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

Лабораторная работа 6.1

Разработка полного ETL-процесса. Оркестровка конвейера данных

Выполнила: st_105

Москва

2025

Задачи:

— Запустить контейнер с Бизнес-кейсом «StockSense», изучить основные элементы DAG в Apache Airflow.

— Создать DAG согласно алгоритму, который предоставит преподаватель.

— Спроектировать верхнеуровневую архитектуру аналитического решения Бизнес-кейса «StockSense» в draw.io.

— Спроектировать архитектуру DAG Бизнес-кейса «StockSense» в draw.io.

ХОД РАБОТЫ

1. Клонирование репозитория

```
mgpu@mgpu-VirtualBox:~/Downloads$ git clone https://github.com/BosenkoTM/worksho
p-on-ETL.git
Cloning into 'workshop-on-ETL'...
remote: Enumerating objects: 675, done.
remote: Counting objects: 100% (68/68), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 675 (delta 22), reused 1 (delta 1), pack-reused 607 (from 1)
Receiving objects: 100% (675/675), 5.84 MiB | 5.25 MiB/s, done.
Resolving deltas: 100% (320/320), done.
mgpu@mgpu-VirtualBox:~/Downloads$
```

2. Проверка и загрузка дампа с Wikimedia за 5 апреля 2025 года

```
mgpu@mgpu-VirtualBox: ~/Downloads
mgpu@mgpu-VirtualBox:~$ git clone https://github.com/BosenkoTM/workshop-on-ETL.git
Cloning into 'workshop-on-ETL'...
remote: Enumerating objects: 675, done.
remote: Counting objects: 100% (68/68), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 675 (delta 22), reused 1 (delta 1), pack-reused 607 (from 1)
Receiving objects: 100% (675/675), 5.84 MiB | 4.66 MiB/s, done.
Resolving deltas: 100% (320/320), done.
mgpu@mgpu-VirtualBox:~$ cd Downloads
mgpu@mgpu-VirtualBox:~/Downloads$ wget https://dumps.wikimedia.org/other/pageviews/2025/2023-04/pageviews-20250405-060000.gz
--2025-04-05 19:02:32--  https://dumps.wikimedia.org/other/pageviews/2025/2023-04/pageviews-20250405-060000.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... 404 Not Found
2025-04-05 19:02:33 ERROR 404: Not Found.

mgpu@mgpu-VirtualBox:~/Downloads$ wget https://dumps.wikimedia.org/other/pageviews/2025/2025-04/pageviews-20250405-060000.gz
--2025-04-05 19:04:35--  https://dumps.wikimedia.org/other/pageviews/2025/2025-04/pageviews-20250405-060000.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: 53769388 (51M) [application/octet-stream]
Saving to: 'pageviews-20250405-060000.gz'

pageviews-20250405- 100%[==================>]  51,28M  1,16MB/s    in 33s

2025-04-05 19:05:09 (1,57 MB/s) - 'pageviews-20250405-060000.gz' saved [53769388/53769388]

mgpu@mgpu-VirtualBox:~/Downloads$ gunzip pageviews-20250405-060000.gz
mgpu@mgpu-VirtualBox:~/Downloads$ awk -F ' ' '{print $1}' pageviews-20250405-060000 | sort | uniq -c | sort -nr | head
```

