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).

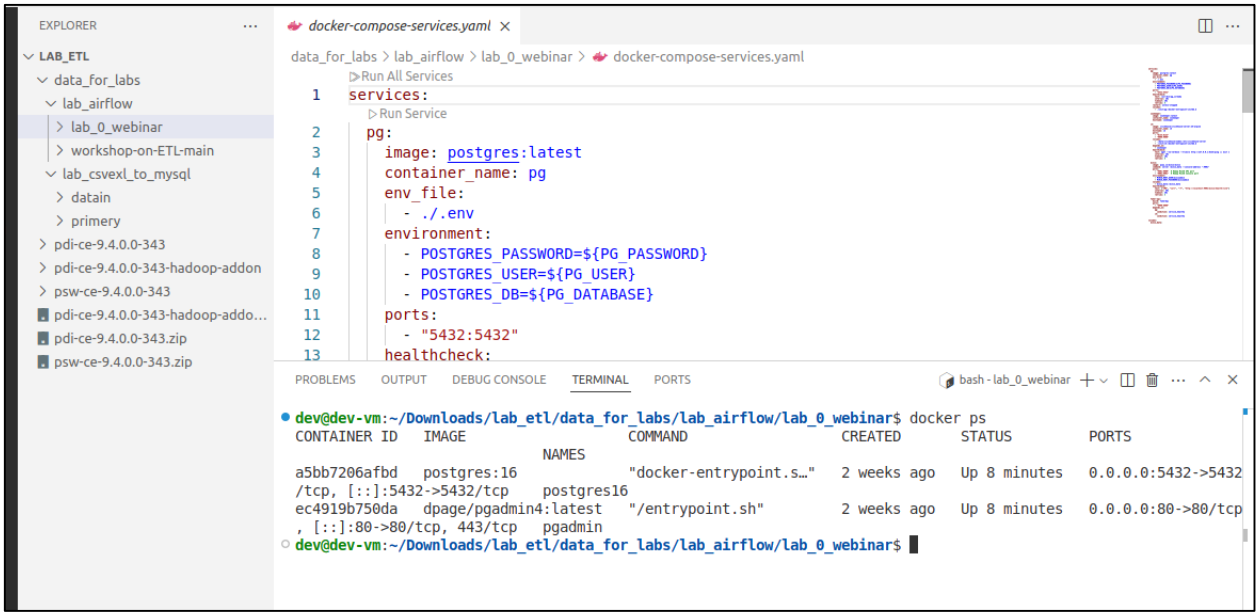


Рис. 1

Удалим все запущенные контейнеры (Рис. 2).

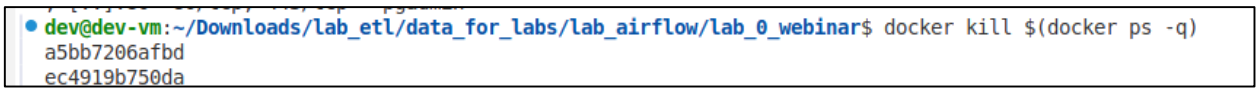


Рис. 2

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

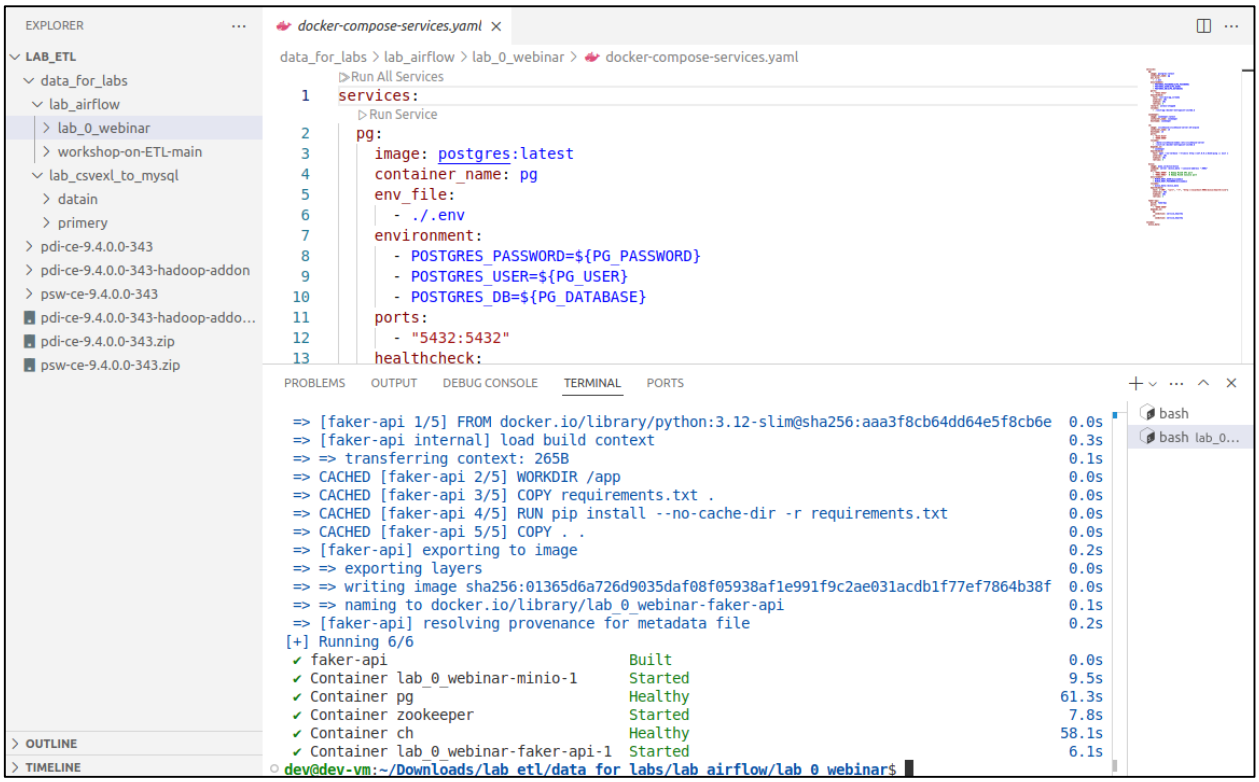


Рис. 3

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

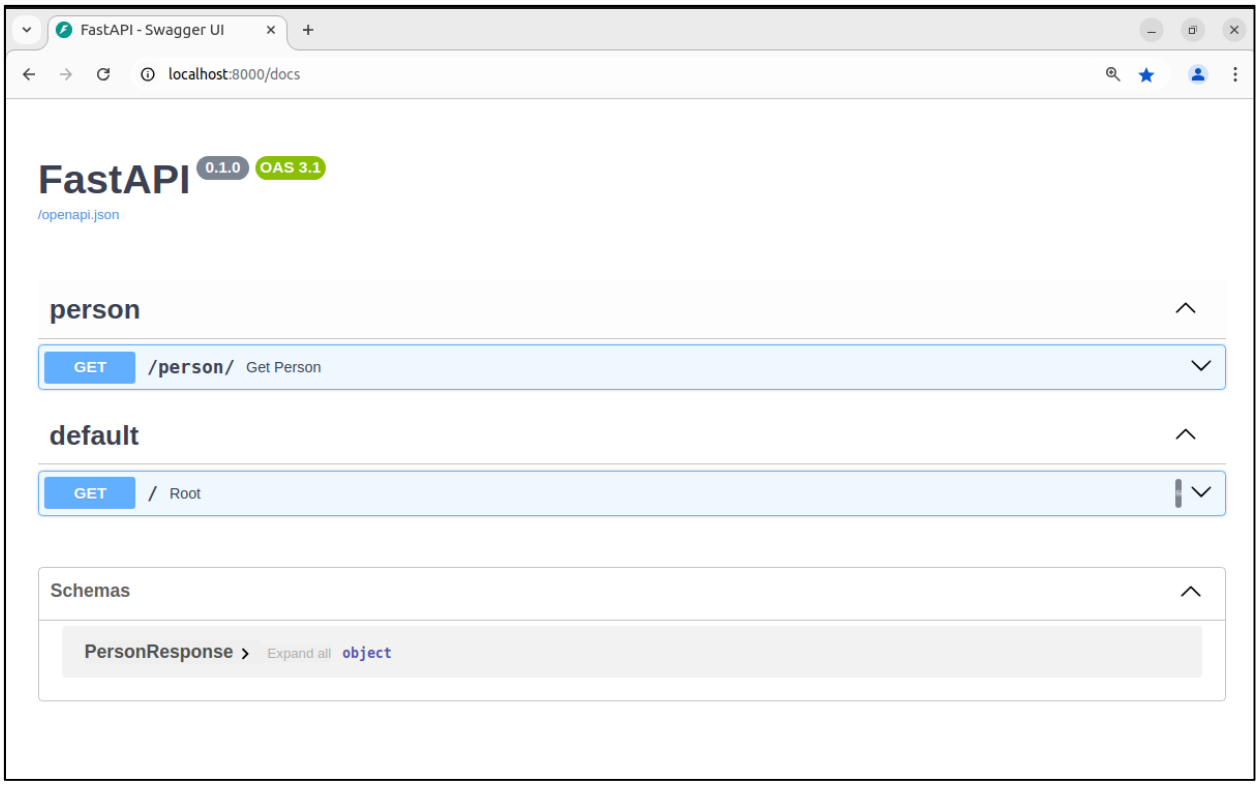


Рис. 4

Далее работоспособность MinIO, успешно (Рис. 5).

