаватель.
- Изучить логи, выполненного DAG. Скачать логи из контейнера в основную ОС.

Пример `docker-compose.yml` создает базу данных в Postgres:

- Host: **localhost**
- Port: **5433**
- Username: **airflow**
- Password: **airflow**
- Database: **airflow**

Эта база данных инициализируется таблицей **pageview\_counts**.

6.1.4. Агрегированные данные бизнес-процесса, полученные в результате работы **DAG** в **Apache Airflow**, выгрузить в **Postgr SQL**.

6.1.5. Спроектировать верхнеуровневую архитектуру аналитического решения задания **Бизнес-кейса «StockSense»** в **draw.io**. Необходимо использовать:

- **Source Layer** - слой источников данных.
- **Storage Layer** - слой хранения данных.
- **Business Layer** - слой для доступа к данным пользователей.

6.1.6. Спроектировать архитектуру **DAG Бизнес-кейса «StockSense»** в **draw.io**. Необходимо использовать:

- **Source Layer** - слой источников данных.
- **Storage Layer** - слой хранения данных.
- **Business Layer** - слой для доступа к данным пользователей.

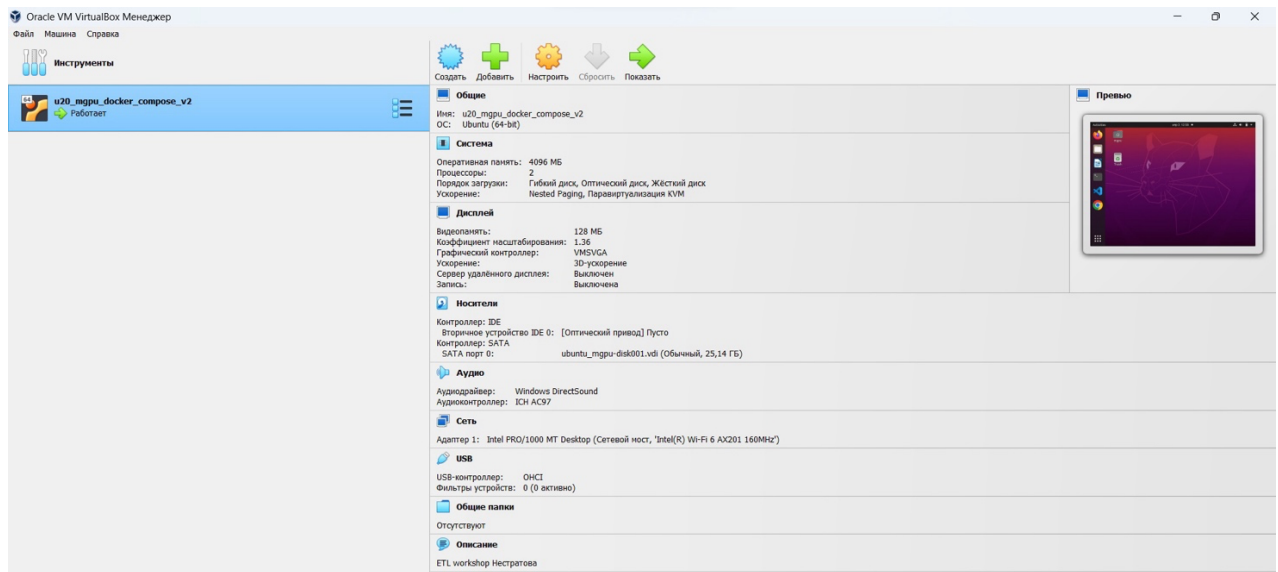
6.1.7. Построить диаграмму Ганта работы **DAG** в **Apache Airflow**.

6.1.8. Результаты исследований представить в виде файла **ФИО-06.pdf**, в котором отражены следующие результаты:

- постановка задачи;
- исходный код всех DAGs, которые требовались для решения задачи, а также представить граф **DAG** в **Apache Airflow**;
- верхнеуровневая архитектура задания **Бизнес-кейса «StockSense»**, выполненная в **draw.io**;
- архитектура **DAG Бизнес-кейса «StockSense»** , выполненная в **draw.io**;
- диаграмма Ганта **DAG** в **Apache Airflow**;
- ERD-схема базы данных Postgr SQL;
- SQL-запросы, позволяющие проверить наличие выгруженных агрегированных данных бизнес-задачи.