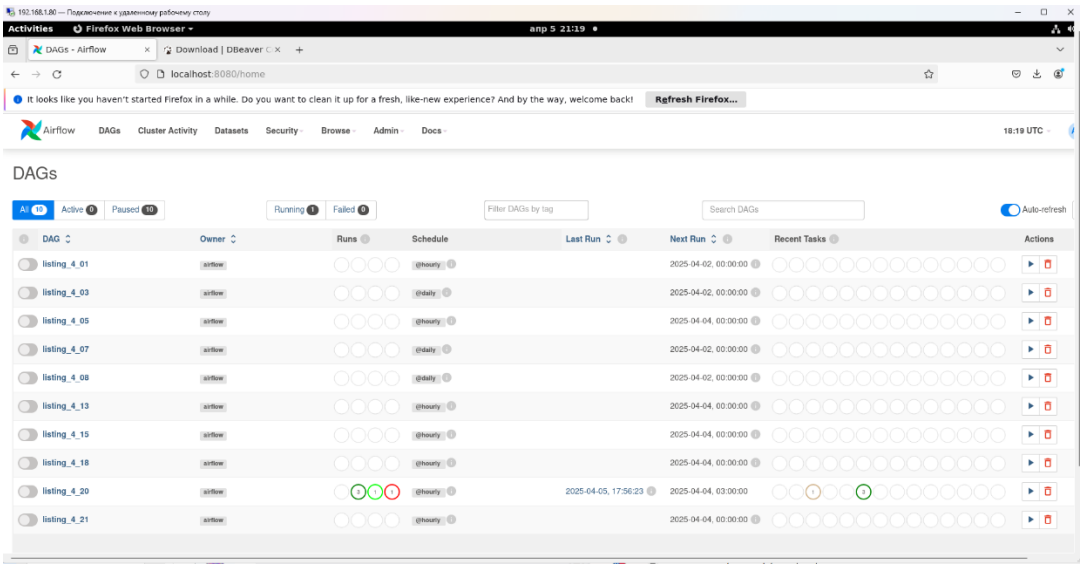
3. Получение популярных доменов за 5 апреля 2025 г.

```
mgpu@mgpu-VirtualBox:~/Downloads$ gunzip pageviews-20250405-060000.gz
mgpu@mgpu-VirtualBox:~/Downloads$ awk -F ' ' '{print $1}' pageviews-20250405-060000 | sort | uniq -c | sort -nr | head
1299751 en.m
1203023 en
 263955 ja.m
 247209 zh
 219490 ru.m
 207470 ja
 163791 es.m
 159906 de.m
 144869 zh.m
 117981 fr.m
mgpu@mgpu-VirtualBox:~/Downloads$
```

