

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

ОТЧЕТ

по дисциплине «Проектный практикум по разработке ETL-решений»
Лабораторная работа 3.1.: Интеграция данных из нескольких источников.
Обработка и согласование данных из разных источников
Направление подготовки – 38.03.05 «Бизнес-информатика».
профиль подготовки – «Аналитика данных и эффективное управление»

Выполнила:

St92

Руководитель:

1

Москва
2025 год

Цель работы: получить практические навыки интеграции, обработки и согласования данных из различных источников с использованием Python и его библиотек.

Задачи

1. Изучить методы чтения данных из разных источников.
2. Освоить техники обработки и очистки данных.
3. Научиться согласовывать данные из разных источников.
4. Реализовать сохранение обработанных данных.

Ход работы:

Подготовим файлы, разархивируем архив:

```
dev@dev-vm:~/Downloads/lecture_0_airflow$ unrar x env_flake8.rar

UNRAR 6.11 beta 1 freeware      Copyright (c) 1993-2022 Alexander Roshal

Extracting from env_flake8.rar

Extracting .pre-commit-config.yaml      OK
Extracting .env.example                 OK
Extracting .flake8                      OK
Extracting .gitignore                   OK
All OK
```

Рисунок 1 – Получение скрытых файлов

Проверим запущенные контейнеры:

```
dev@dev-vm:~/Downloads/lecture_0_airflow$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
dev@dev-vm:~/Downloads/lecture_0_airflow$
```

Рисунок 2 – Выполнение команды для просмотра активных контейнеров

Запустим все сервисы:

```
dev@dev-vm:~/Downloads/lecture_0_airflow$ make up-services
docker compose -f docker-compose-services.yaml up -d --build
[+] Running 17/47
  minio [          ] Pulling
  pg [██████████] Pulling
  ch [          ] Pulling
  zookeeper [██████████] 103.6MB / 114.5MB Pulling
```