## Самостоятельная работа

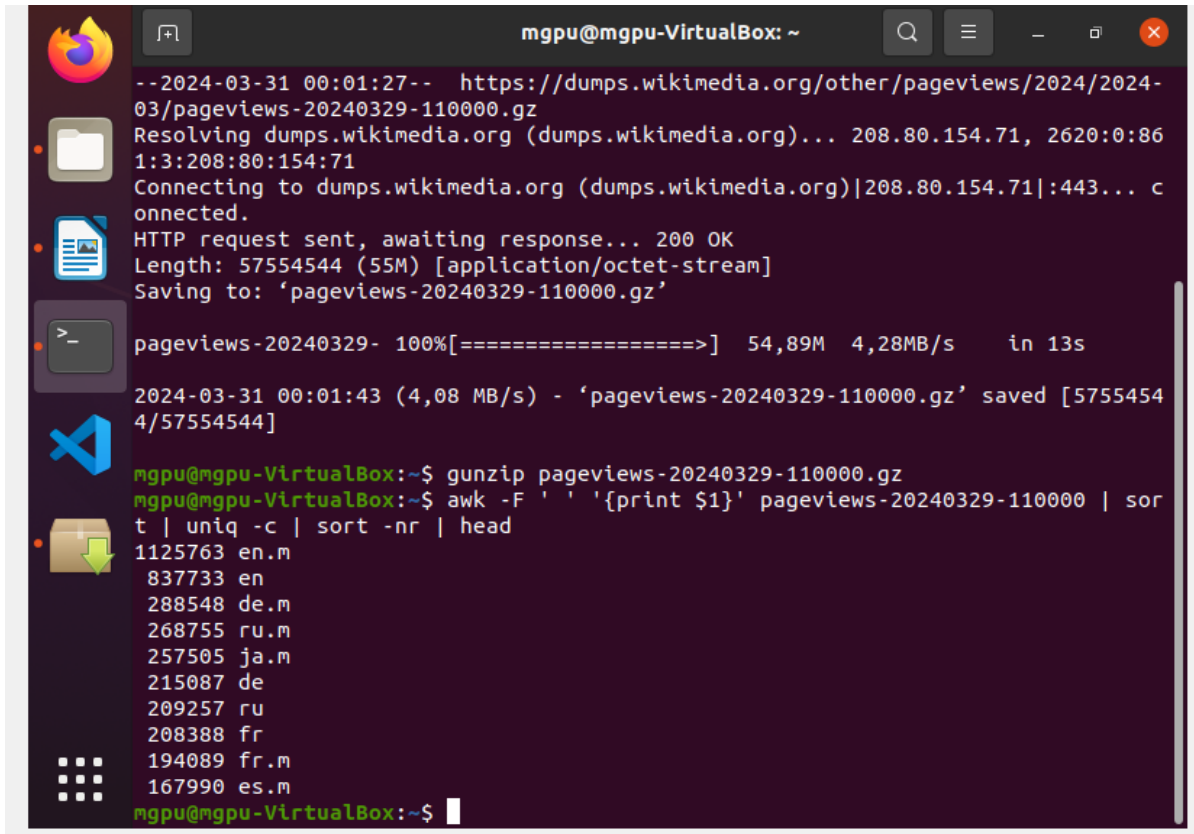
### Разворачиваем VM ubuntu\_mgpu.ova в VirtualBox.



Клонируем на ПК задание Бизнес-кейс «StockSense» в домашний каталог VM. Скачиваем данные наиболее часто используемых кодах доменов, разархивируем.

```
mgpu@mgpu-VirtualBox: ~  
mgpu@mgpu-VirtualBox:~$ git clone https://github.com/BosenkoTM/workshop-on-ETL.  
git  
Cloning into 'workshop-on-ETL'...  
remote: Enumerating objects: 170, done.  
remote: Counting objects: 100% (60/60), done.  
remote: Compressing objects: 100% (27/27), done.  
remote: Total 170 (delta 32), reused 60 (delta 32), pack-reused 110  
Receiving objects: 100% (170/170), 51.53 KiB | 533.00 KiB/s, done.  
Resolving deltas: 100% (70/70), done.  
mgpu@mgpu-VirtualBox:~$ wget https://dumps.wikimedia.org/other/pageviews/2024/2  
024-03/pageviews-20240329-110000.gz  
--2024-03-31 00:01:27-- https://dumps.wikimedia.org/other/pageviews/2024/2024-  
03/pageviews-20240329-110000.gz  
Resolving dumps.wikimedia.org (dumps.wikimedia.org)... 208.80.154.71, 2620:0:86  
1:3:208:80:154:71  
Connecting to dumps.wikimedia.org (dumps.wikimedia.org)|208.80.154.71|:443... c  
onected.  
HTTP request sent, awaiting response... 200 OK  
Length: 57554544 (55M) [application/octet-stream]  
Saving to: 'pageviews-20240329-110000.gz'  
  
pageviews-20240329- 100%[=====] 54,89M 4,28MB/s in 13s  
  
2024-03-31 00:01:43 (4,08 MB/s) - 'pageviews-20240329-110000.gz' saved [5755454  
4/57554544]  
  
mgpu@mgpu-VirtualBox:~$ gunzip pageviews-20240329-110000.gz
```

Выводим первые строки данных за 29 марта 2024 года 11:00.



```
mgpu@mgpu-VirtualBox: ~  
--2024-03-31 00:01:27-- https://dumps.wikimedia.org/other/pageviews/2024/2024-03/pageviews-20240329-110000.gz  
Resolving dumps.wikimedia.org (dumps.wikimedia.org)... 208.80.154.71, 2620:0:861:3:208:80:154:71  
Connecting to dumps.wikimedia.org (dumps.wikimedia.org)|208.80.154.71|:443... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 57554544 (55M) [application/octet-stream]  
Saving to: 'pageviews-20240329-110000.gz'  
  
pageviews-20240329- 100%[=====] 54,89M 4,28MB/s in 13s  
  
2024-03-31 00:01:43 (4,08 MB/s) - 'pageviews-20240329-110000.gz' saved [57554544/57554544]  
  
mgpu@mgpu-VirtualBox:~$ gunzip pageviews-20240329-110000.gz  
mgpu@mgpu-VirtualBox:~$ awk -F ' ' '{print $1}' pageviews-20240329-110000 | sort | uniq -c | sort -nr | head  
1125763 en.m  
837733 en  
288548 de.m  
268755 ru.m  
257505 ja.m  
215087 de  
209257 ru  
208388 fr  
194089 fr.m  
167990 es.m  
mgpu@mgpu-VirtualBox:~$
```

Меняем файл listing\_4\_20.py в папке dags.

