

Департамент образования 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 14.03

Выполнила:

Студентка группы АДЭУ-211

Кравцова Алёна Евгеньевна

Руководитель:

Босенко Т.М

Москва

2025

Шаги настройки и запуска контейнеров.

Просмотр запущенных контейнеров командой `docker ps` (Рис. 1).

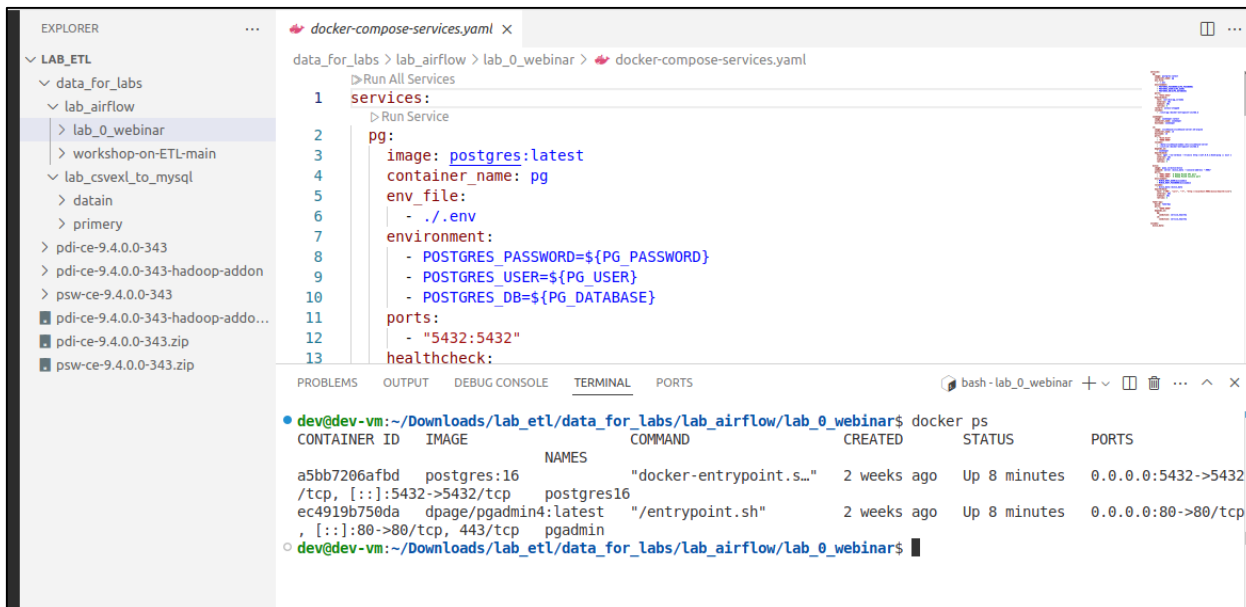


Рис. 1

Удалим все запущенные контейнеры (Рис. 2).

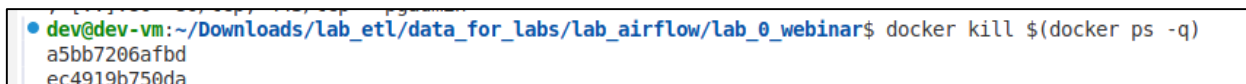


Рис. 2

Далее необходимо поднять все сервисы командой `make up-services` (Рис. 3).

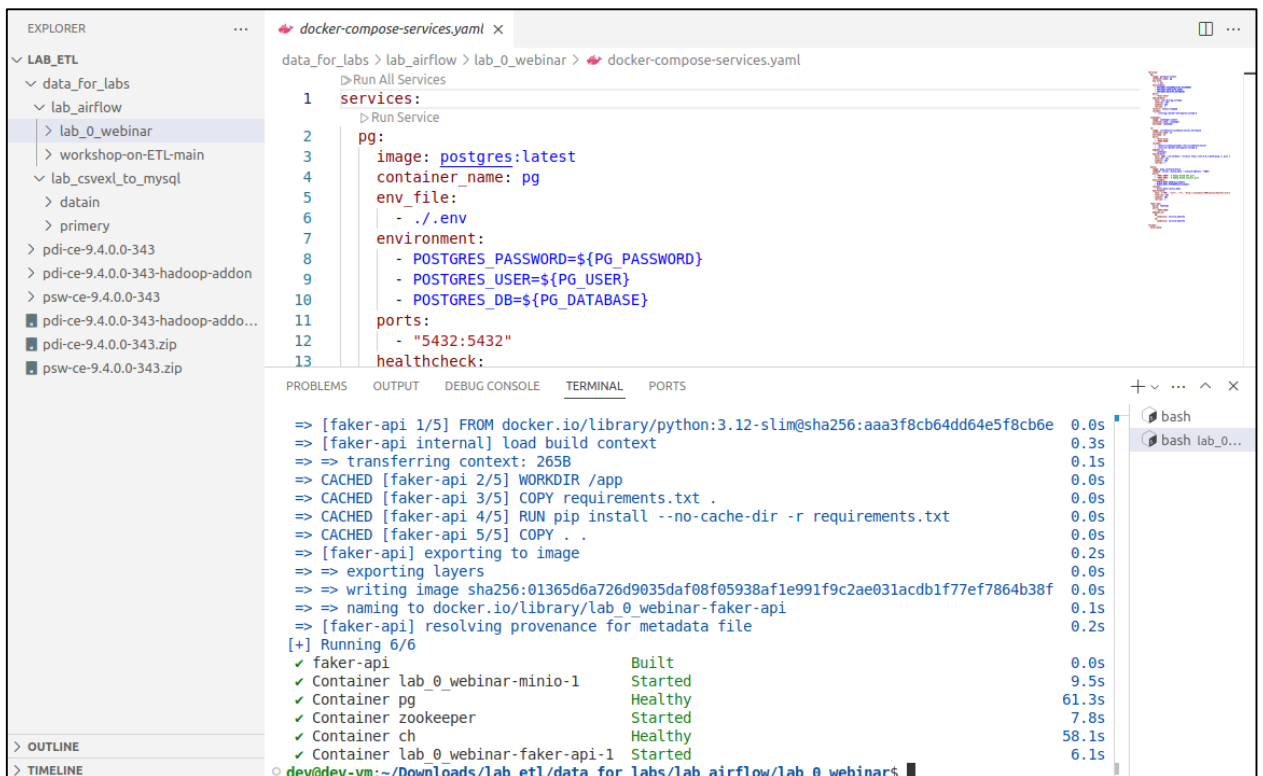


Рис. 3

Проверим работоспособность Fast Арі, все открывается (Рис. 4).