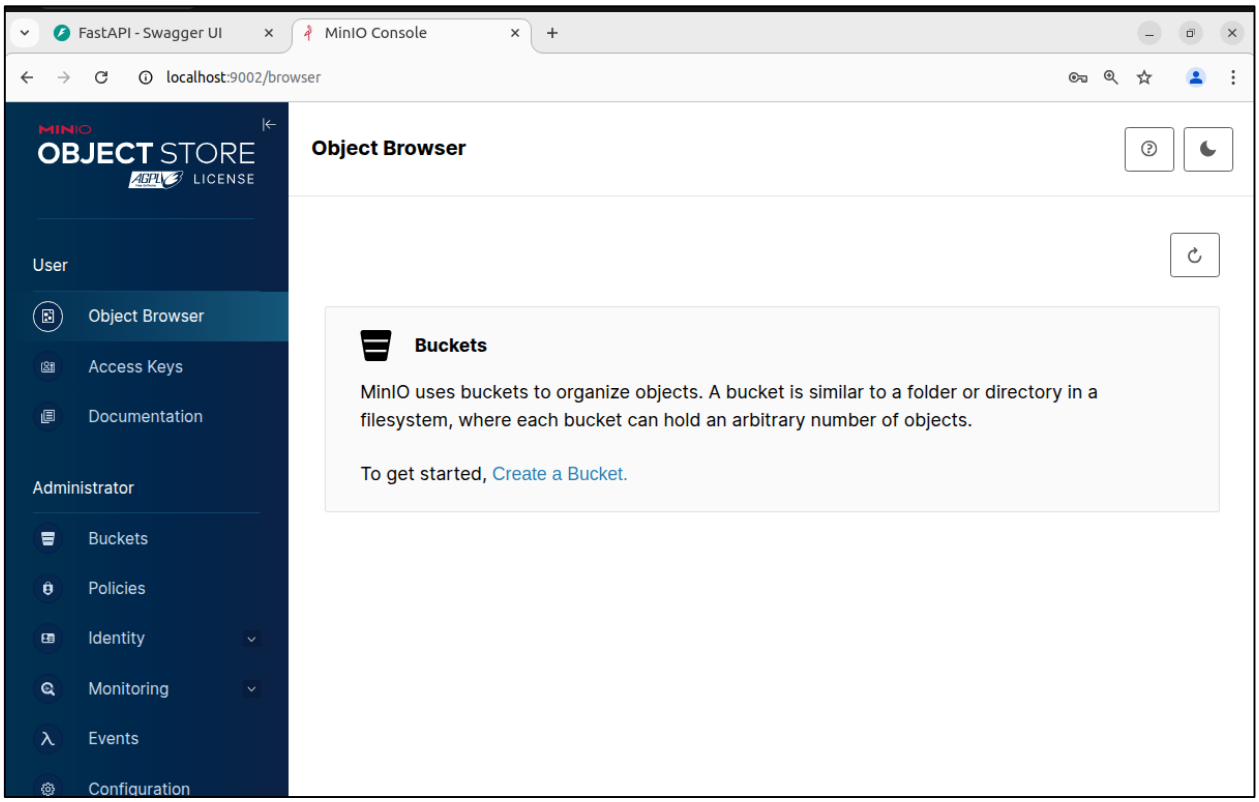


Рис. 5

Далее запустим airflow (Рис. 6).

```
[+] 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).

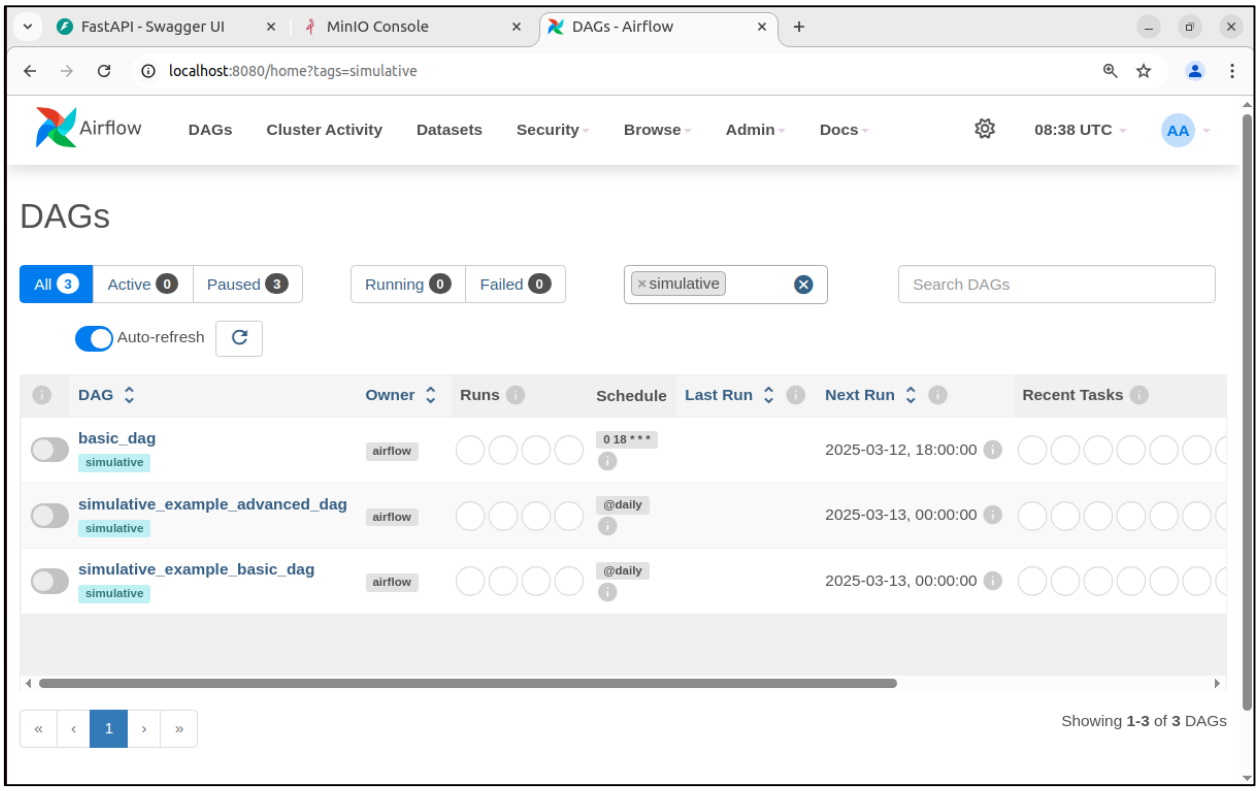


Рис. 7

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).

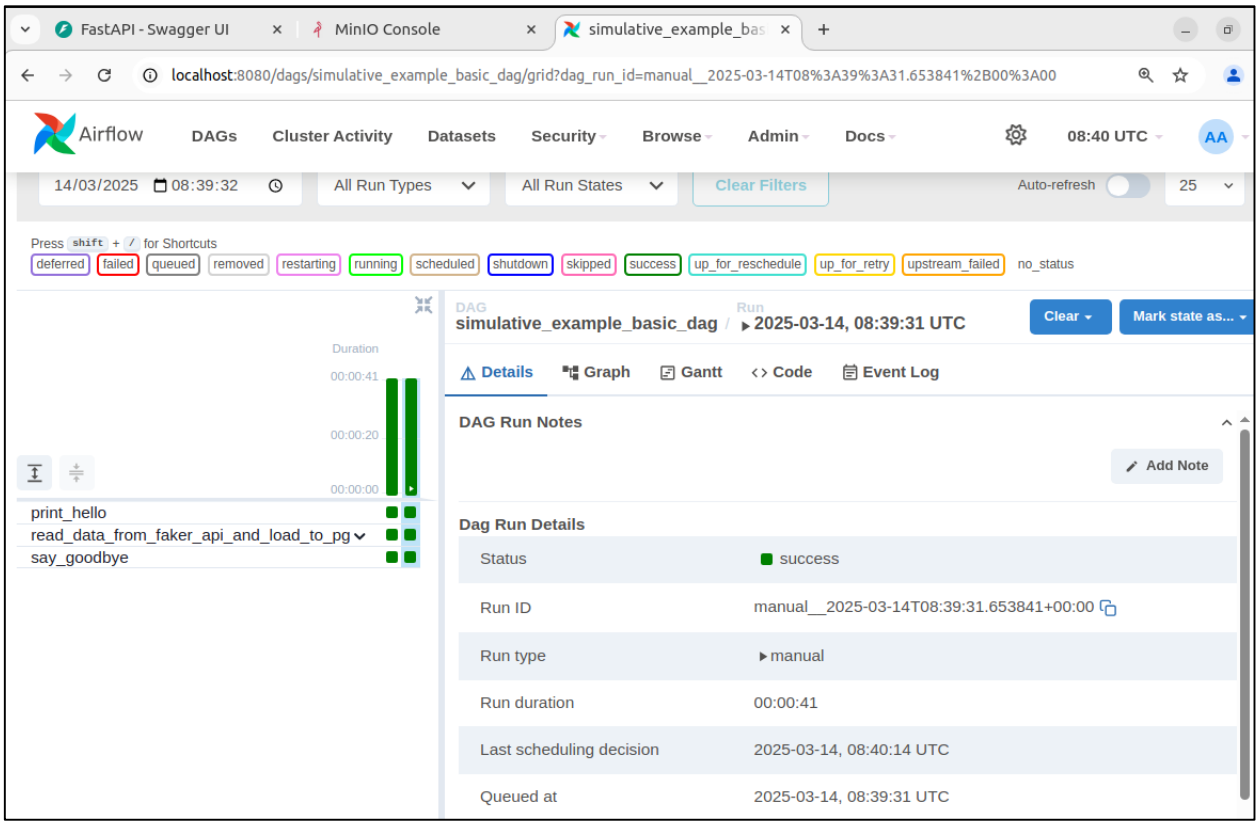


Рис. 8

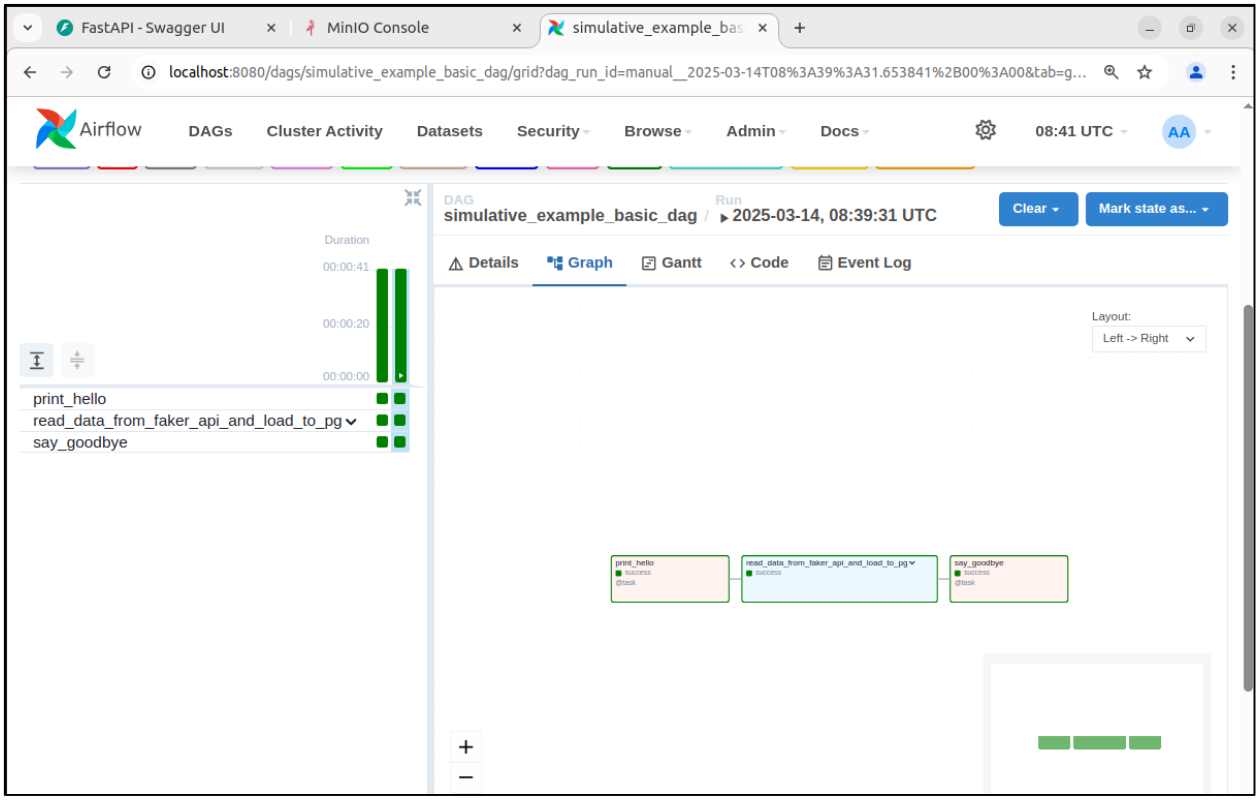


Рис. 9

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

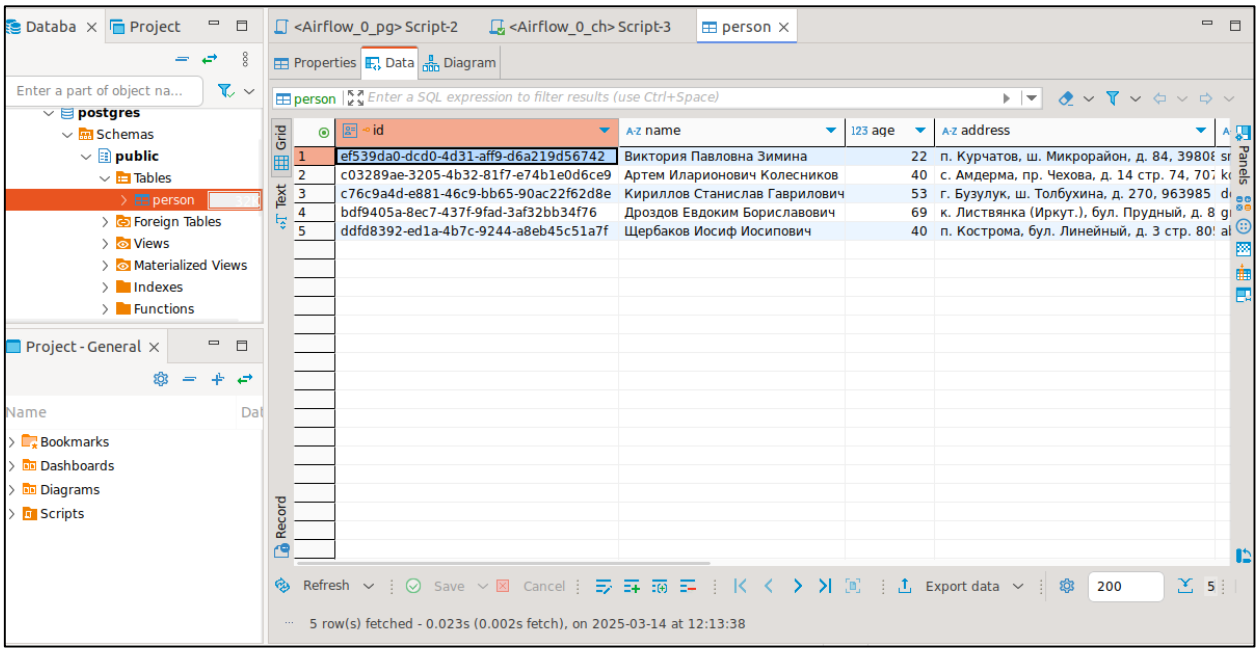


Рис. 10

Настроим dag для одновременного запуска нескольких задач (Рис. 11) и вызовем 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 даг, успешно (Рис. 13).