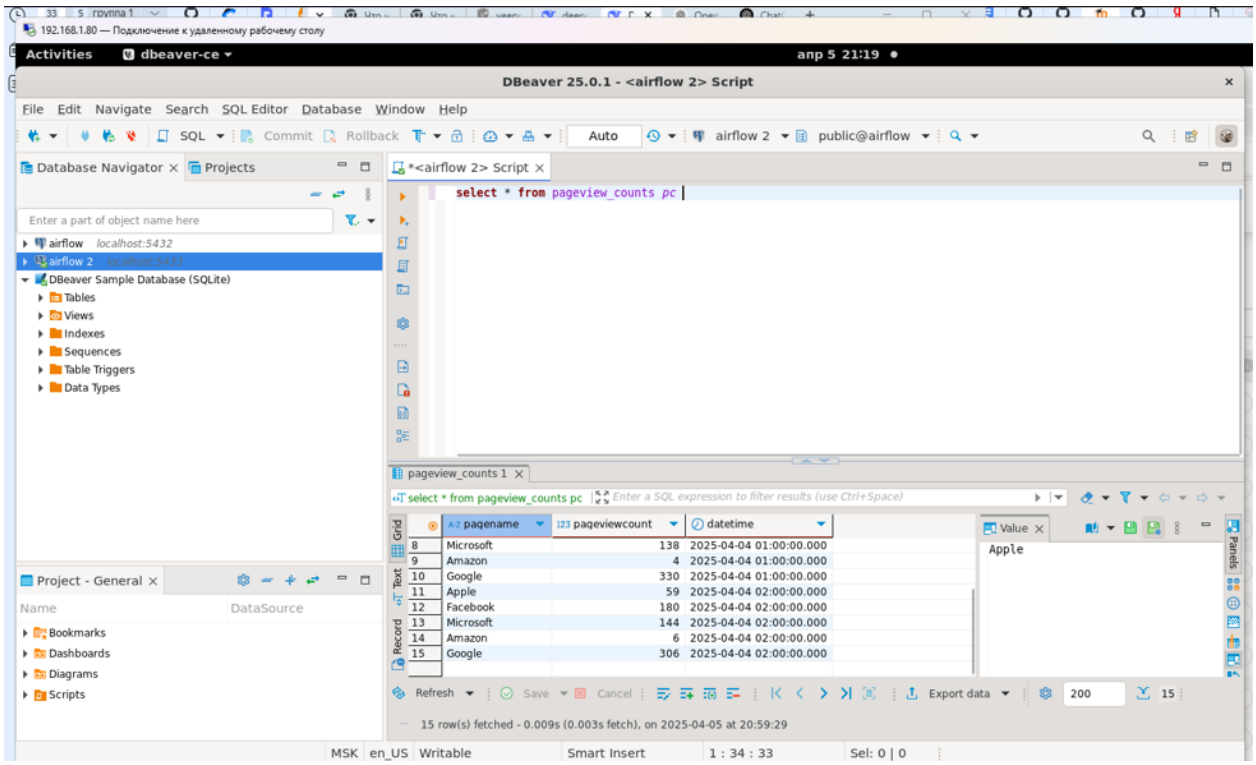
4. Запуск airflow

```
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_stocksense_25$ sudo docker compose down -v --remove-orphans
[sudo] password for mgpu:
[+] Running 8/8
 ✔ Container business_case_stocksense_25-scheduler-1      Removed                    1.3s
 ✔ Container business_case_stocksense_25-webserver-1      Removed                    1.3s
 ✔ Container business_case_stocksense_25-wiki_results-1   Removed                    0.0s
 ✔ Container business_case_stocksense_25-init-1           Removed                    0.0s
 ✔ Container business_case_stocksense_25-postgres-1       Removed                    0.0s
 ✔ Volume business_case_stocksense_25_postgres_data       Removed                    0.3s
 ✔ Volume business_case_stocksense_25_logs                Removed                   12.5s
 ✔ Network business_case_stocksense_25_default            Removed                    0.4s
mgpu@mgpu-VirtualBox:~/workshop-on-ETL/business_case_stocksense_25$ sudo docker compose up --build
[+] Running 8/8
 ✔ Network business_case_stocksense_25_default            Created                    0.3s
 ✔ Volume "business_case_stocksense_25_logs"              Created                    0.0s
```

5. Проверка работоспособности



6. Проверка загрузки данных в postgres



ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Вариант 15

— Получить данные за 2 месяца для сайта Apple

— Постройте график, который отображает топ-5 по числу просмотров за последний месяц на основе данных, полученных через Airflow.

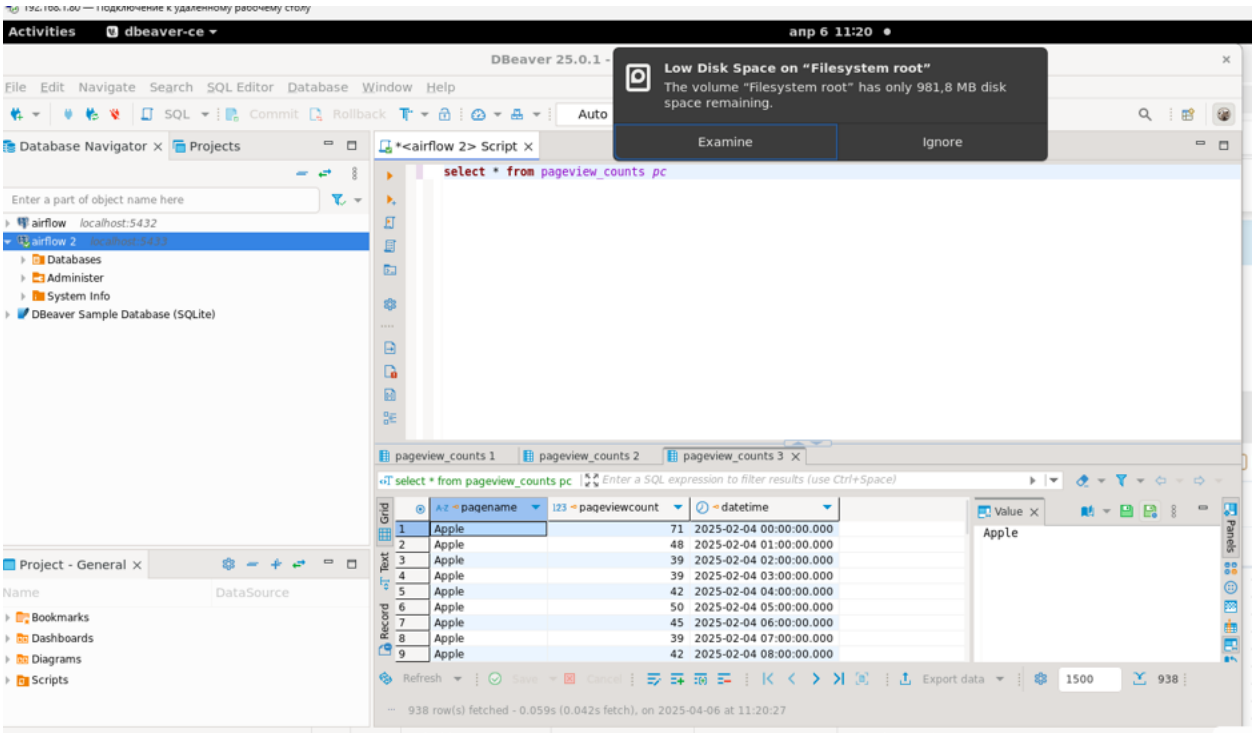
1. Меняем файл dag для загрузки данных с сайта Apple