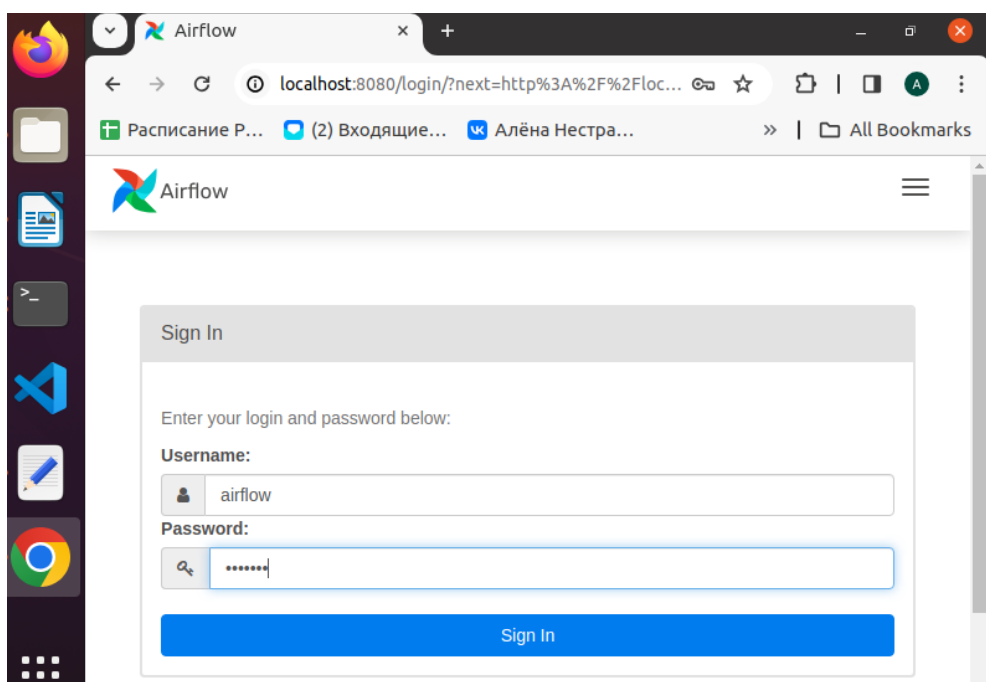


```
*listing_4_20.py  
~/workshop-on-ETL/business_case_stocksense/dags  
Open Save  
45 task_id="extract_gz", bash_command="gunzip --force /tmp/-  
wikipageviews.gz", dag=dag  
46 )  
47  
48  
49 def _fetch_pageviews(pagenames, execution_date):  
50     result = dict.fromkeys(pagenames, 0)  
51     with open("/tmp/wikipageviews", "r") as f:  
52         for line in f:  
53             domain_code, page_title, view_counts, _ = line.split(" ")  
54             if domain_code == "ru" and page_title in pagenames:  
55                 result[page_title] = view_counts  
56  
57     with open("/tmp/postgres_query.sql", "w") as f:  
58         for pagename, pageviewcount in result.items():  
59             f.write(  
60                 "INSERT INTO pageview_counts VALUES ("  
61                     f"'{pagename}', {pageviewcount}, '{execution_date}'"  
62                 ");\n"  
63             )  
64  
65  
66 fetch_pageviews = PythonOperator(  
67     task_id="fetch_pageviews",  
68     python_callable=_fetch_pageviews,  
69     op_kwargs={"pagenames": {"Telegram", "Ozon", "Avito", "Wildberries",  
"Yandex"}},  
70     dag=dag
```