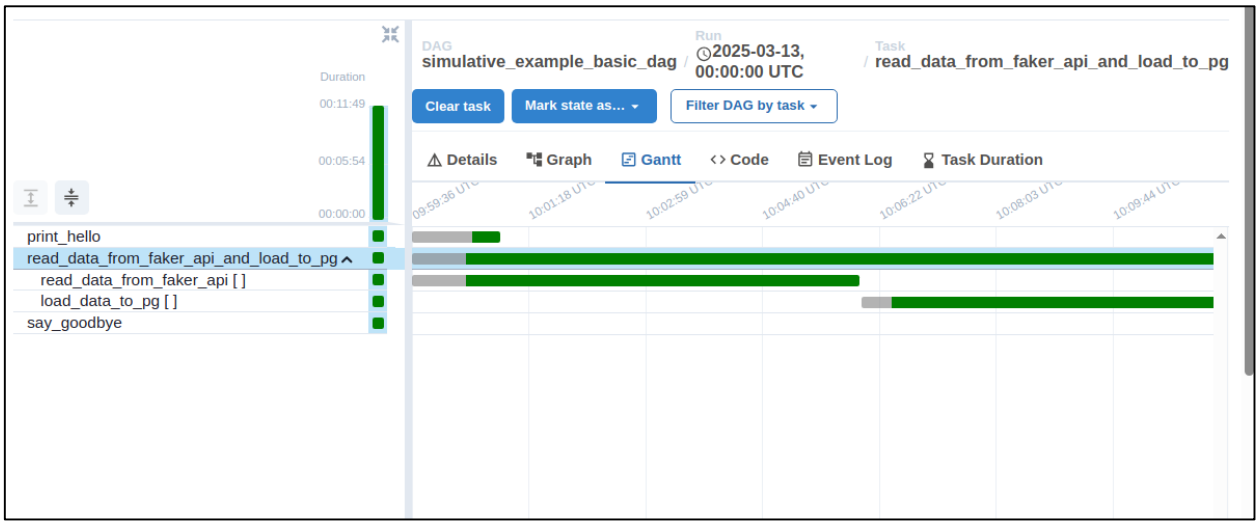


Рис. 13

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

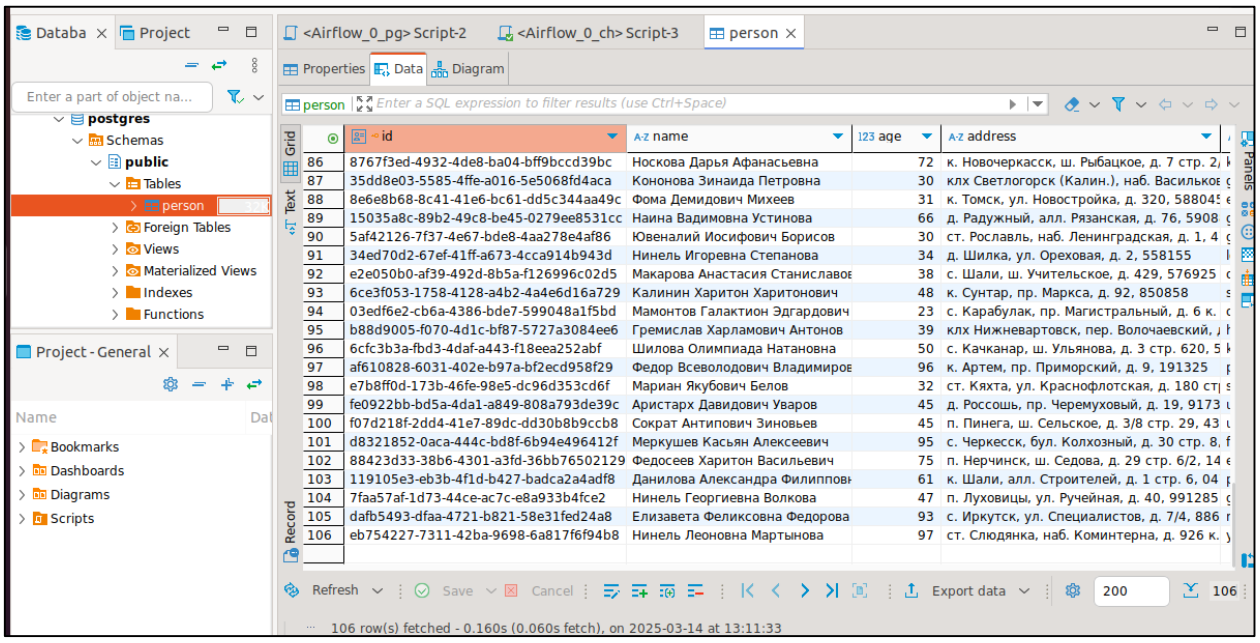


Рис. 14

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).

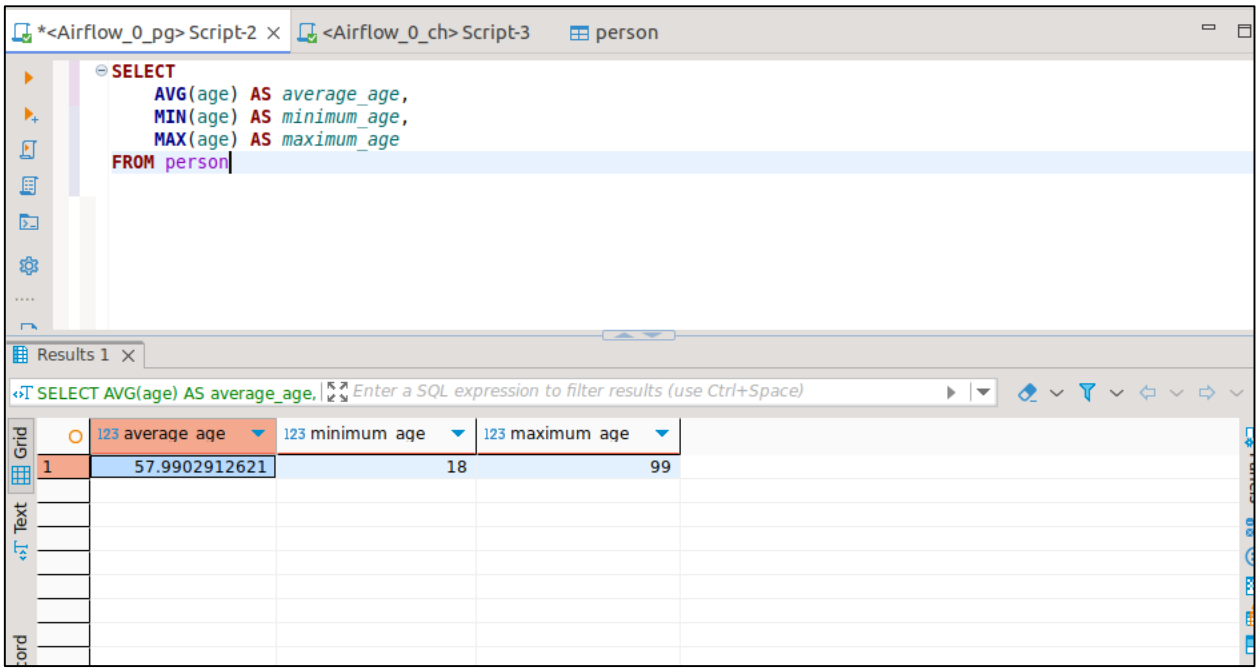


Рис. 15

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

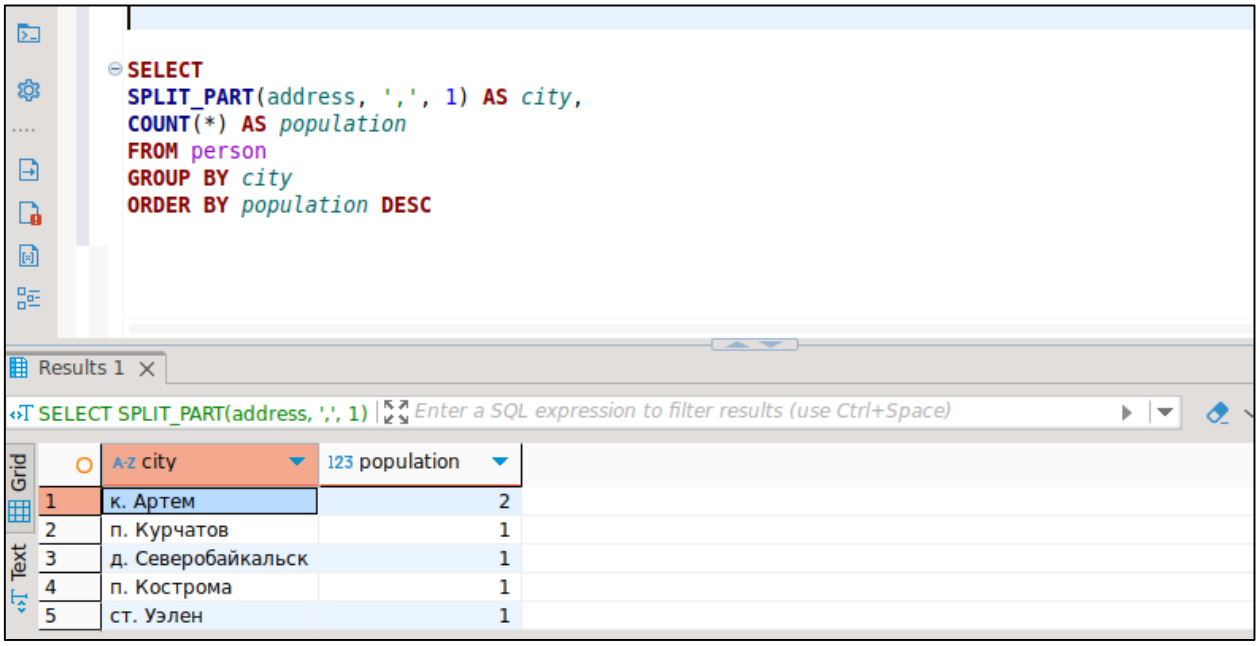
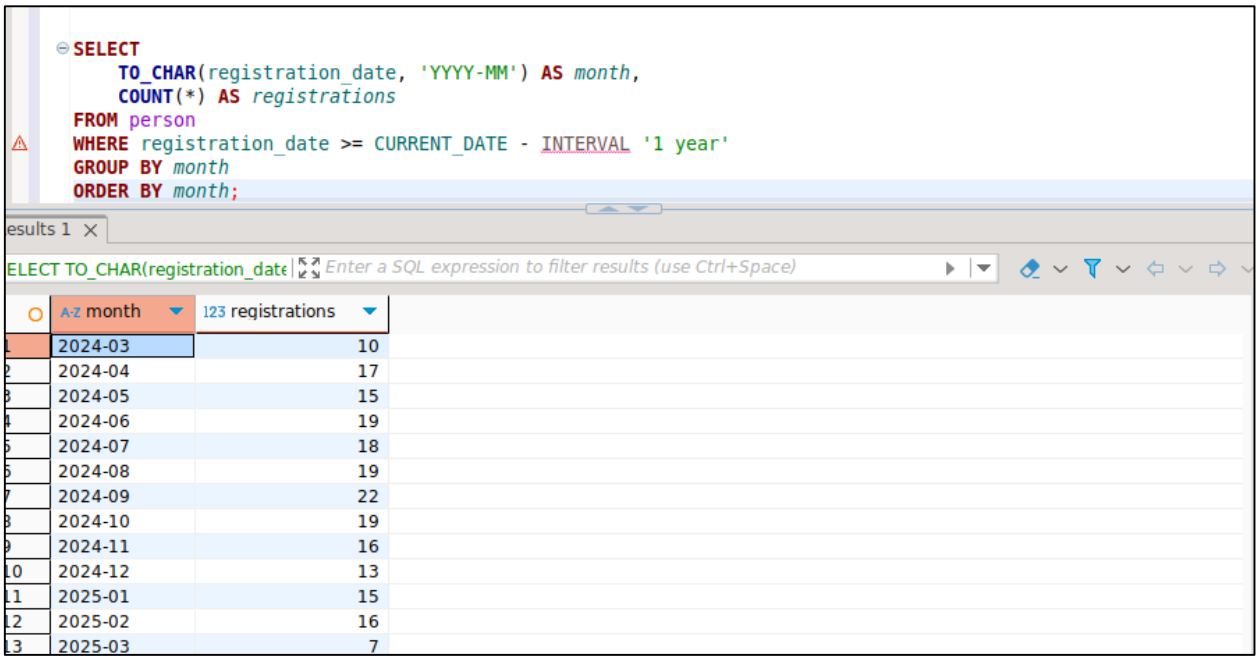


Рис. 16

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).