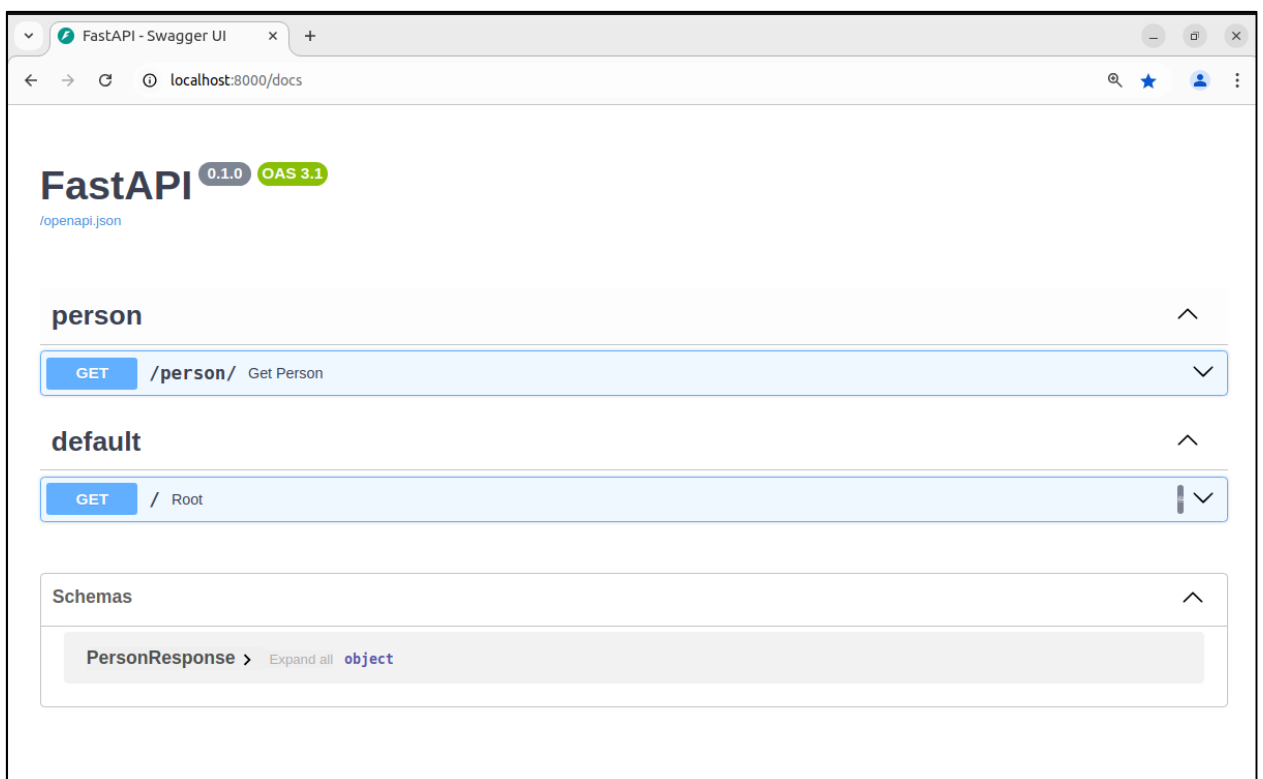


Рис. 4

Далее работоспособность MinIO, успешно (Рис. 5).

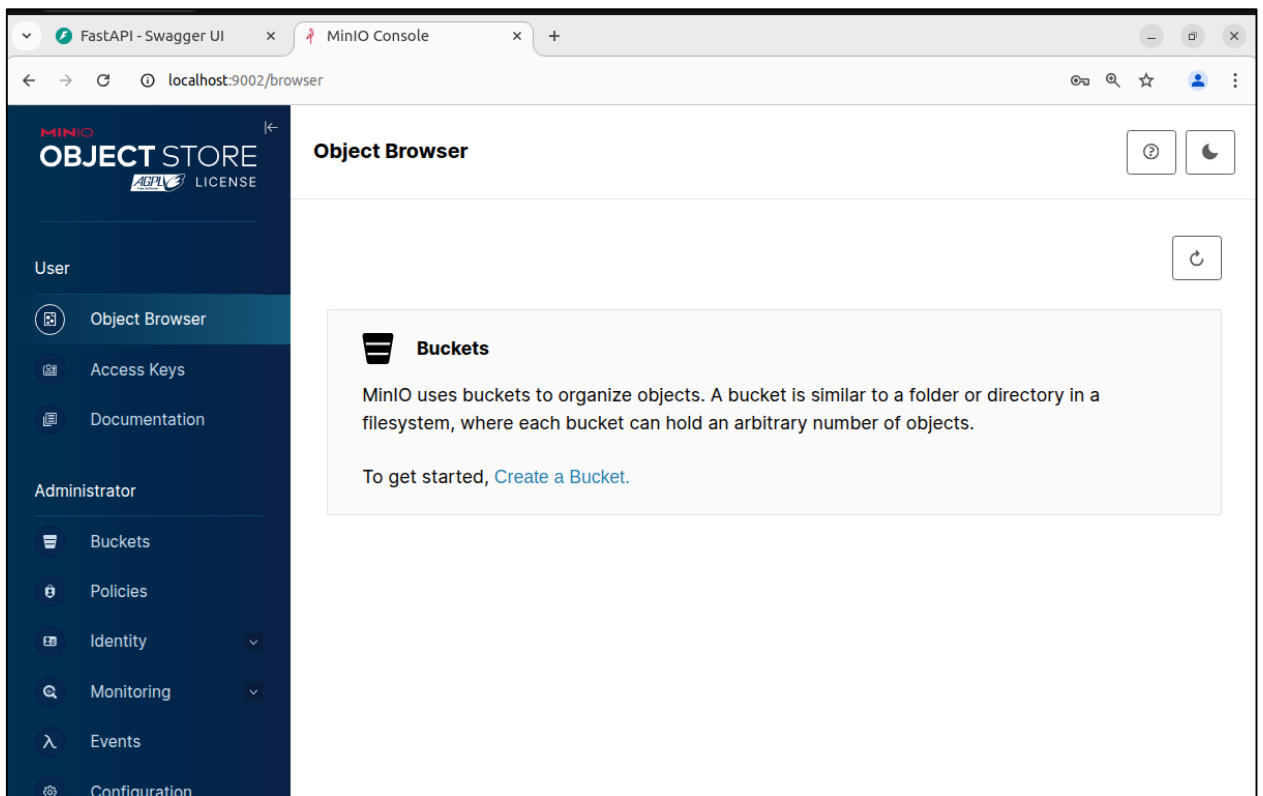


Рис. 5

Далее запустим airflow (Рис. 6).

```
if the --remove-orphans flag to clean it up.  
[+] Running 8/8  
✓ Volume "lab_0_webinar_postgres-db-volume" Crea...  
✓ Container lab_0_webinar-postgres-1 Healthy  
✓ Container lab_0_webinar-redis-1 Healthy  
✓ Container lab_0_webinar-airflow-init-1 Exited  
✓ Container lab_0_webinar-airflow-triggerer-1 St...  
✓ Container lab_0_webinar-airflow-webserver-1 St...  
✓ Container lab_0_webinar-airflow-scheduler-1 St...  
✓ Container lab_0_webinar-airflow-worker-1 Start...
```

Рис. 6

Проверим его работоспособность, успешно (Рис. 7).