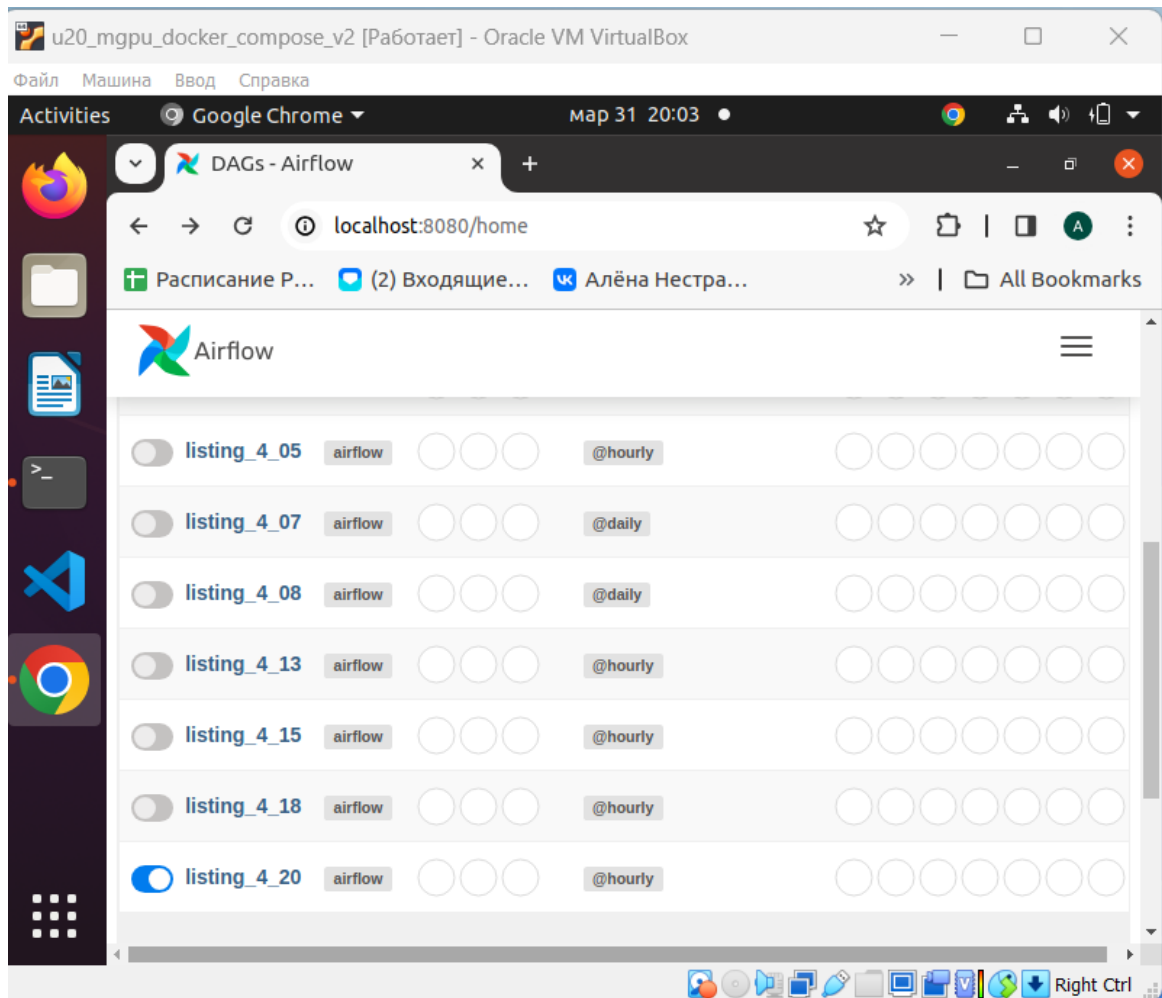
Запускаем контейнер.

```
mgpu@mgpu-VirtualBox: ~/workshop-on-ETL/business_cas...
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_stocksense$ sudo docker compose up -d
[sudo] password for mgpu:
[+] Building 5.3s (12/12) FINISHED                                docker:default
=> [webserver internal] load build definition from Dockerfile      0.0s
=> => transferring dockerfile: 208B                                0.0s
=> [scheduler internal] load build definition from Dockerfile     0.1s
=> => transferring dockerfile: 208B                                0.0s
=> [init internal] load build definition from Dockerfile           0.1s
=> => transferring dockerfile: 208B                                0.0s
=> [init internal] load metadata for docker.io/apache/airflow:2.0.0-pyt 0.0s
=> [webserver internal] load .dockerignore                         0.1s
=> => transferring context: 2B                                      0.0s
=> [scheduler internal] load .dockerignore                         0.1s
=> => transferring context: 2B                                      0.0s
=> [init internal] load .dockerignore                             0.1s
=> => transferring context: 2B                                      0.0s
=> [webserver 1/2] FROM docker.io/apache/airflow:2.0.0-python3.8 0.3s
=> [scheduler 2/2] RUN pip install --user --no-cache-dir apache-air 4.7s
=> [webserver] exporting to image                                  0.1s
=> => exporting layers                                             0.1s
=> => writing image sha256:4cc7a9943e5ee09d8615599ba06253056d1e5775ed55 0.0s
=> => naming to docker.io/library/business_case_stocksense-webserver 0.0s
=> [init] exporting to image                                       0.1s
=> => exporting layers                                             0.1s
=> => writing image sha256:e252b95a3e267388cdc9af0157e482d873b2d0a5d36b 0.0s
=> => naming to docker.io/library/business_case_stocksense-init    0.0s
=> [scheduler] exporting to image                                  0.1s
=> => exporting layers                                             0.1s
=> => writing image sha256:c65fef19733a53d200dc51d1804313485bda58e390bd 0.0s
=> => naming to docker.io/library/business_case_stocksense-scheduler 0.0s
[+] Running 5/7
  ... Network business_case_stocksense_default                Created           1.9s
  ... Volume "business_case_stocksense_logs"                  Created           1.8s
  ... Container business_case_stocksense-postgres-1           Started           0.8s
  ... Container business_case_stocksense-wiki_results-1       Started           0.7s
  ... Container business_case_stocksense-scheduler-1          Started           1.6s
  ... Container business_case_stocksense-init-1                Started           1.3s
  ... Container business_case_stocksense-webserver-1           Started           1.6s
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_stocksense$
```