```
SELECT
    TO_CHAR(registration_date, 'YYYY-MM') AS month,
    COUNT(*) AS registrations
FROM person
WHERE registration_date >= CURRENT_DATE - INTERVAL '1 year'
GROUP BY month
ORDER BY month;
```

esults 1

ELECT TO_CHAR(registration_date

Enter a SQL expression to filter results (use Ctrl+Space)

	month	registrations
1	2024-03	10
2	2024-04	17
3	2024-05	15
4	2024-06	19
5	2024-07	18
6	2024-08	19
7	2024-09	22
8	2024-10	19
9	2024-11	16
10	2024-12	13
11	2025-01	15
12	2025-02	16
13	2025-03	7

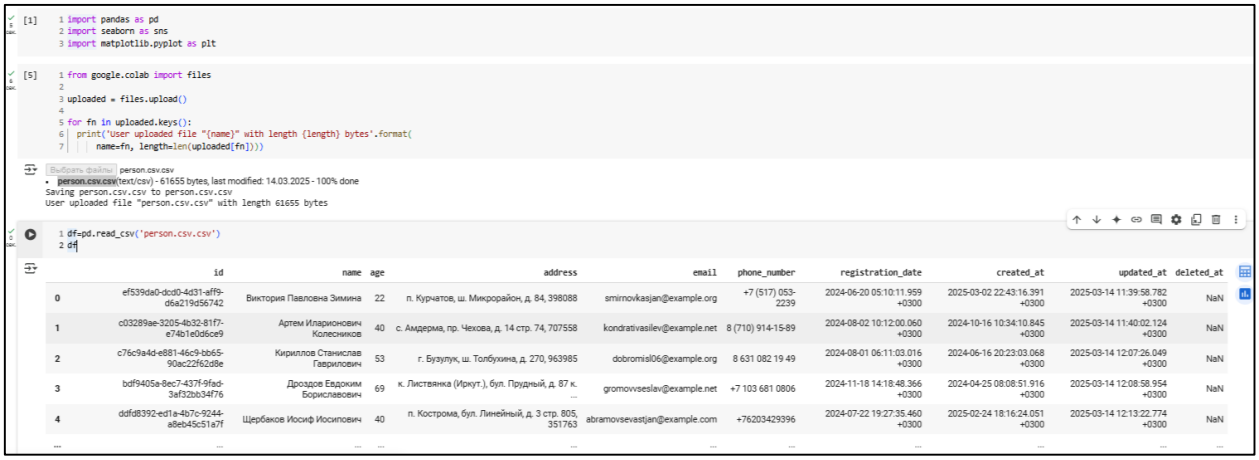
Рис. 17

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).