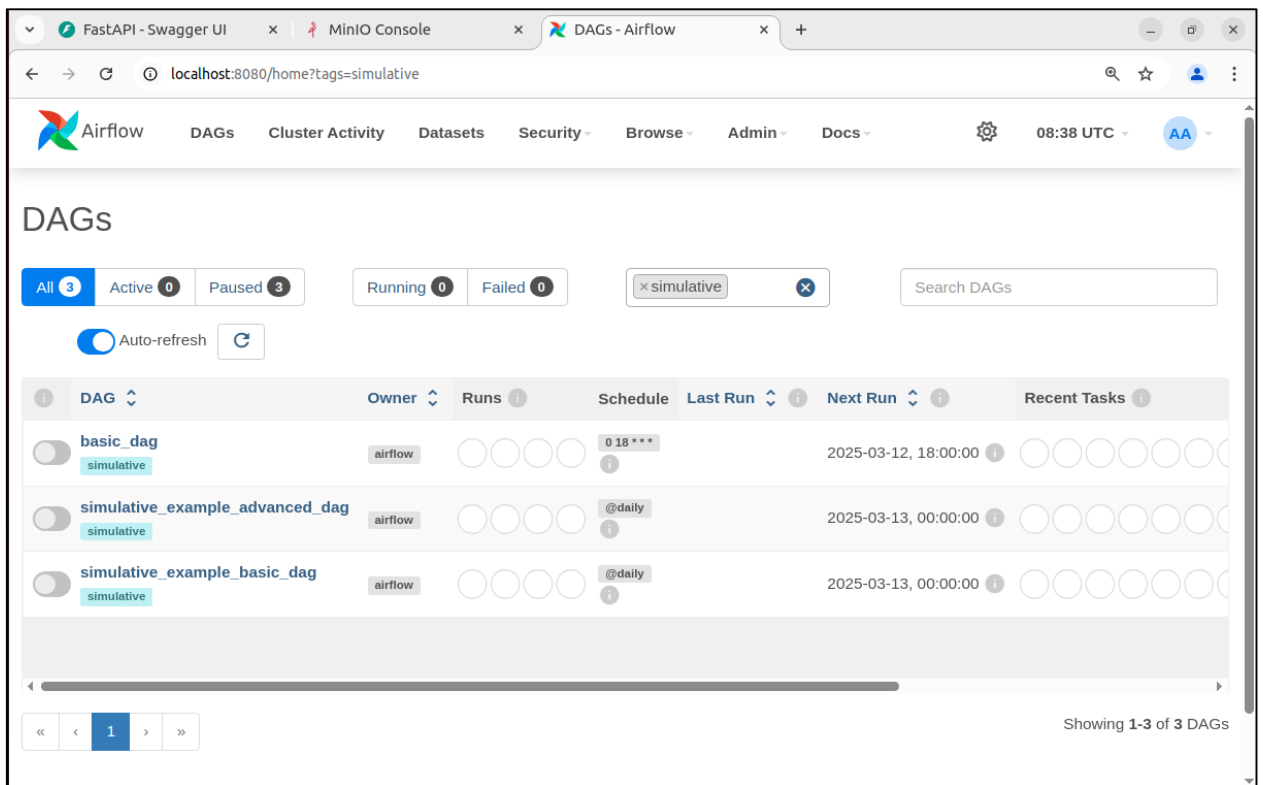


Рис. 7

Задание 1. Заполнение таблицы и анализ данных в PostgreSQL с визуализацией

1.1. Используйте библиотеку Faker для генерации фейковых данных. Вставьте сгенерированные данные в таблицу person в базе данных PostgreSQL.

Запустим даг с генерацией данных (Рис. 8, 9).