Рисунок 3 – Выполнение команды для запуска всех сервисов

Посмотрим все контейнеры:

```
dev@dev-vm:~/Downloads/lecture_0_airflow$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
ddc080b5ed2d	lecture_0_airflow-faker-api	"uvicorn app.app:app..."	26 seconds ago	Up 10 seconds
07b975e68347	lecture_0_airflow-faker-api-1			
07b975e68347	clickhouse/clickhouse-server:24-alpine	"/entrypoint.sh"	26 seconds ago	Up 23 seconds (healthy)
3ec5f00bc747	zookeeper:latest	"/docker-entrypoint..."	26 seconds ago	Up 24 seconds
b350d97cfa09	quay.io/minio/minio	"/usr/bin/docker-ent..."	26 seconds ago	Up 24 seconds (healthy: starting)
d7cb034ca83f	lecture_0_airflow-minio-1			
d7cb034ca83f	postgres:latest	"docker-entrypoint.s..."	26 seconds ago	Up 24 seconds (healthy)

Рисунок 4 – Выполнение команды для просмотра активных контейнеров

Посмотрим, как работает сервис по порту 8000:

MinIO Console x FastAPI - Swagger UI x +

localhost:8000/docs#/person/get_person_person_get

pgAdmin 4 MinIO Console

Responses

Curl

curl -X 'GET' \
'http://localhost:8000/person/' \
-H 'accept: application/json'

Request URL

http://localhost:8000/person/

Server response