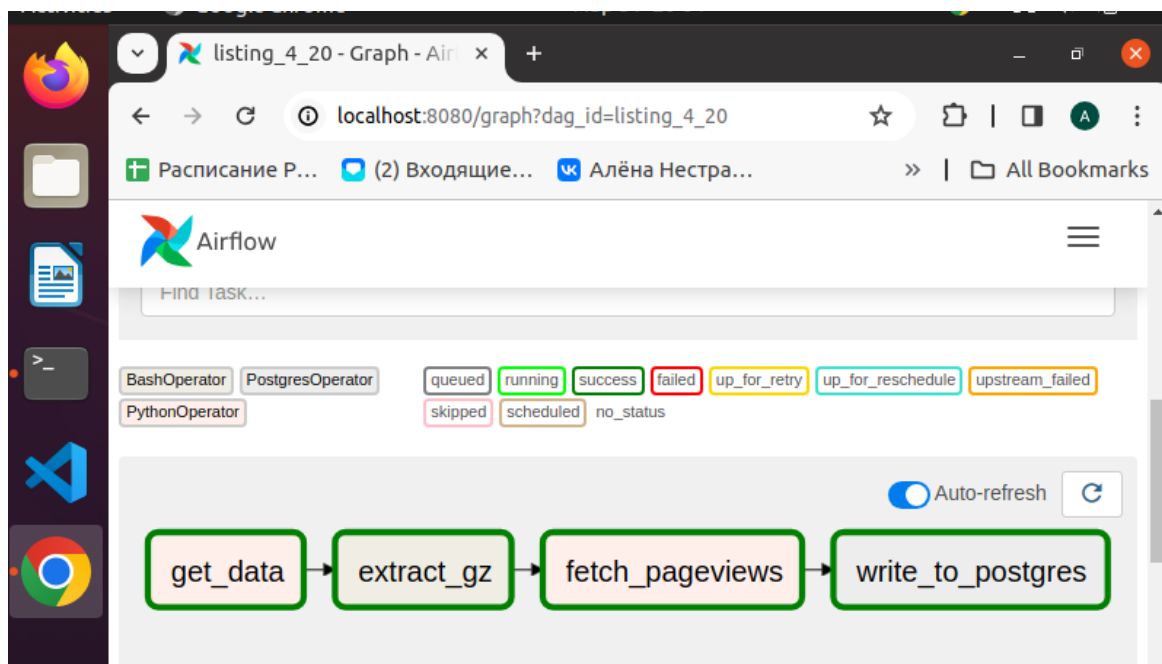
Заходим на <http://localhost:8080/>



Запускаем только файл listing\_4\_20.py

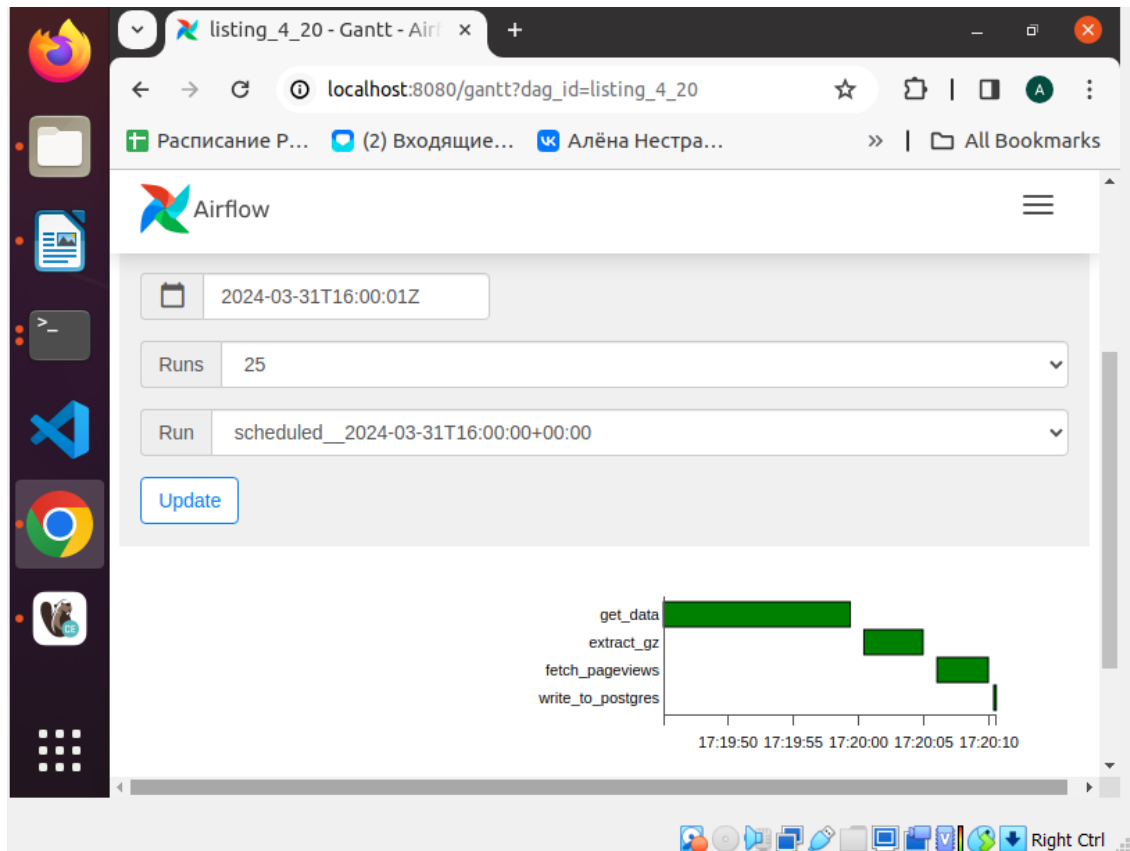