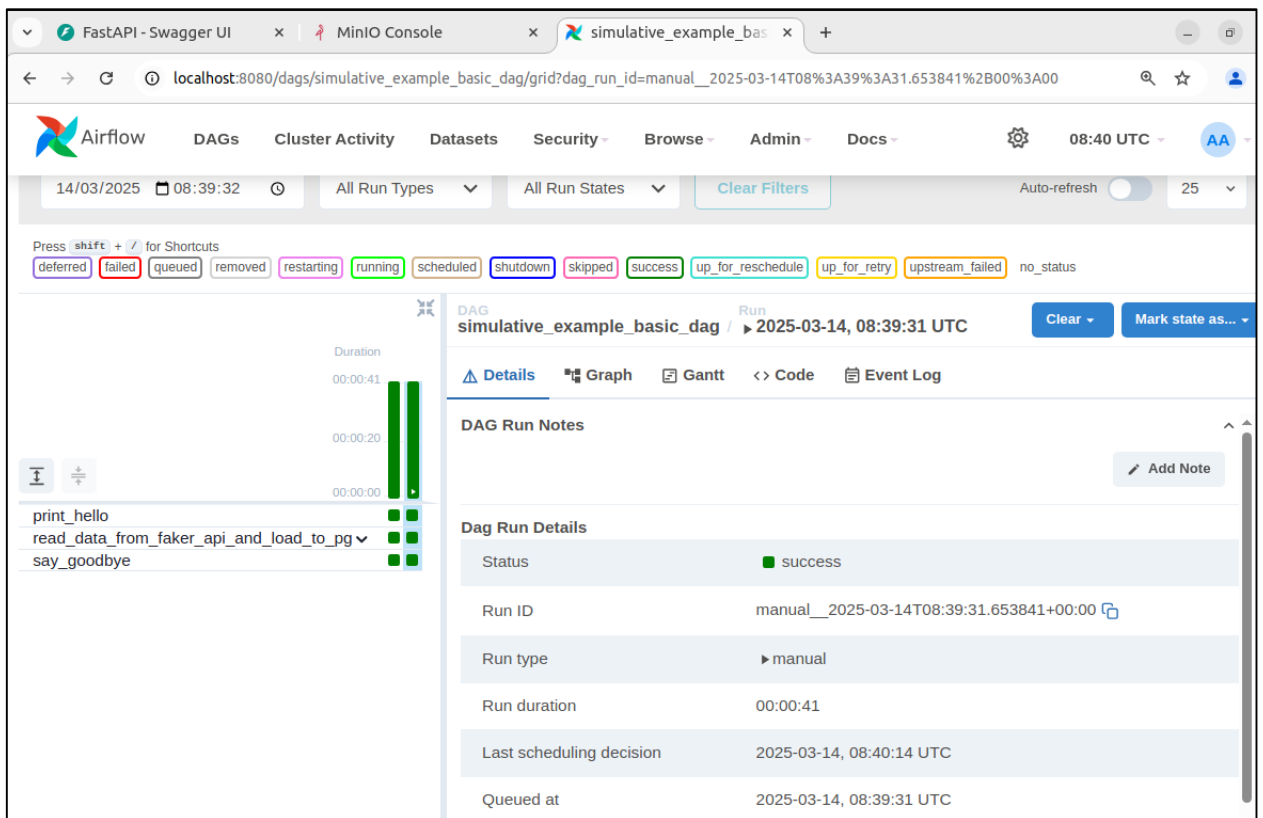


Рис. 8

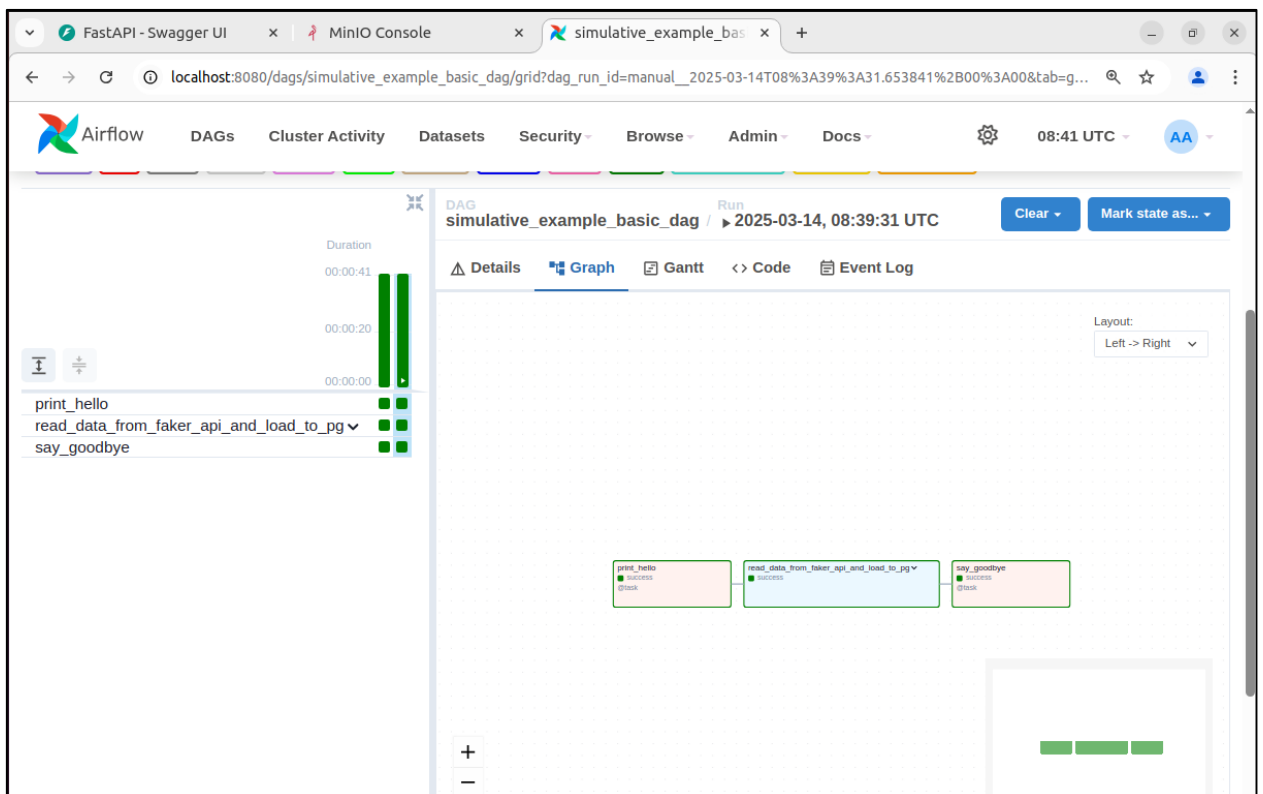


Рис. 9

После выполнения dag в Dbeaver появляются данные о клиентах (Рис. 10).

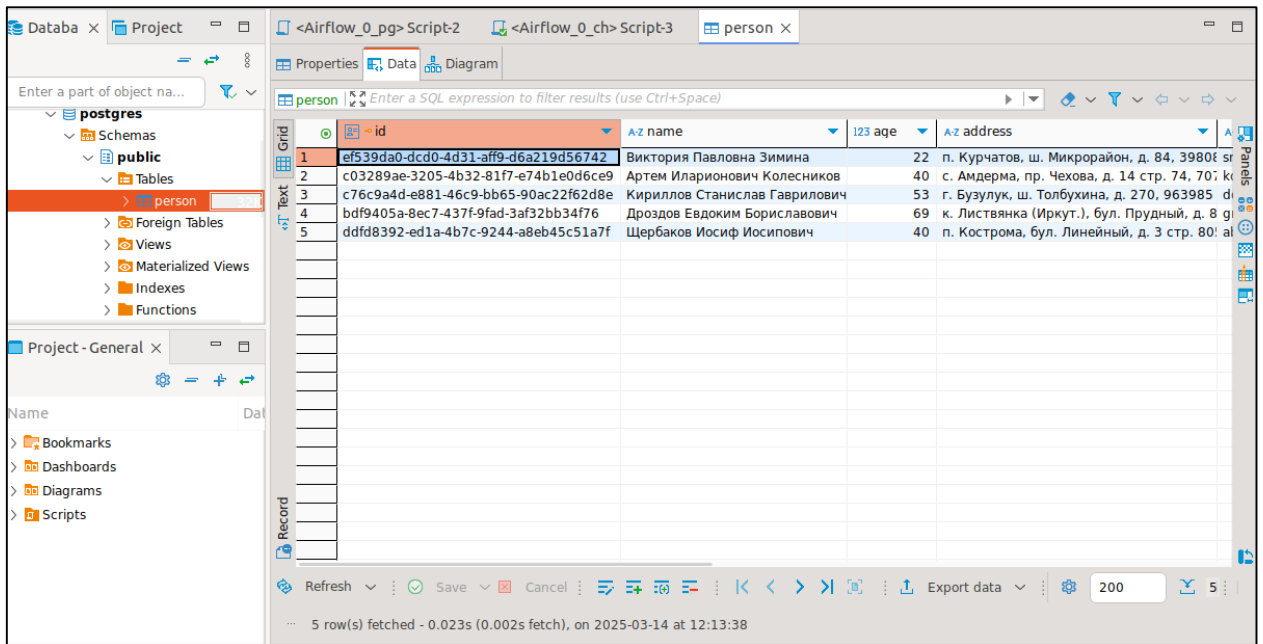


Рис. 10

Настроим dag для одновременного запуска нескольких задач (Рис. 11) и вызовом 100 задач, чтобы загрузить 100 записей в БД (Рис. 12).

```
with DAG(
    dag_id="simulative_example_basic_dag",
    schedule="@daily",
    start_date=datetime.datetime(2023, 1, 1), # установлено в прошлом
    catchup=False,
    tags=["simulative"],
    concurrency=150, # позволяет запустить много задач одновременно
    max_active_runs=1, # один запуск DAG в единицу времени
) as dag:
```

Рис. 11

```
# Запускаем 100 параллельных вызовов с помощью динамического маппинга
data_ids = read_data_from_faker_api.expand(dummy=list(range(100)))
load_results = load_data_to_pg.expand(data_id=data_ids)
return load_results
```

Рис. 12

После этого необходимо остановить и заново запустить airflow. Далее выполним dag, успешно (Рис. 13).