```
[1] 1 import pandas as pd
    2 import seaborn as sns
    3 import matplotlib.pyplot as plt

[5] 1 from google.colab import files
    2
    3 uploaded = files.upload()
    4
    5 for fn in uploaded.keys():
    6   print('User uploaded file "{name}" with length {length} bytes'.format(
    7       name=fn, length=len(uploaded[fn])))
```

Выбрать файлы person.csv

- person.csv(text/csv) - 61655 bytes, last modified: 14.03.2025 - 100% done

Saving person.csv.csv to person.csv.csv
User uploaded file "person.csv.csv" with length 61655 bytes

```
1 df=pd.read_csv('person.csv.csv')
2 df
```

	id	name	age	address	email	phone_number	registration_date	created_at	updated_at	deleted_at
0	ef539da0-dcd0-4d31-aff9-d6a219d56742	Виктория Павловна Зимина	22	п. Курчатов, ш. Микрорайон, д. 84, 398088	smirnovkasjan@example.org	+7 (517) 053-2239	2024-06-20 05:10:11.959 +0300	2025-03-02 22:43:16.391 +0300	2025-03-14 11:39:58.782 +0300	NaN
1	c03289ae-3205-4b32-81f7-e74b1e0d6ce9	Артем Илларионович Колесников	40	с. Амдерма, пр. Чехова, д. 14 стр. 74, 707558	kondrativasilev@example.net	8 (710) 914-15-89	2024-08-02 10:12:00.060 +0300	2024-10-16 10:34:10.845 +0300	2025-03-14 11:40:02.124 +0300	NaN
2	c76c9a4d-e881-46c9-bb65-90ac22f62d8e	Кириллов Станислав Гаврилович	53	г. Бузулук, ш. Толбухина, д. 270, 963985	dobromisl06@example.org	8 631 082 19 49	2024-08-01 06:11:03.016 +0300	2024-06-16 20:23:03.068 +0300	2025-03-14 12:07:26.049 +0300	NaN
3	bdf9405a-8ec7-437f-9fad-3af32bb34f76	Дроздов Евдоким Борислабович	69	к. Листвянка (Иркут.), бул. Прудный, д. 87 к. ...	gromovvseslav@example.net	+7 103 681 0806	2024-11-18 14:18:48.366 +0300	2024-04-25 08:08:51.916 +0300	2025-03-14 12:08:58.954 +0300	NaN
4	ddfd8392-ed1a-4b7c-9244-a8eb45c51a7f	Щербаков Иосиф Иосипович	40	п. Кострома, бул. Линейный, д. 3 стр. 805, 351763	abramovsevastjan@example.com	+76203429396	2024-07-22 19:27:35.460 +0300	2025-02-24 18:16:24.051 +0300	2025-03-14 12:13:22.774 +0300	NaN
...	...	...	...	...	...	...	...	...	...	...

Рис. 18

Далее просмотрим общее описание данных (Рис. 19).

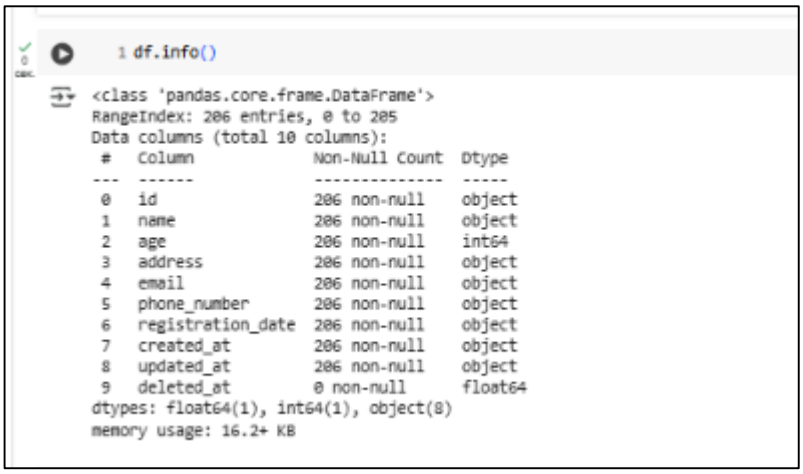


Рис. 19

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

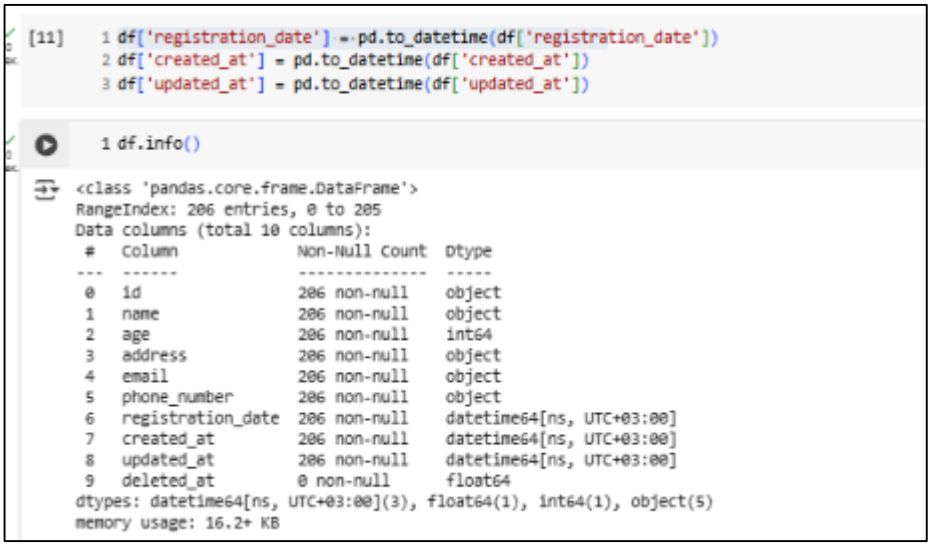


Рис. 20

Далее построим график распределения возрастов (Рис. 21).