Проверяем, что все стадии отработали





## Смотрим Диаграмму Ганта

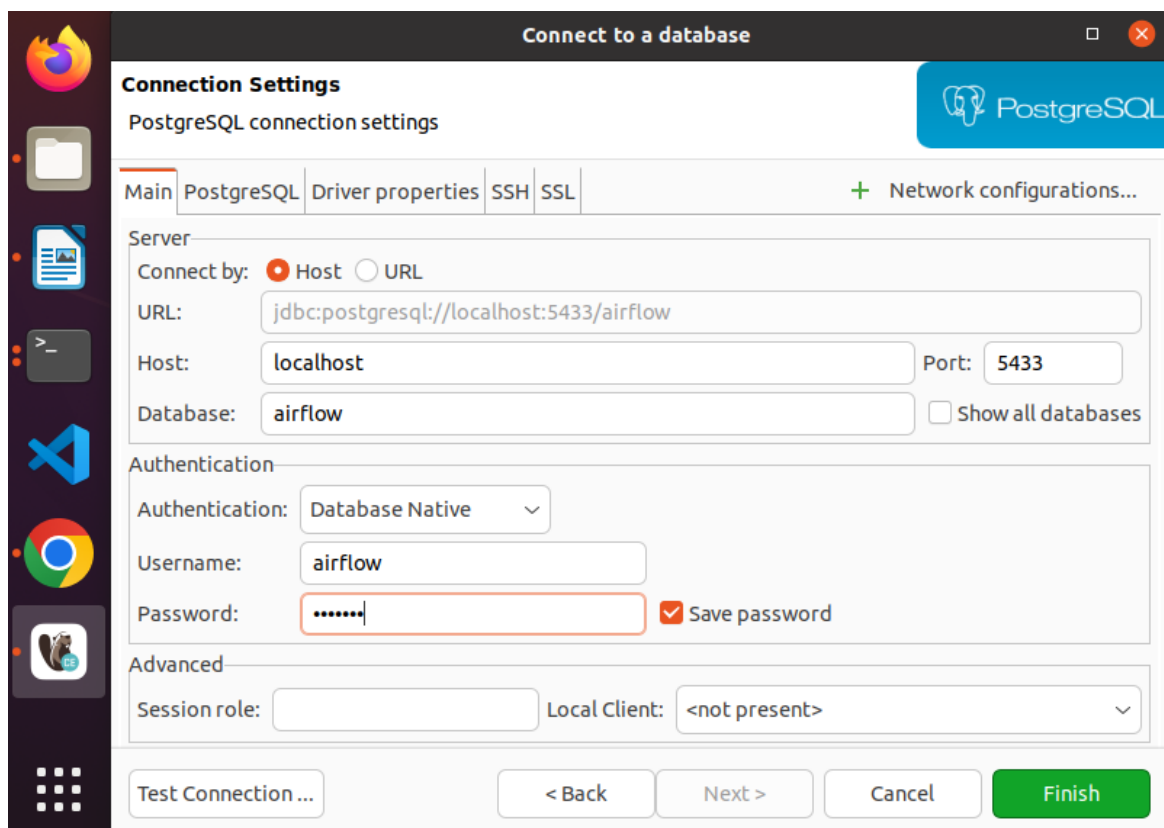
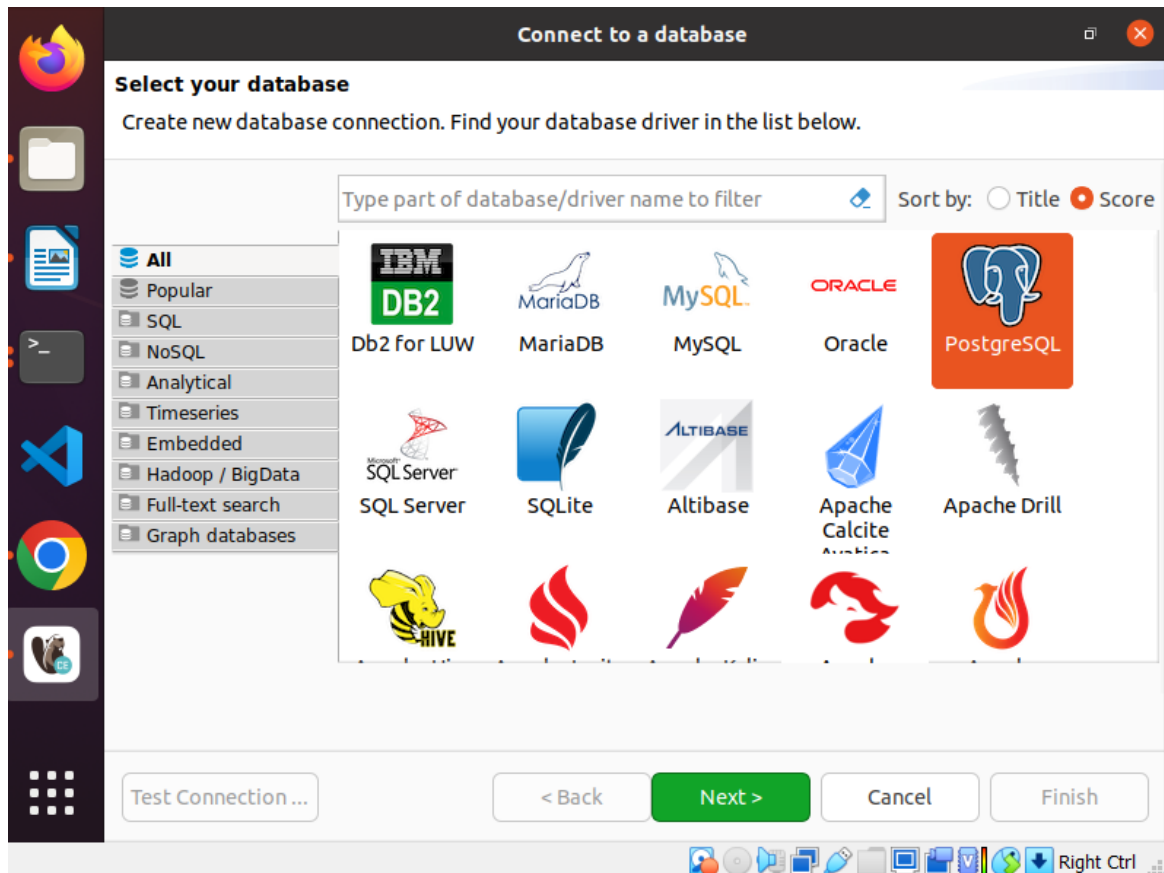