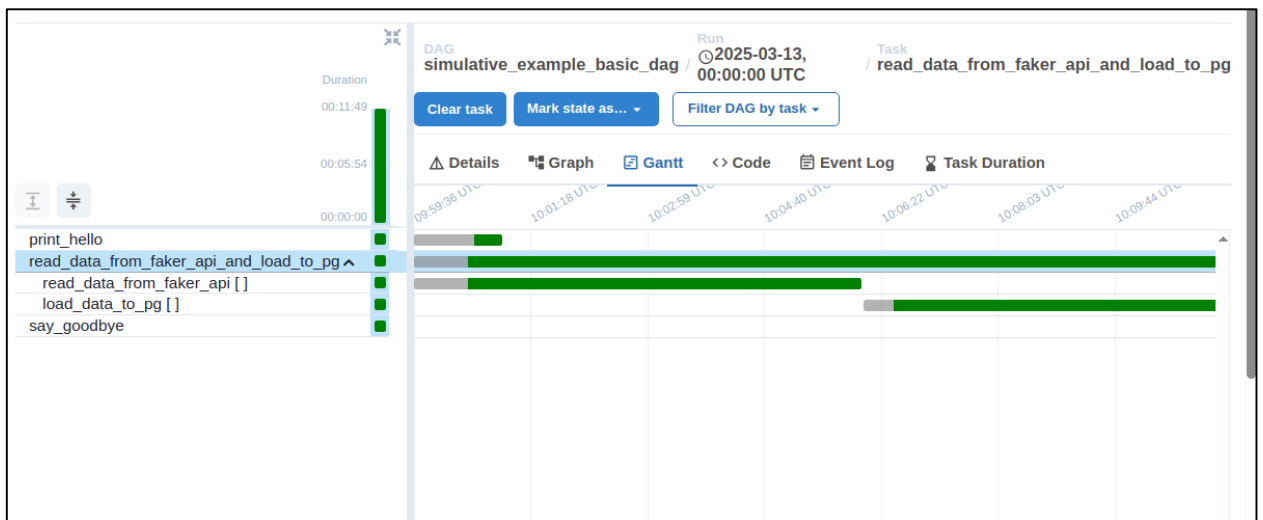


Рис. 13

Далее проверим в dbeaver, что загружено 100 записей, успешно (Рис. 14).

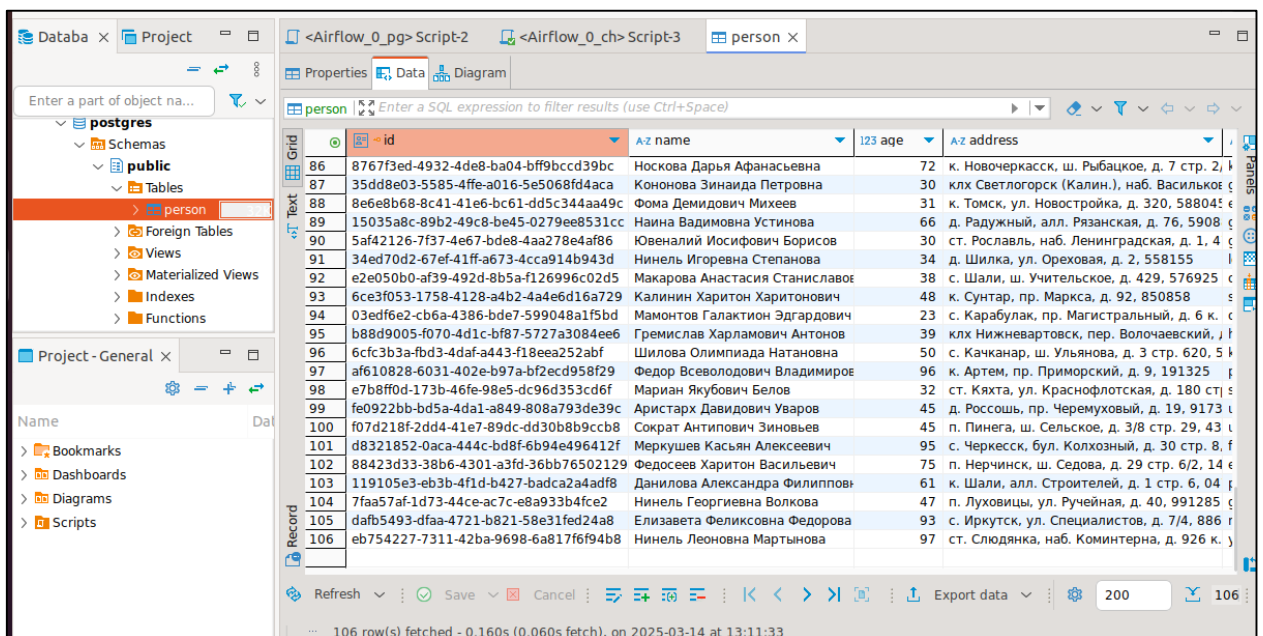


Рис. 14

1.2 Найдите средний, минимальный и максимальный возраст людей в таблице person.

Сделаем запрос в sql для получения информации о среднем возрасте. Итак, средний возраст – 57, самый юный – 18, самый старый – 99 (Рис. 15).

SQL Query:

```
SELECT
  AVG(age) AS average_age,
  MIN(age) AS minimum_age,
  MAX(age) AS maximum_age
FROM person
```

Results 1 X

SQL Filter: `SELECT AVG(age) AS average_age,` Enter a SQL expression to filter results (use Ctrl+Space)

	123 average age	123 minimum age	123 maximum age
1	57.9902912621	18	99

Рис. 15

1.3 Определите топ-5 городов, в которых проживает наибольшее количество людей. Таким образом, пункт «Артем» самый популярный (Рис. 16).

SQL Query:

```
SELECT
  SPLIT_PART(address, ',', 1) AS city,
  COUNT(*) AS population
FROM person
GROUP BY city
ORDER BY population DESC
```

Results 1 X

SQL Filter: `SELECT SPLIT_PART(address, ',', 1)` Enter a SQL expression to filter results (use Ctrl+Space)

	A-Z city	123 population
1	к. Артем	2
2	п. Курчатов	1
3	д. Северобайкальск	1
4	п. Кострома	1
5	ст. Уэлен	1

Рис. 16

1.4. Найдите количество регистраций в каждом месяце за последний год. Ставим интервал в год от текущего года. Итак, наибольшее количество регистраций в сентябре 2024 года (Рис. 17).