   Листинг

```
from urllib import request
from datetime import datetime, timedelta

import airflow.utils.dates
from airflow import DAG
from airflow.operators.bash import BashOperator
from airflow.operators.python import PythonOperator
from airflow.providers.postgres.operators.postgres import PostgresOperator
from airflow.operators.dummy import DummyOperator

dag = DAG(
    dag_id="apple_pageviews_2_months",
    start_date=airflow.utils.dates.days_ago(60),  # Начало 2 месяца назад
    end_date=airflow.utils.dates.days_ago(0),     # Завершение сегодня
    schedule_interval="@hourly",
    template_searchpath="/tmp",
    max_active_runs=1,
    catchup=True,  # Важно для выполнения всех прошлых задач
)
```

```
def _get_data(execution_date, output_path):
    url = (
        "https://dumps.wikimedia.org/other/pageviews/"
        f"{execution_date.year}/{execution_date.year}-{execution_date.month:0>2}/"
        f"pageviews-{execution_date.year}{execution_date.month:0>2}"
        f"{execution_date.day:0>2}-{execution_date.hour:0>2}0000.gz"
    )
    request.urlretrieve(url, output_path)

get_data = PythonOperator(
    task_id="get_data",
    python_callable=_get_data,
    op_kwargs={
        "output_path": "/tmp/wikipageviews.gz",
    },
    dag=dag,
)

extract_gz = BashOperator(
    task_id="extract_gz",
    bash_command="gunzip --force /tmp/wikipageviews.gz",
    dag=dag
)

def _fetch_pageviews(execution_date):
    result = {"Apple": 0}  # Только Apple
    with open("/tmp/wikipageviews", "r") as f:
        for line in f:
            domain_code, page_title, view_counts, _ = line.split(" ")
            if domain_code == "en" and page_title == "Apple":
                result["Apple"] = view_counts

    with open("/tmp/postgres_query.sql", "w") as f:
        f.write(
            "INSERT INTO pageview_counts VALUES ("
            f"'Apple', {result['Apple']}, '{execution_date}'"
            ");\n"
        )

fetch_pageviews = PythonOperator(
    task_id="fetch_pageviews",
    python_callable=_fetch_pageviews,
    dag=dag,
)

write_to_postgres = PostgresOperator(
    task_id="write_to_postgres",
    postgres_conn_id="my_postgres",
    sql="postgres_query.sql",
    dag=dag,
)

get_data >> extract_gz >> fetch_pageviews >> write_to_postgres
```