Устанавливаем DBeaver. Через него подключимся к бд и напишем sql запрос к данным.

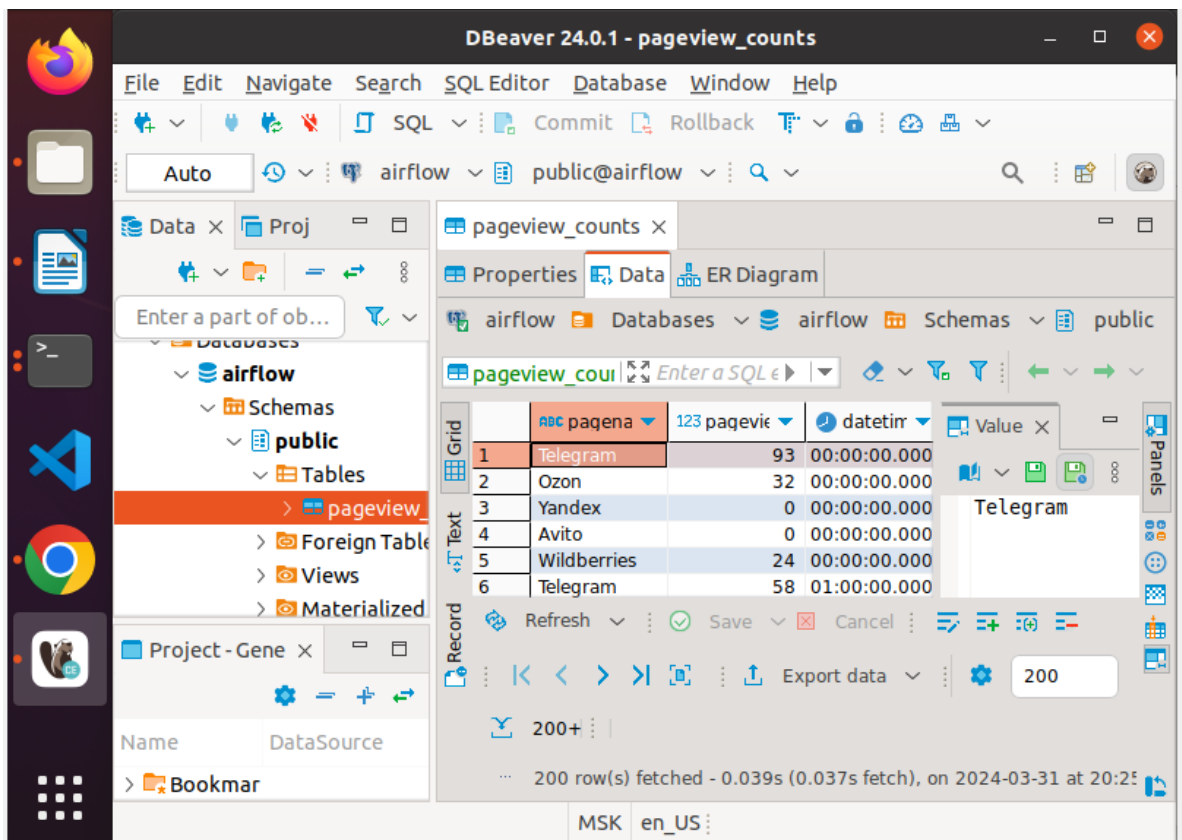
```
mgpu@mgpu-VirtualBox: ~/workshop-on-ETL/business_cas...
1 upgraded, 0 newly installed, 0 to remove and 76 not upgraded.
Need to get 24,4 MB of archives.
After this operation, 70,1 MB disk space will be freed.
Get:1 http://ru.archive.ubuntu.com/ubuntu focal-updates/main amd64 snapd amd64
2.61.3+20.04 [24,4 MB]
Fetched 24,4 MB in 3s (8 341 kB/s)
(Reading database ... 181347 files and directories currently installed.)
Preparing to unpack .../snapd_2.61.3+20.04_amd64.deb ...
Unpacking snapd (2.61.3+20.04) over (2.58+20.04.1) ...
Setting up snapd (2.61.3+20.04) ...
Installing new version of config file /etc/apparmor.d/usr.lib.snapd.snap-confi
e.real ...
snapd.failure.service is a disabled or a static unit not running, not starting
it.
snapd.snap-repair.service is a disabled or a static unit not running, not start
ing it.
Failed to restart snapd.mounts-pre.target: Operation refused, unit snapd.mounts
-pre.target may be requested by dependency only (it is configured to refuse man
ual start/stop).
See system logs and 'systemctl status snapd.mounts-pre.target' for details.
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for dbus (1.12.16-2ubuntu2.3) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_stocksense$ sudo snap inst
all dbeaver-ce
dbeaver-ce 24.0.1.202403241413 from DBeaver (dbeaver-corp) installed
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_stocksense$
```