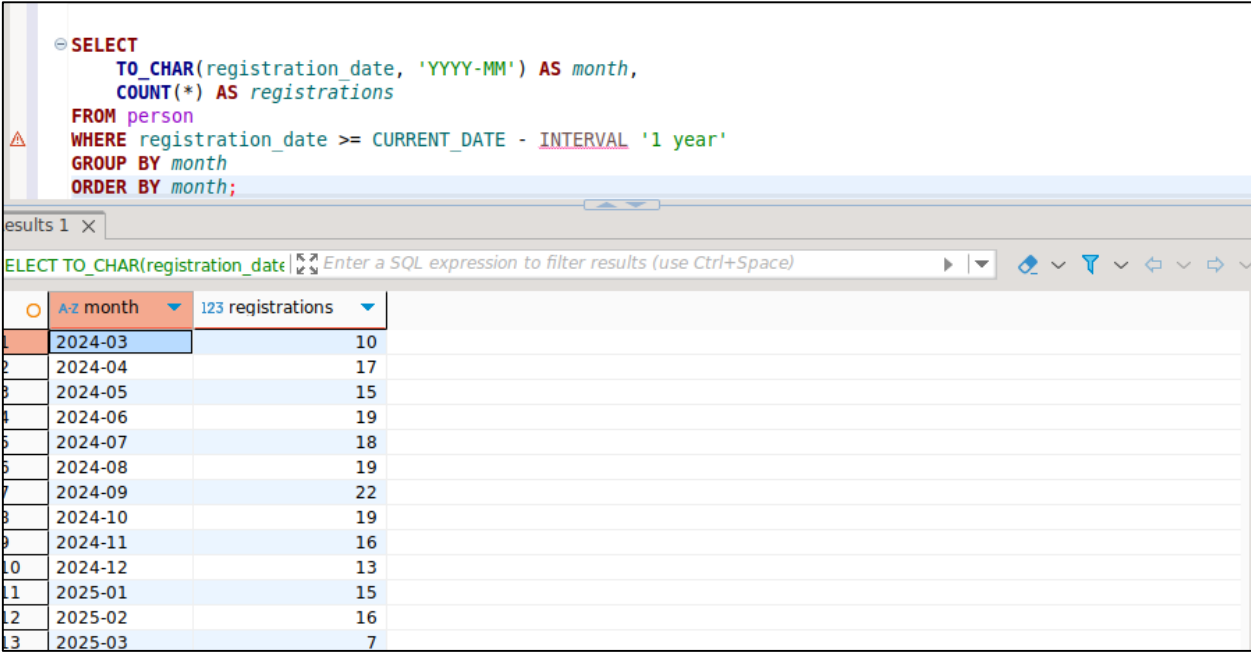


Рис. 17

1.5. Визуализация данных

Создайте графики для визуализации результатов анализа:

- Гистограмма распределения возраста.
- Диаграмма топ-5 городов по количеству проживающих.
- Линейный график количества регистраций по месяцам.

Первоначально необходимо установить необходимые библиотеки и загрузить датасет (Рис. 18).

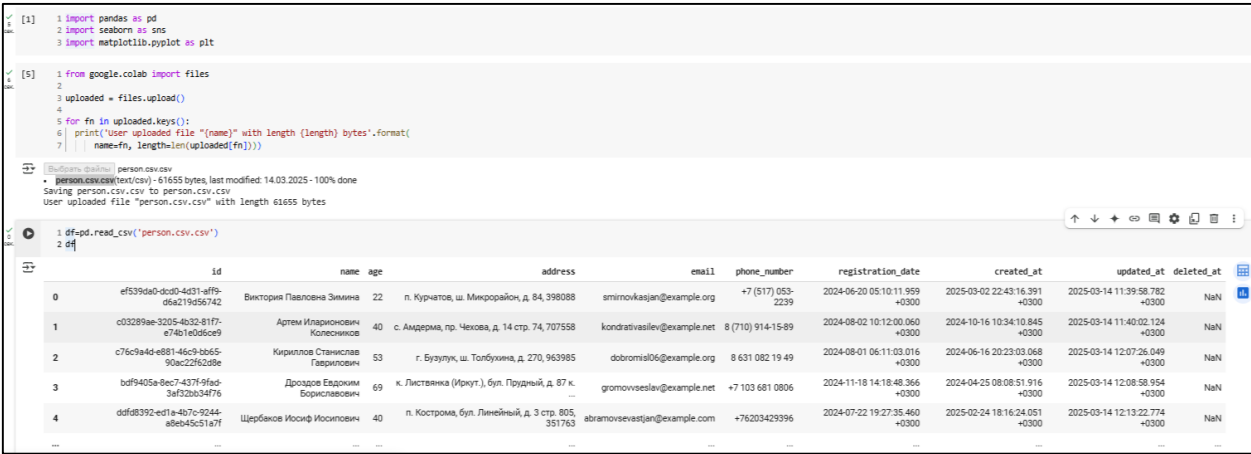


Рис. 18

Далее просмотрим общее описание данных (Рис. 19).

```
1 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 206 entries, 0 to 205
Data columns (total 10 columns):
#   Column              Non-Null Count  Dtype
---  -
0   id                   206 non-null   object
1   name                 206 non-null   object
2   age                  206 non-null   int64
3   address              206 non-null   object
4   email                206 non-null   object
5   phone_number         206 non-null   object
6   registration_date     206 non-null   object
7   created_at           206 non-null   object
8   updated_at           206 non-null   object
9   deleted_at           0 non-null      float64
dtypes: float64(1), int64(1), object(8)
memory usage: 16.2+ KB
```

Рис. 19

Видим, что необходимо перевести столбцы с датами в формат даты (Рис. 20).

```
[11] 1 df['registration_date'] = pd.to_datetime(df['registration_date'])
    2 df['created_at'] = pd.to_datetime(df['created_at'])
    3 df['updated_at'] = pd.to_datetime(df['updated_at'])

1 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 206 entries, 0 to 205
Data columns (total 10 columns):
#   Column              Non-Null Count  Dtype
---  -
0   id                   206 non-null   object
1   name                 206 non-null   object
2   age                  206 non-null   int64
3   address              206 non-null   object
4   email                206 non-null   object
5   phone_number         206 non-null   object
6   registration_date     206 non-null   datetime64[ns, UTC+03:00]
7   created_at           206 non-null   datetime64[ns, UTC+03:00]
8   updated_at           206 non-null   datetime64[ns, UTC+03:00]
9   deleted_at           0 non-null      float64
dtypes: datetime64[ns, UTC+03:00](3), float64(1), int64(1), object(5)
memory usage: 16.2+ KB
```

Рис. 20

Далее построим график распределения возрастов (Рис. 21).