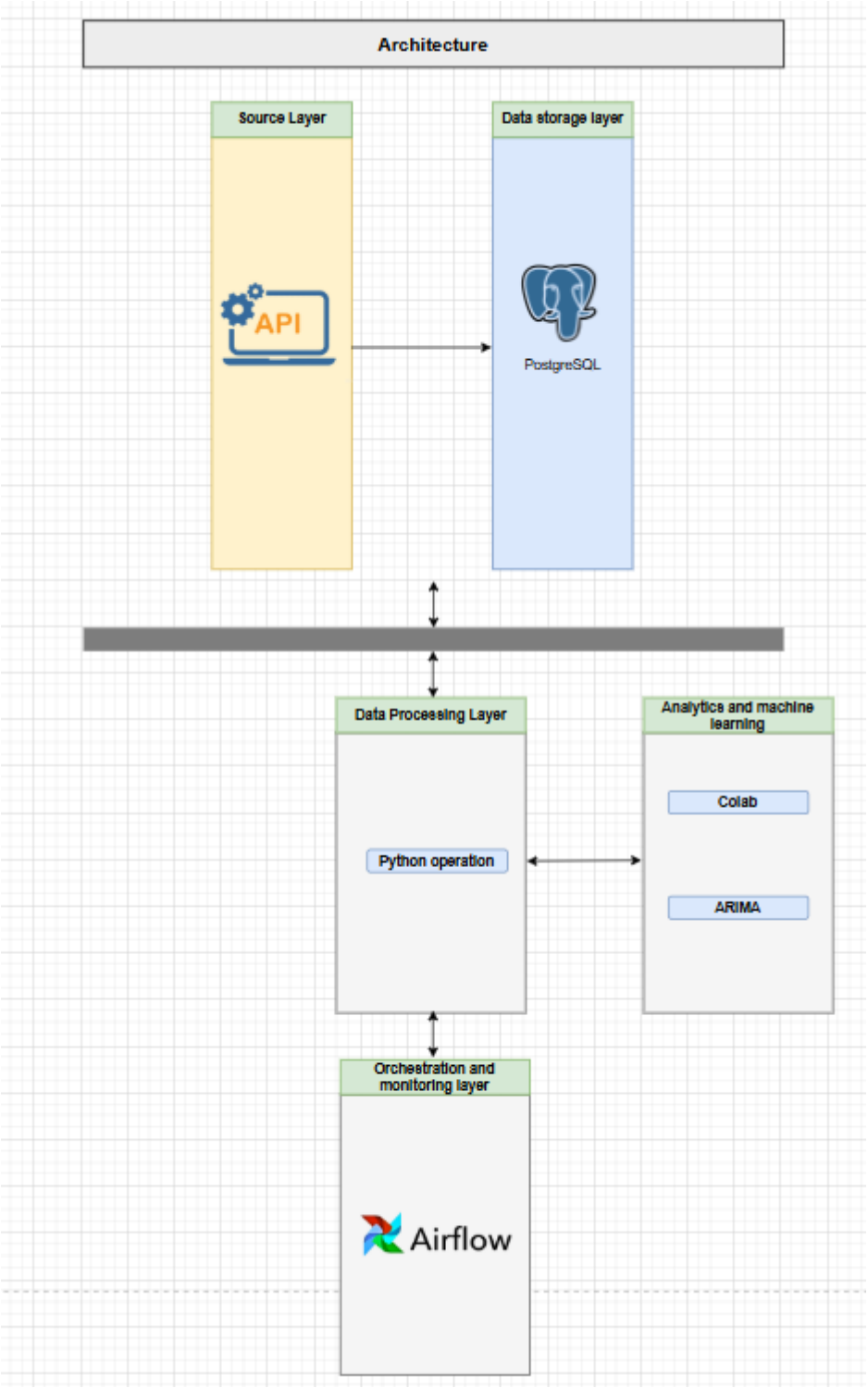
2. Проверка загрузки данных в postgres (загрузка 938 строк из 1440)