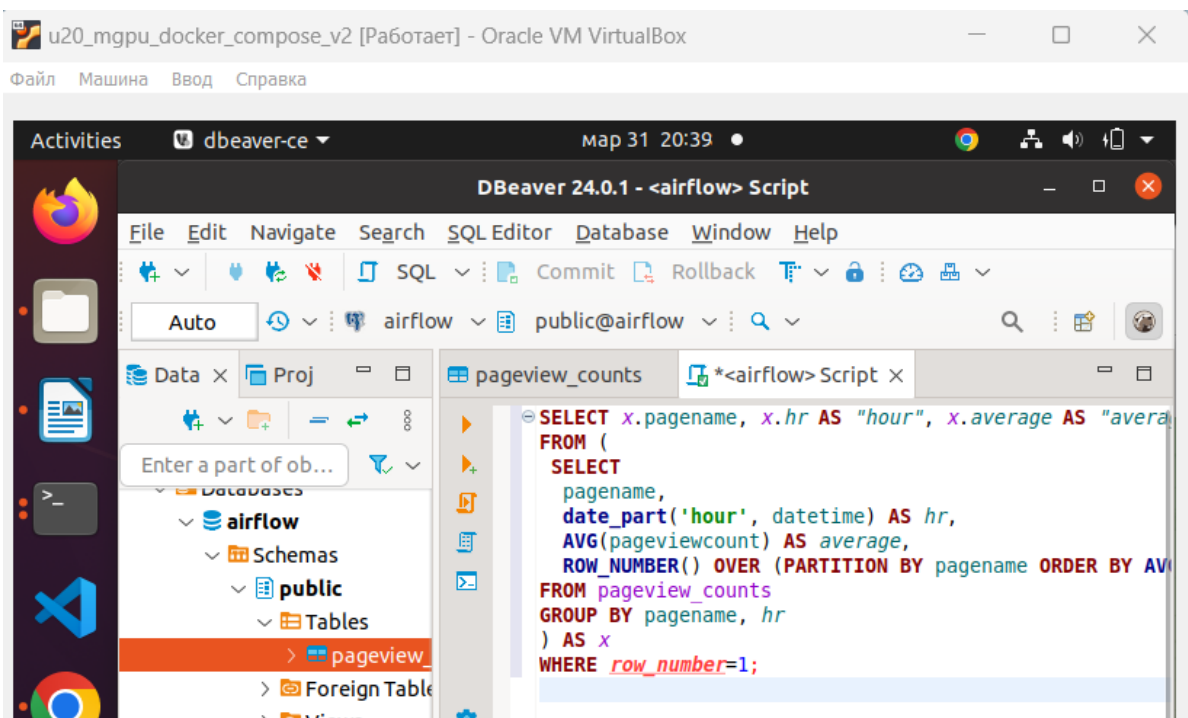
Оно появляется в каталоге приложений. Открываем его. Добавляем новое подключение Postgre.



Разворачиваем таблицу с данными о посещениях страниц компаний, добавленных в файл.



Пишем запрос в терминале



Выполняем запрос

DBeaver 24.0.1 - <airflow> Script

File Edit Navigate Search SQL Editor Database Window Help

Auto airflow public@airflow

pageview\_counts \*<airflow> Script

WHERE row\_number=1;

pageview\_counts 1

SELECT x.pageid

	ABC pagena	123 hour	123 average
1	Avito	9	0.5
2	Ozon	12	253
3	Telegram	12	412.5
4	Wildberries	11	202
5	Yandex	12	1

Avito

Refresh Save Cancel Export data 200

5 5 row(s) fetched - 0.005s, on 2024-03-31 at 20:43:53

MSK en\_US Writable S

## Верхнеуровневая архитектура аналитического решения задания Бизнес-кейса «StockSense» в draw.io.