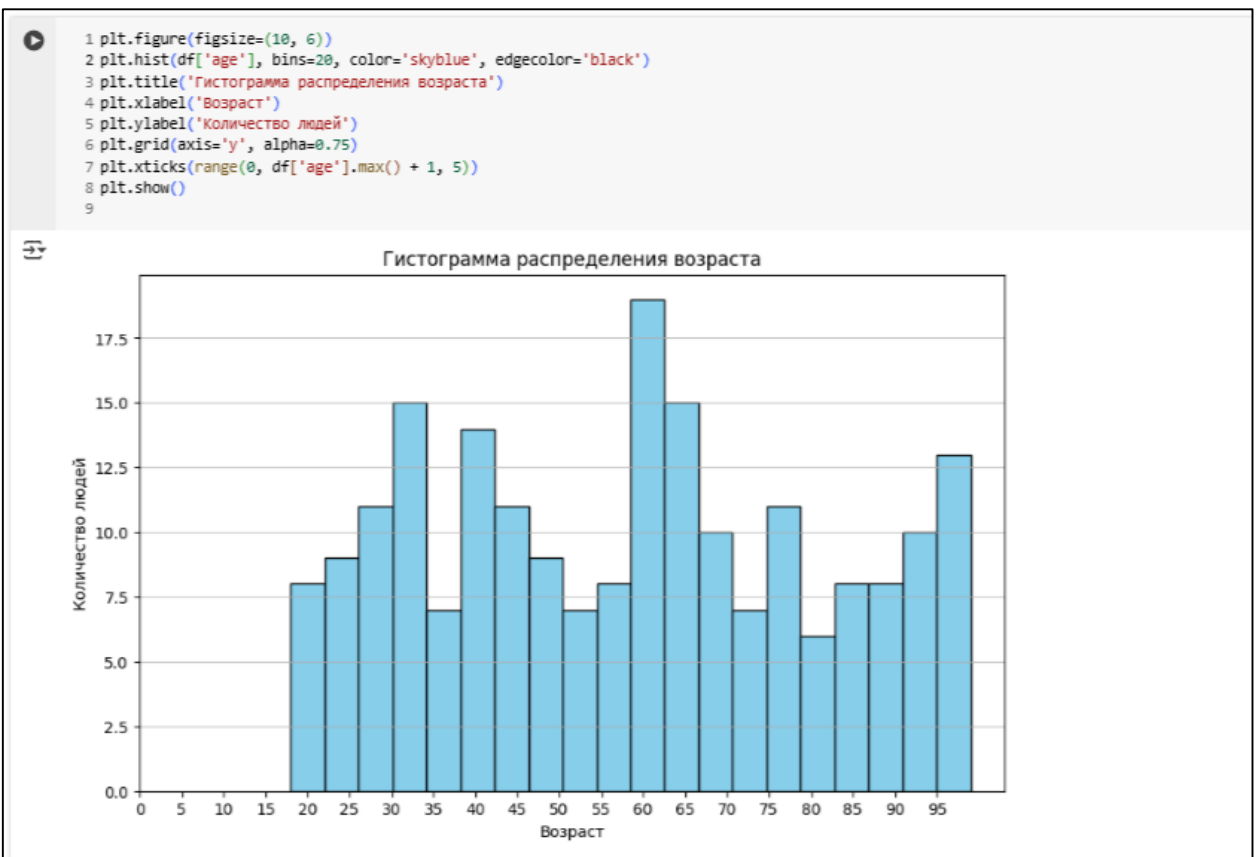
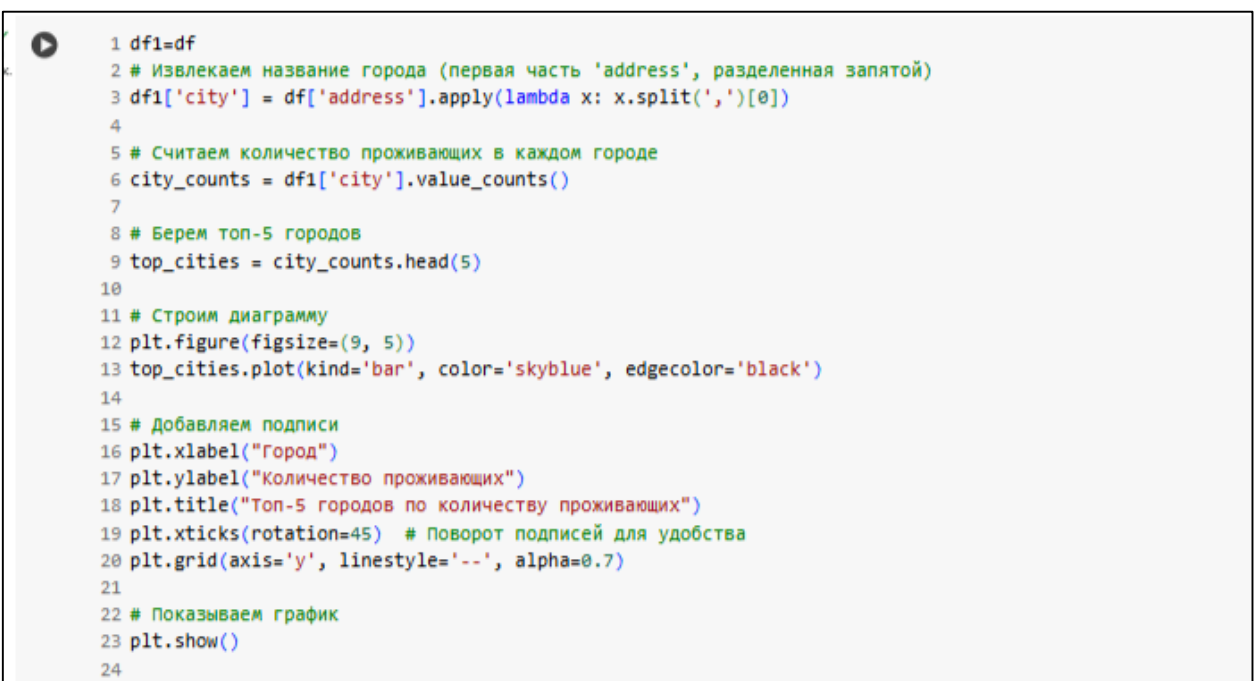


Рис. 21

Далее необходимо сделать гистограмму топ-5 городов (Рис. 22).