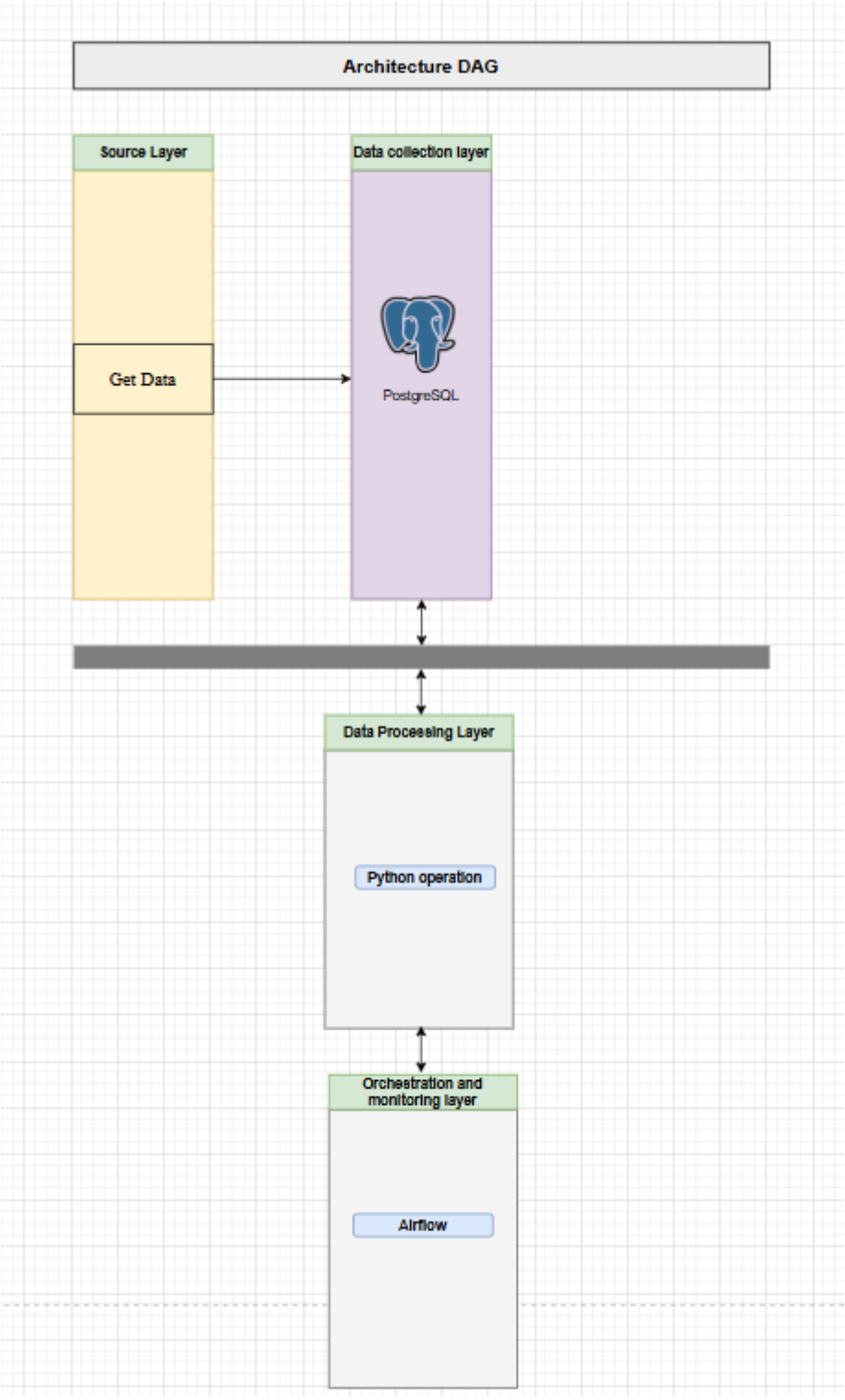
3. После выгрузки данных из postgres был проведен анализ файла csv

Анализ файла представлен в [google colab]

Верхнеуровневая архитектура



Архитектура DAG



Выводы по работе:

В ходе работы был изучен файл DAG и изменен для выполнения индивидуального задания, построен прогноз с помощью ARIMA