```python
plt.figure(figsize=(10, 6))
plt.hist(df['age'], bins=20, color='skyblue', edgecolor='black')
plt.title('Гистограмма распределения возраста')
plt.xlabel('Возраст')
plt.ylabel('Количество людей')
plt.grid(axis='y', alpha=0.75)
plt.xticks(range(0, df['age'].max() + 1, 5))
plt.show()

```

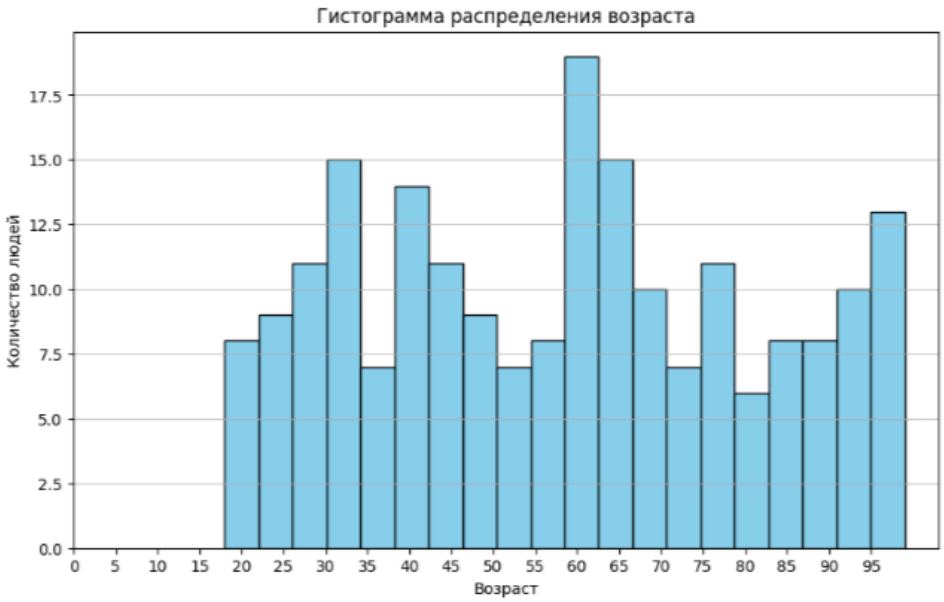


Рис. 21

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

```python
df1=df
# Извлекаем название города (первая часть 'address', разделенная запятой)
df1['city'] = df['address'].apply(lambda x: x.split(',')[0])