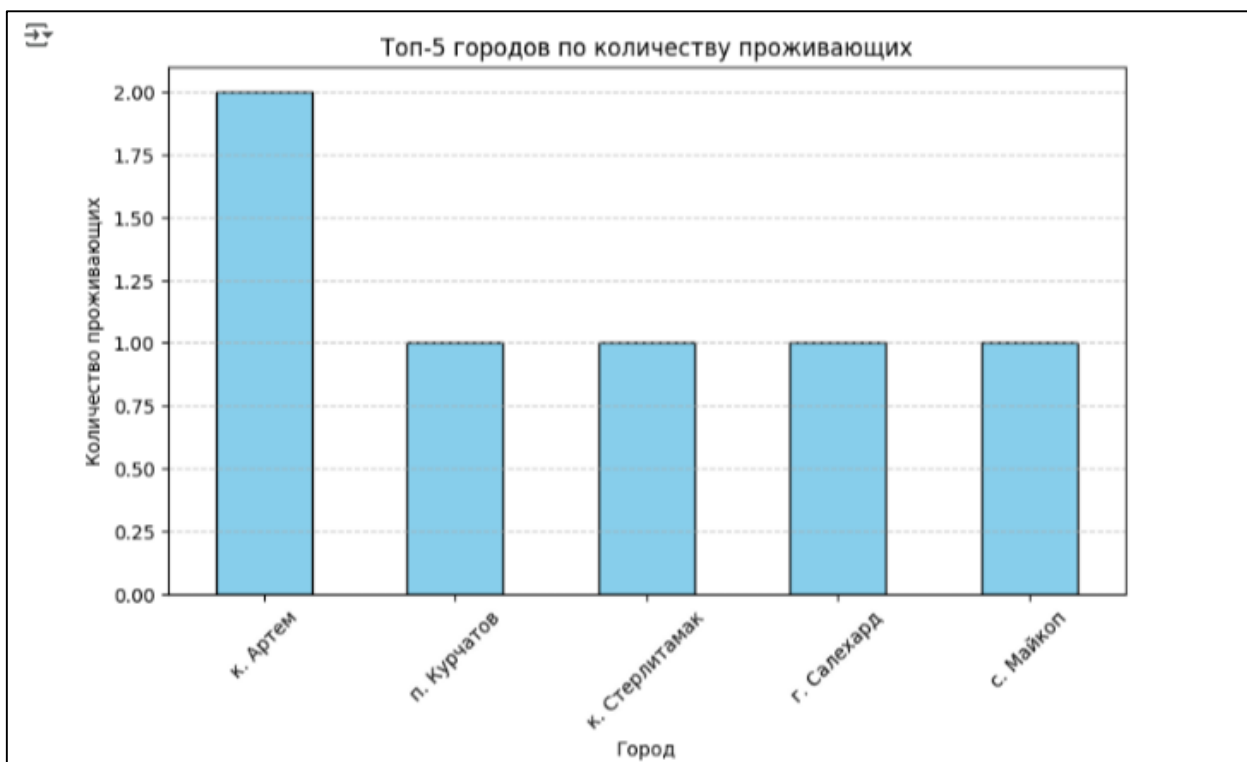


Рис. 22

Далее статистика по количеству регистраций за каждый месяц.

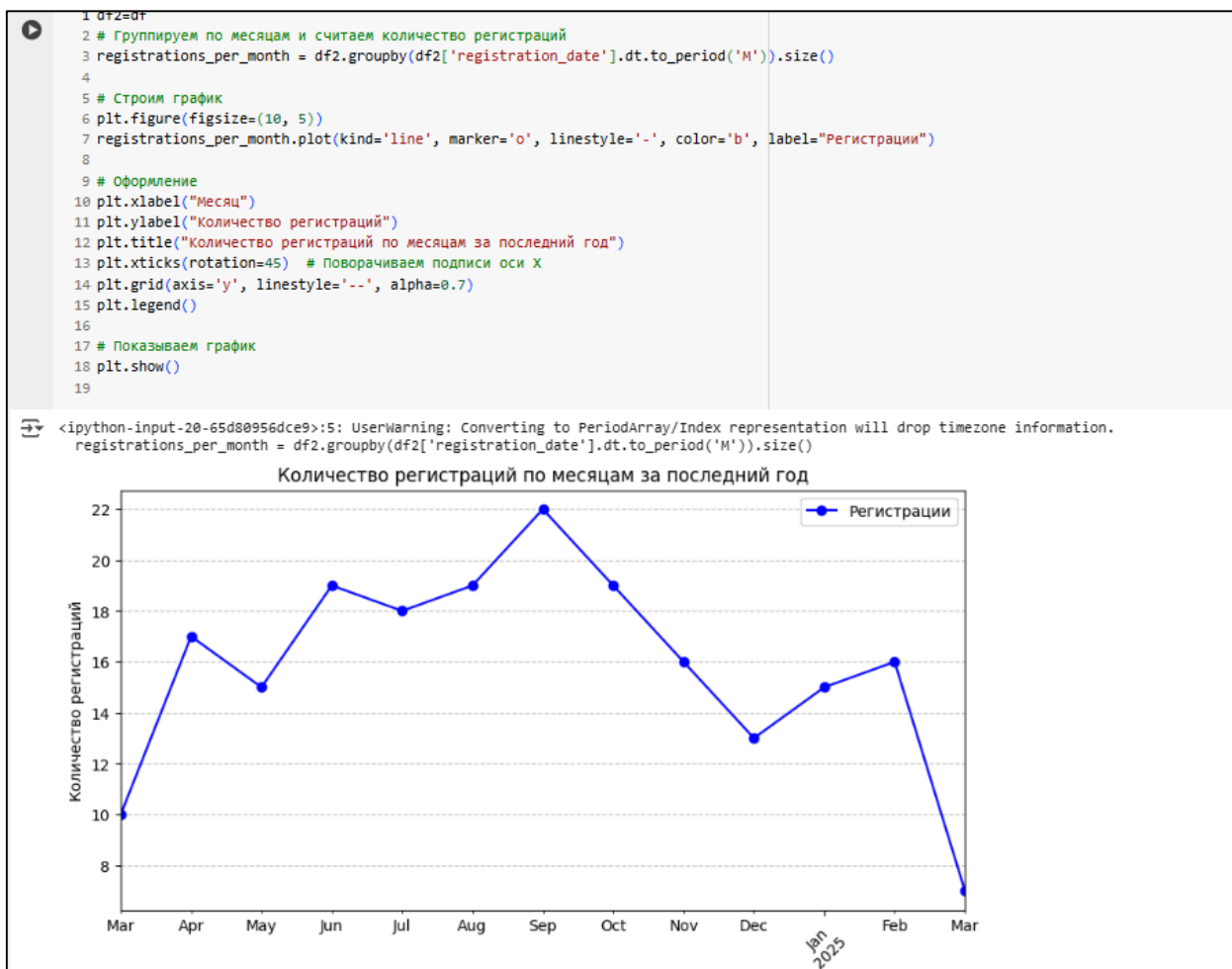


Рис. 23

Таким образом, все созданные визуализации соответствуют результатам запросов к БД.

Вывод по 1 заданию: были запущены все необходимые контейнеры, также выполнен и настроен dag на выполнение необходимого количества вызовов. Запросы к БД совпали с результатами визуализаций.