# Считаем количество проживающих в каждом городе
city_counts = df1['city'].value_counts()

# Берем топ-5 городов
top_cities = city_counts.head(5)

# Строим диаграмму
plt.figure(figsize=(9, 5))
top_cities.plot(kind='bar', color='skyblue', edgecolor='black')

# Добавляем подписи
plt.xlabel("Город")
plt.ylabel("Количество проживающих")
plt.title("Топ-5 городов по количеству проживающих")
plt.xticks(rotation=45)  # Поворот подписей для удобства
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Показываем график
plt.show()

```

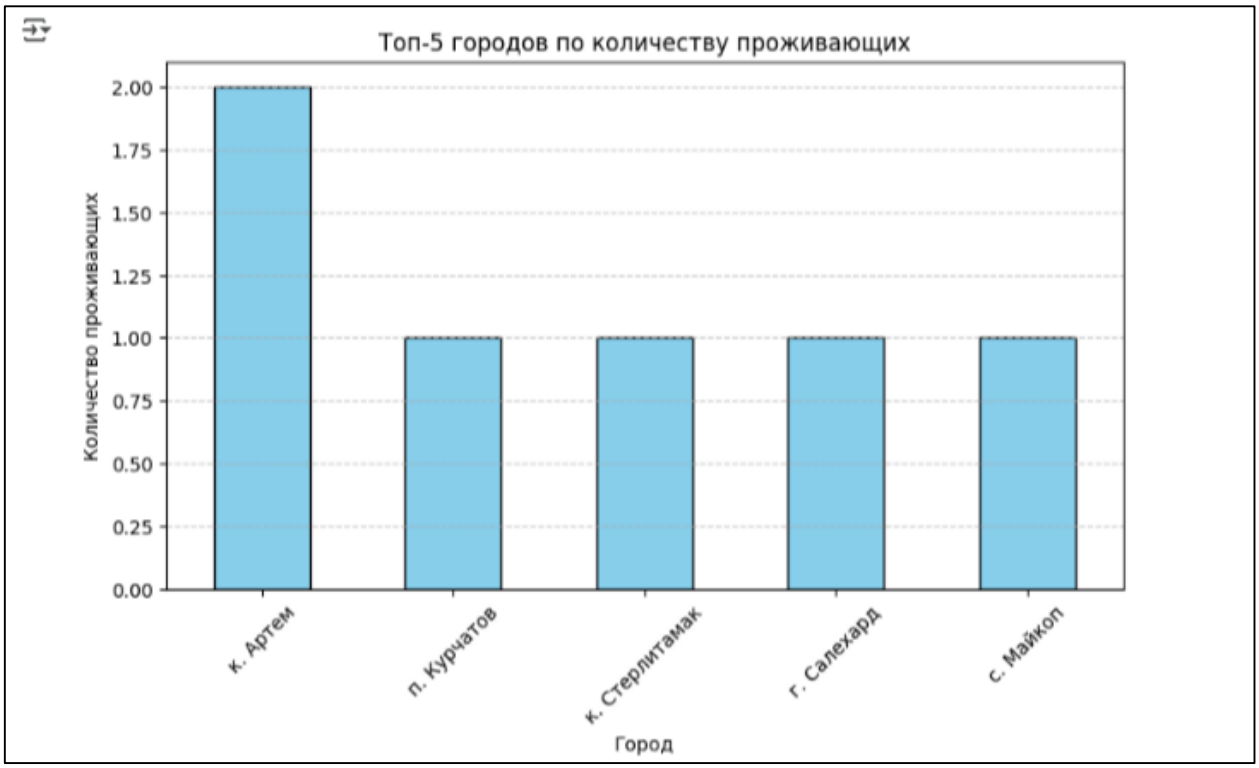


Рис. 22

Далее статистика по количеству регистраций за каждый месяц.

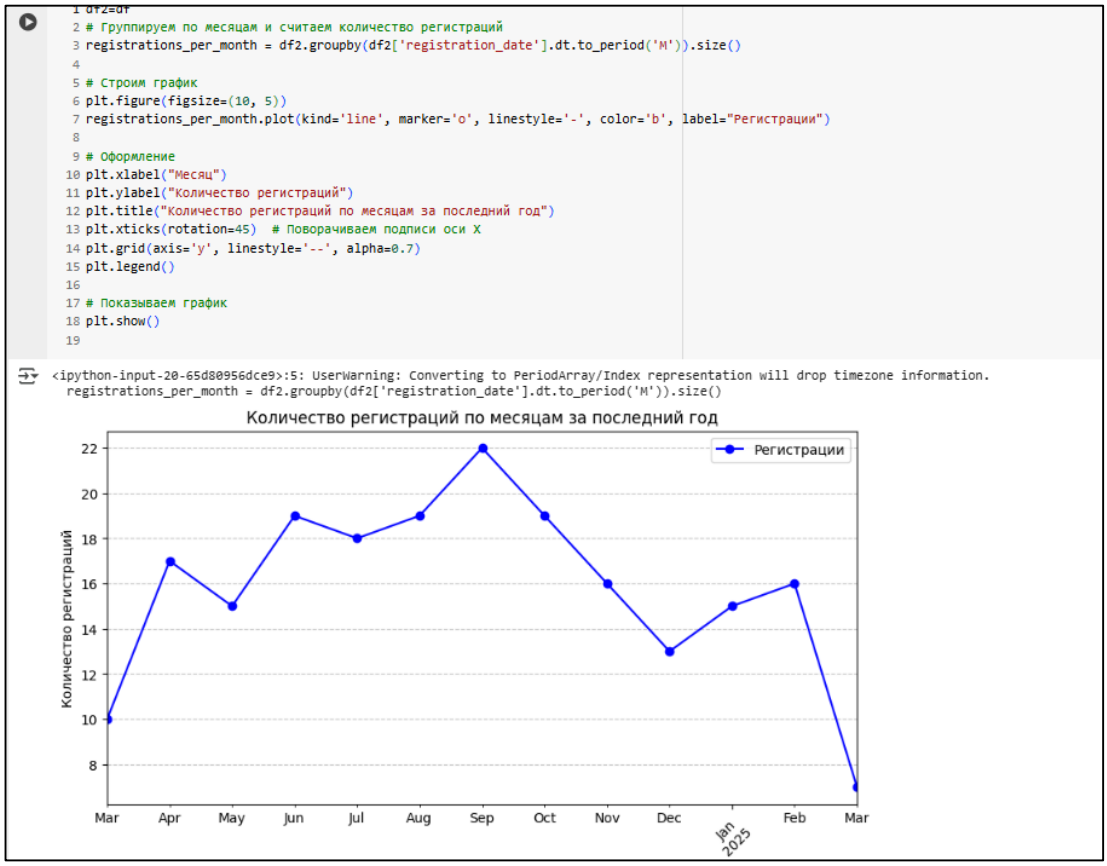


Рис. 23

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

Задание 2. Провести анализ данных, загруженных в таблицу person_count_by_city в ClickHouse, и создать визуализации для полученных результатов.

Сначала необходимо выполнить dag *simulative_example_advanced_dag*, успешно (Рис. 24).

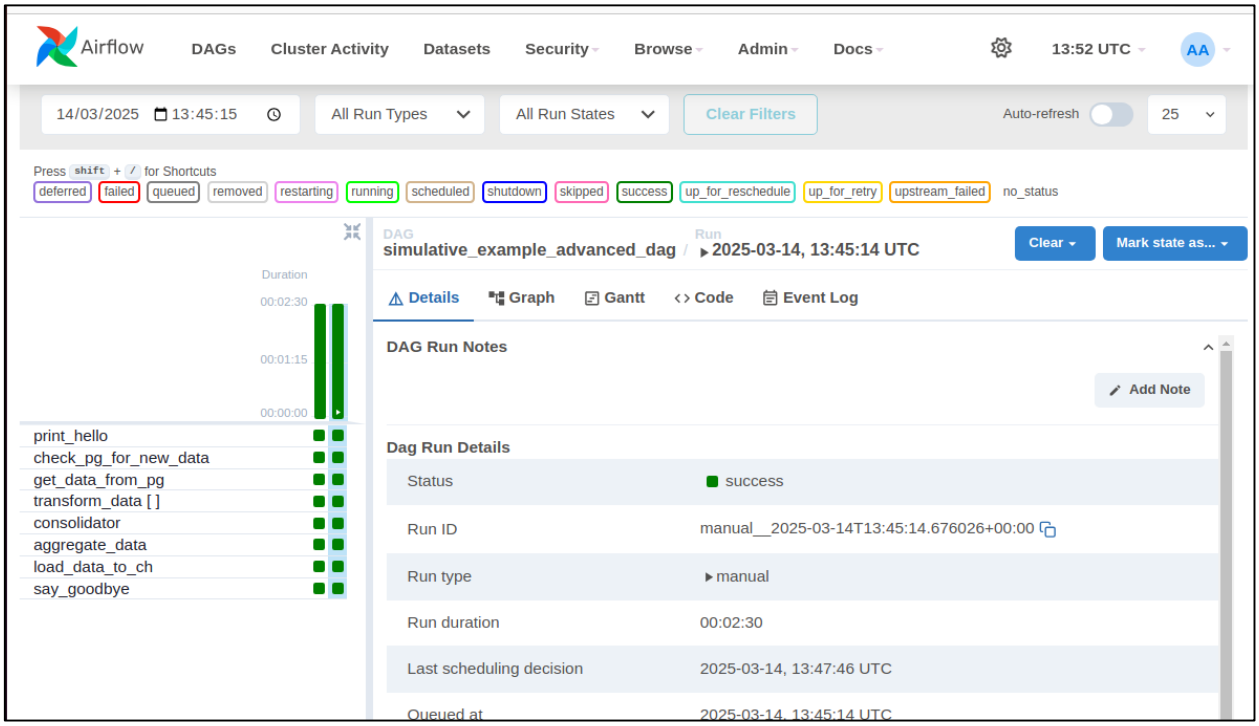


Рис. 24

Далее проверим Dbeaver, что появились города из таблицы person (Рис. 25).

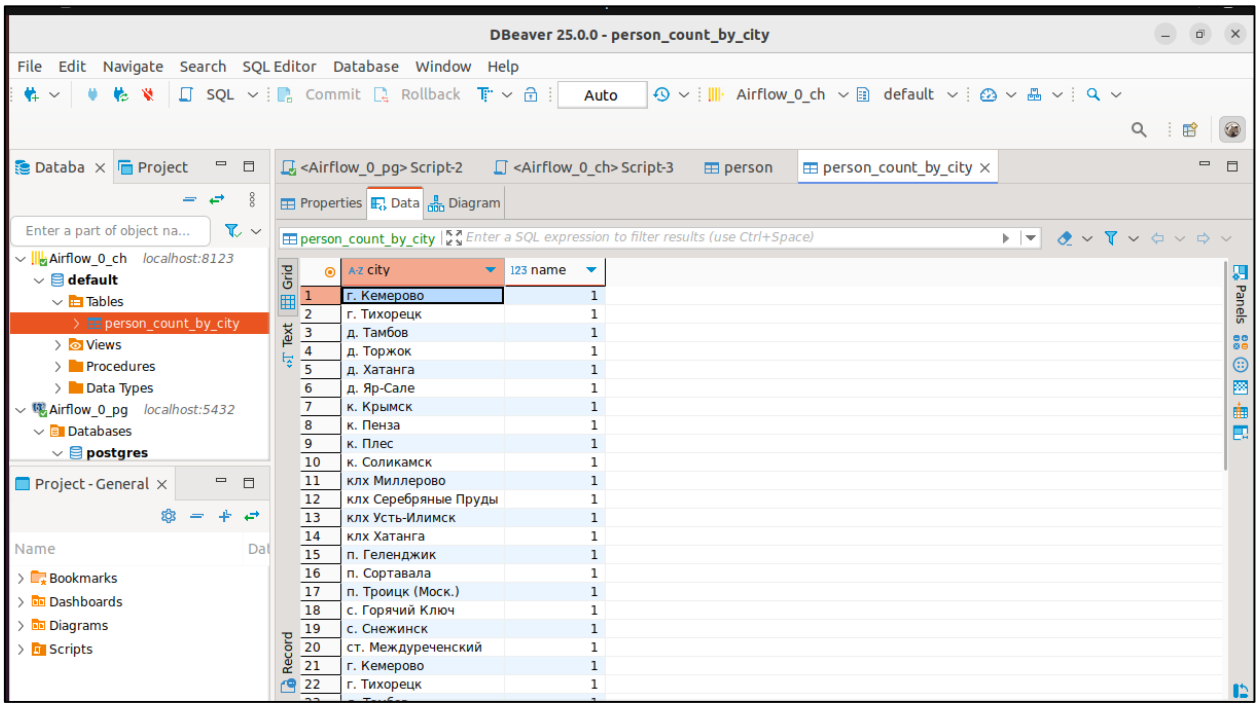


Рис. 25

2.1 Найдите топ-10 городов с наибольшим количеством людей (Рис. 26).

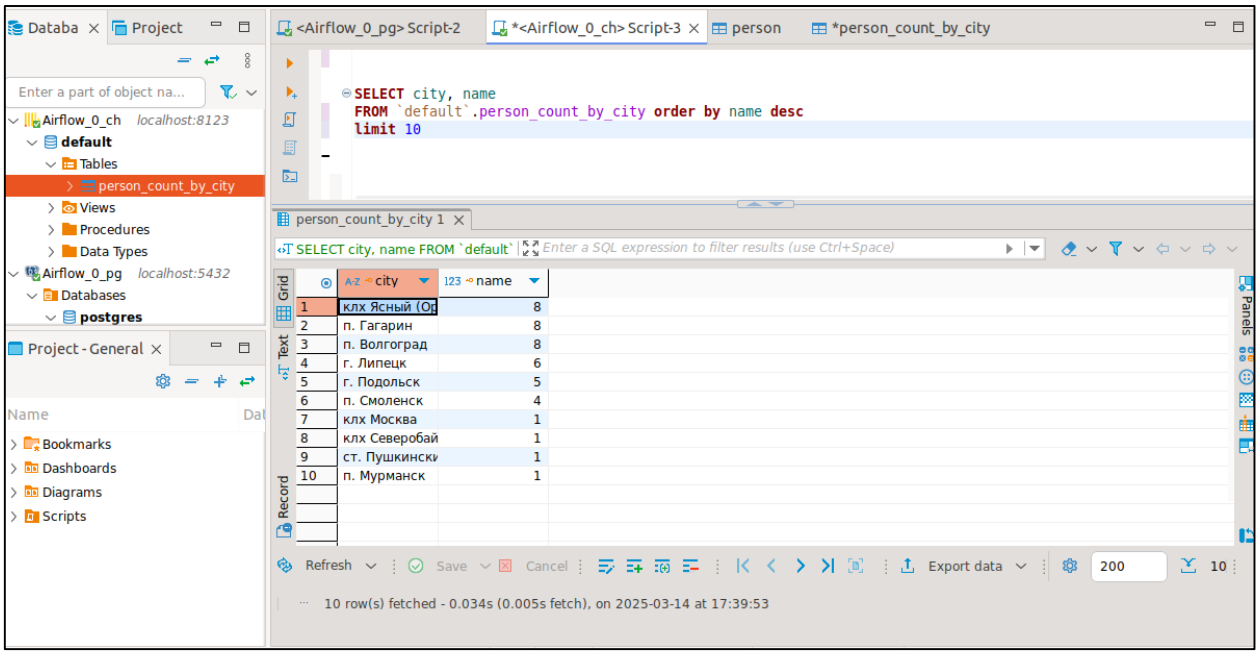


Рис. 26

Далее извлечем все строки, которые начинаются с «г» - то есть город т.к. он идет первым с помощью функции clickhouse - *splitByChar* и посчитаем среднее количество людей во всех городах (Рис. 27).

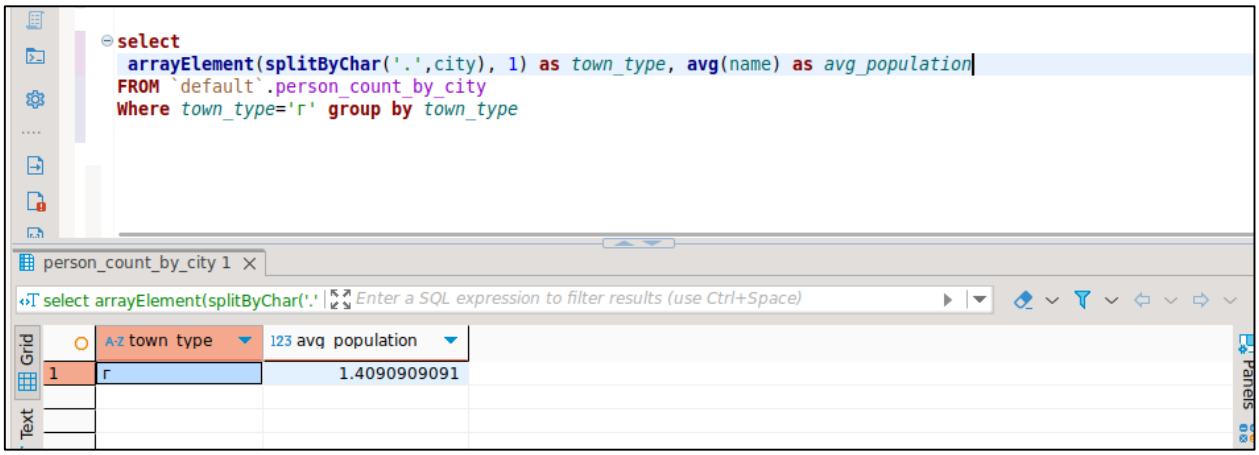


Рис. 27

2.2. Постройте гистограмму распределения количества людей по городам.

Рассмотрено в п.2.4 с построением визуализаций.

2.3. Города с наибольшим максимальным количеством людей (Рис. 28).

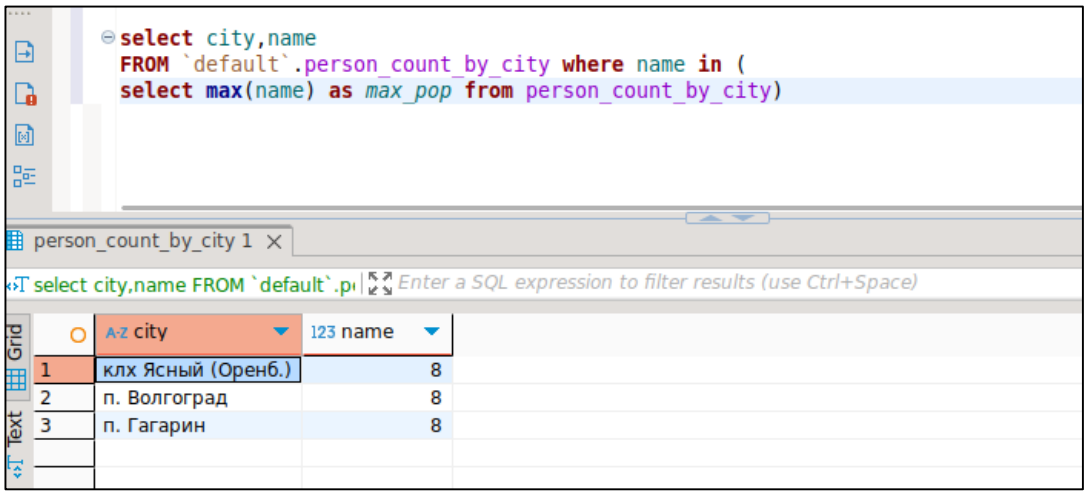


Рис. 28

Города с минимальным количеством людей (Рис. 29).

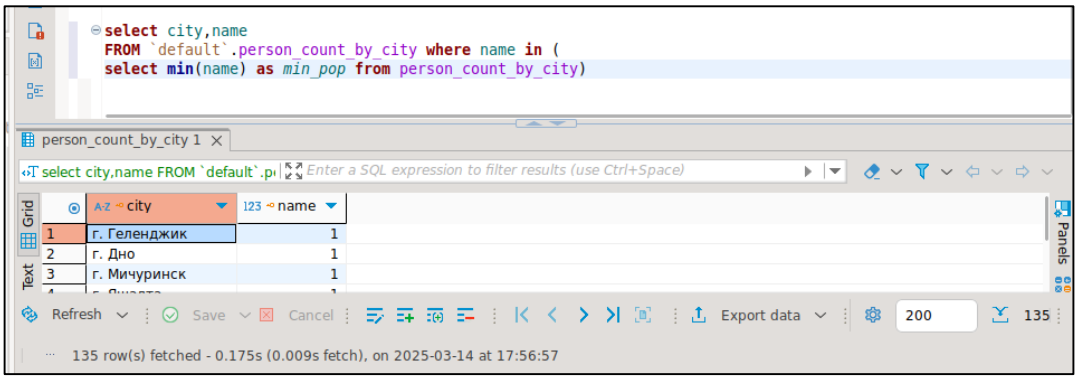


Рис. 29

2.4. Создайте графики

Топ-10 городов по количеству людей (Рис. 30).

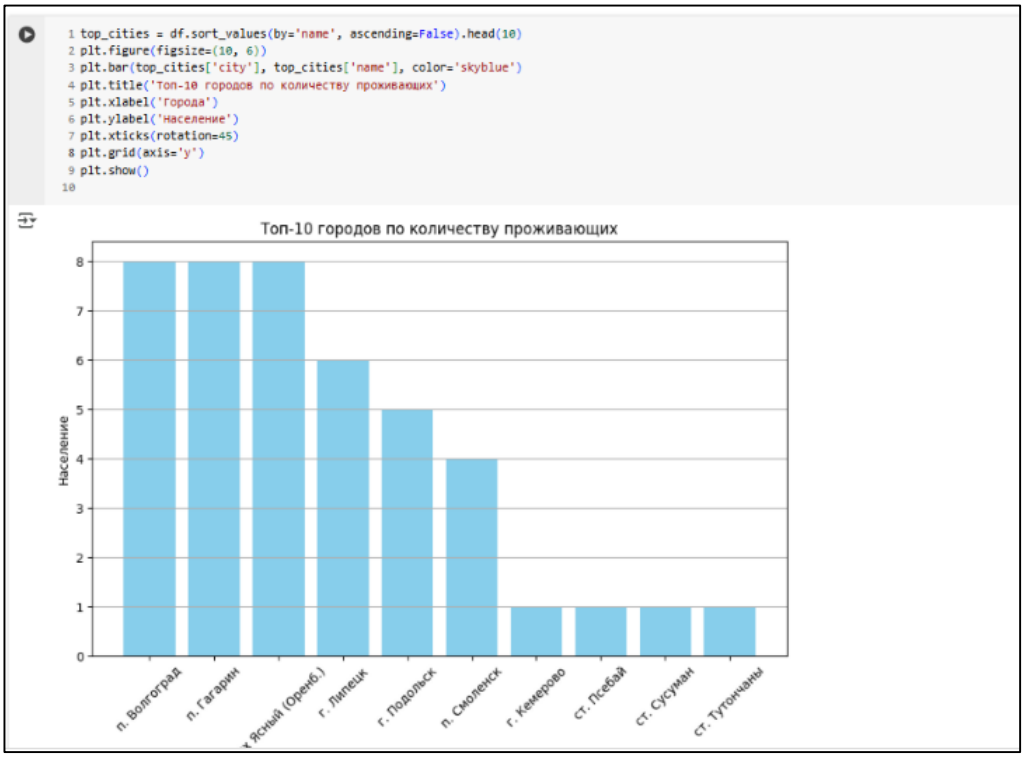


Рис. 30

Гистограмма распределения количества людей по городам (Рис. 31).

```python
plt.figure(figsize=(10, 6))
plt.hist(df['name'], bins=5, color='lightcoral', edgecolor='black')
plt.title('Гистограмма распределения населения по городам')
plt.xlabel('Население')
plt.ylabel('Количество городов')
plt.grid(axis='y', alpha=0.75)
plt.show()
```

Рис. 31

Линейный график, показывающий минимальное и максимальное количество людей в городах (Рис. 32).

```python
min_population = df['name'].min()
max_population = df['name'].max()
line_data = pd.DataFrame({
    'Города': ['Минимум', 'Максимум'],
    'Население': [min_population, max_population]
})
plt.figure(figsize=(10, 6))
plt.plot(line_data['Города'], line_data['Население'], marker='o', color='purple', linestyle='--')
plt.title('Минимальное и максимальное количество людей в городах')
plt.xlabel('Тип')
plt.ylabel('Население')
plt.grid()
plt.show()
```

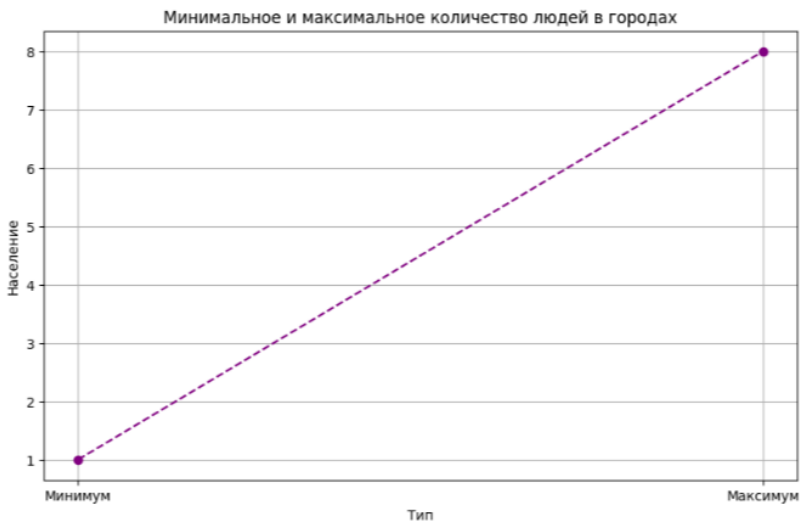


Рис. 32

Вывод: были запущены все необходимые контейнеры, также выполнен и настроен dag на выполнение необходимого количества вызовов. Были построены визуализации разных параметров из двух таблиц, результаты визуализаций совпали с результатами запросов к БД.