Code Details

200

Response body

{
 "id": "95fd3f48-7ae5-4bad-bc5f-3b6ad3f2ae5d",
 "name": "Алексеев Леонид Викентьевич",
 "age": 19,
 "address": "г. Черусти, алл. Ореховая, д. 4 стр. 60, 574892",
 "email": "david_63@example.org",
 "phone_number": "+7 (443) 940-5561",
 "registration_date": "2024-06-08T11:33:19.659028",
 "created_at": "2024-05-27T03:20:33.898527",
 "updated_at": "2025-04-10T22:08:28.707792",
 "deleted_at": null
}

Response headers

content-length: 416
content-type: application/json
date: Thu, 10 Apr 2025 22:08:28 GMT
server: uvicorn

Рисунок 5 – Активный FastAPI

Проверим временное хранилище по порту 9002:

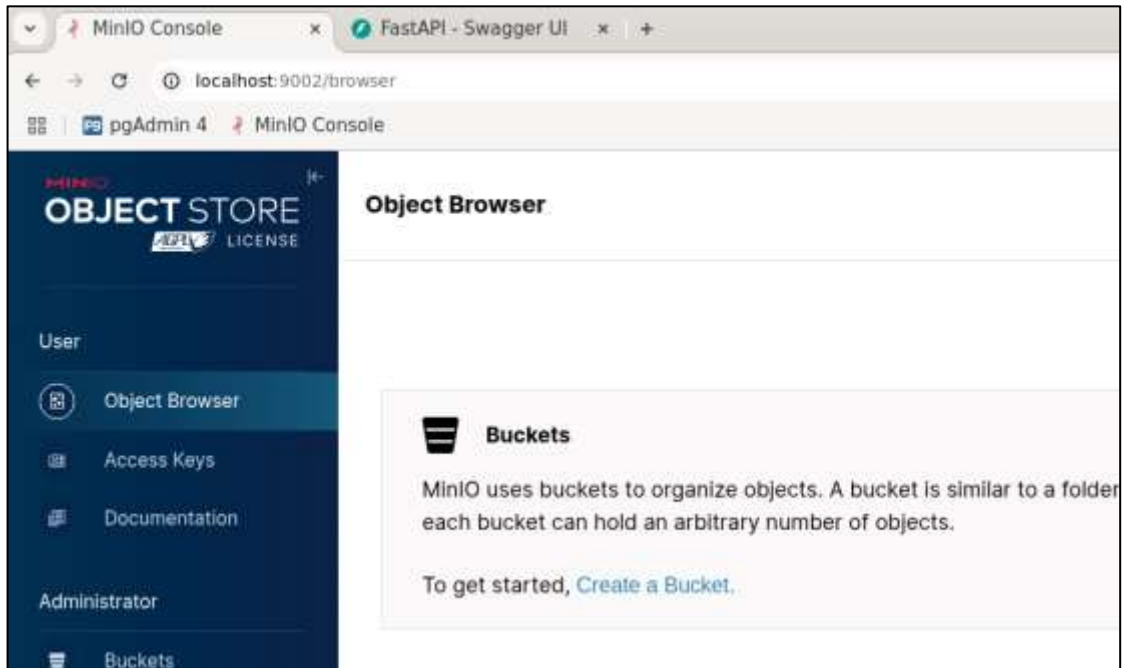


Рисунок 6 – Активный MinIO Console

Подключимся к ClickHouse и Postgres через DBeaver:

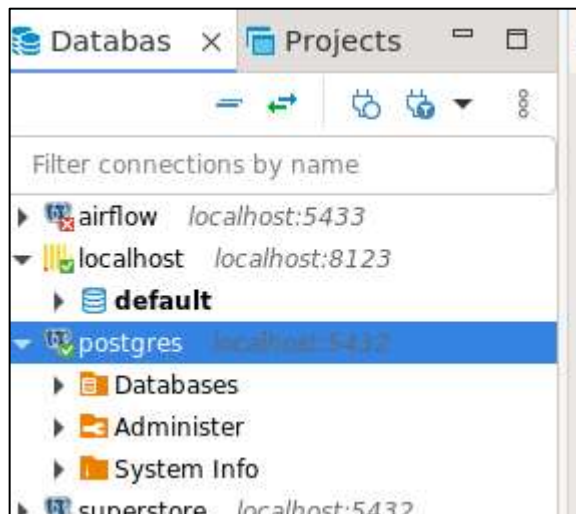


Рисунок 7 – Успешное подключение к базам данных

Посмотрим содержимое таблицы в Postgres:

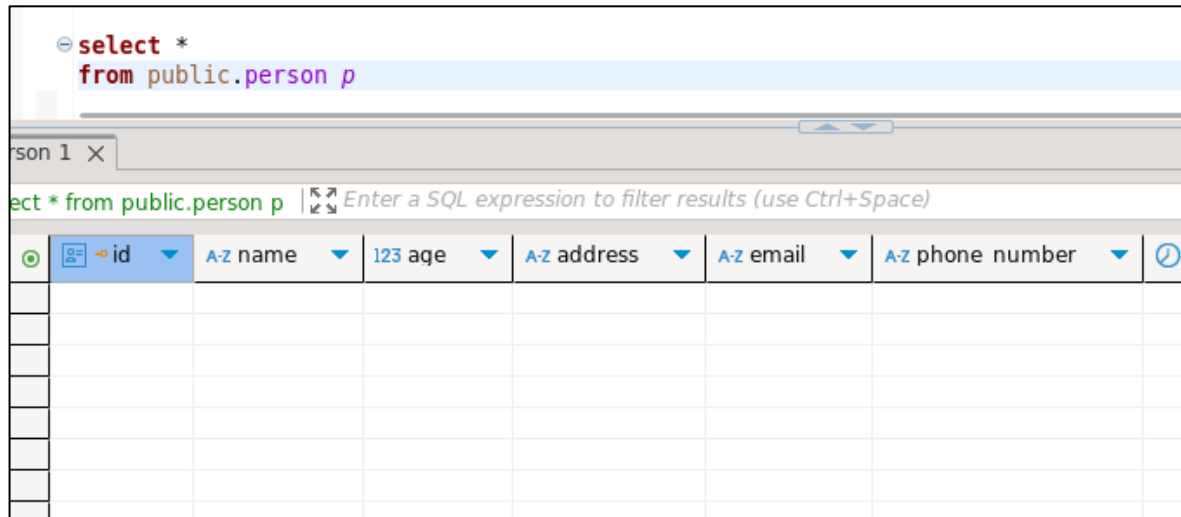


Рисунок 8 – Выполнение SQL-запроса к БД

Посмотрим на созданную базу данных в clickhouse:

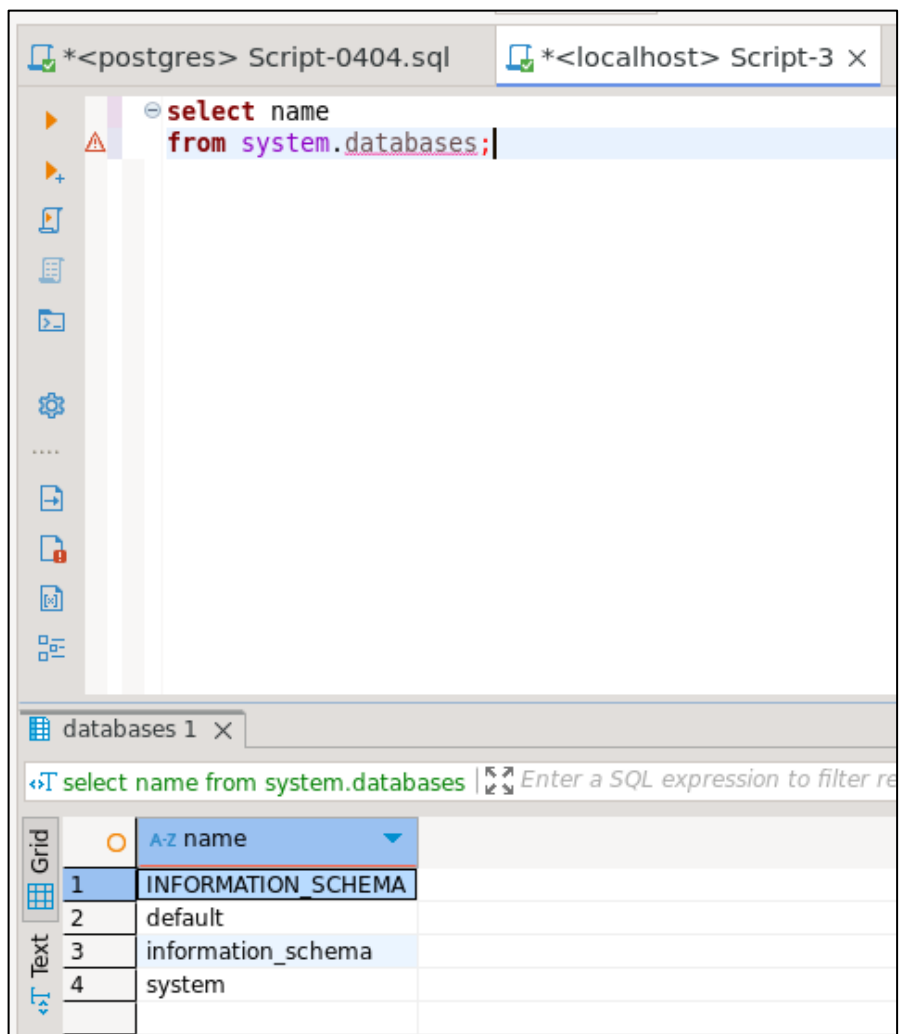


Рисунок 9 – Выполнение SQL-запроса к базе данных

Запустим Airflow:

```
dev@dev-vm:~/Downloads/lecture_0_airflow$ make up-af
docker compose -f docker-compose-af.yaml up -d --build
[+] Running 0/7
  :: airflow-scheduler Pulling                                1.7s
  :: airflow-triggerer Pulling                                1.7s
  :: postgres Pulling                                         1.7s
  :: airflow-init Pulling                                     1.7s
  :: airflow-worker Pulling                                   1.7s
  :: redis Pulling                                             1.7s
  :: airflow-webserver Pulling                                1.7s
```

Рисунок 10 – Запуск Airflow

Посмотрим монитор ресурсов:

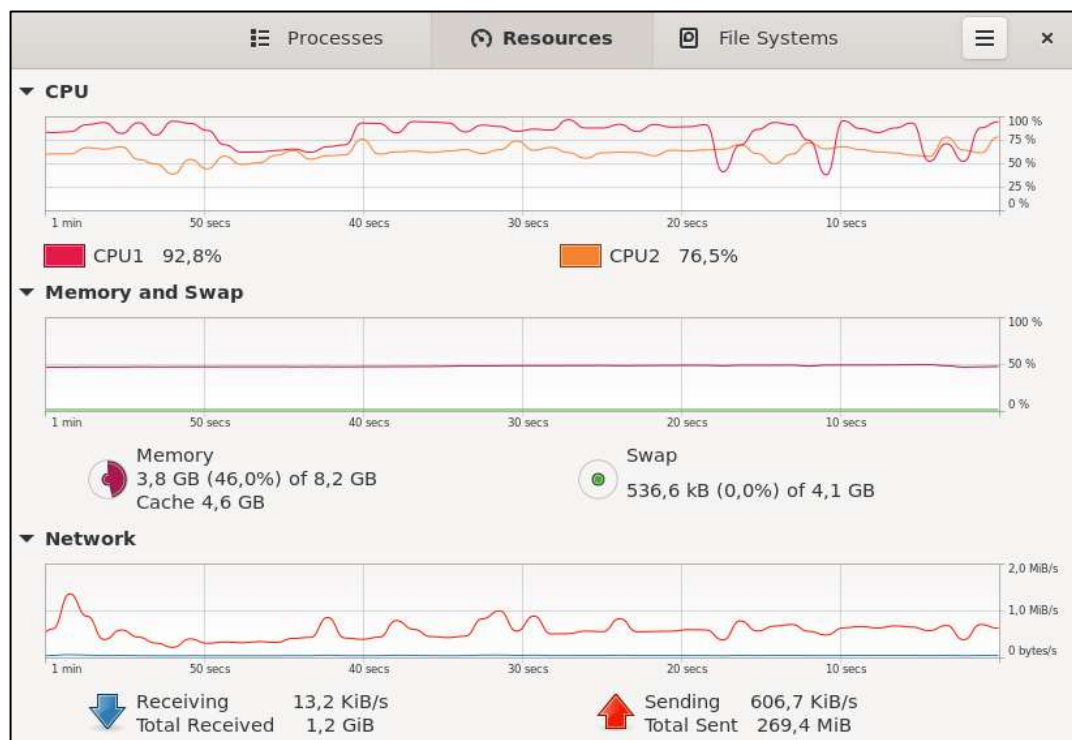


Рисунок 11 – Монитор ресурсов ВМ

Зайдем в Airflow по порту 8080:

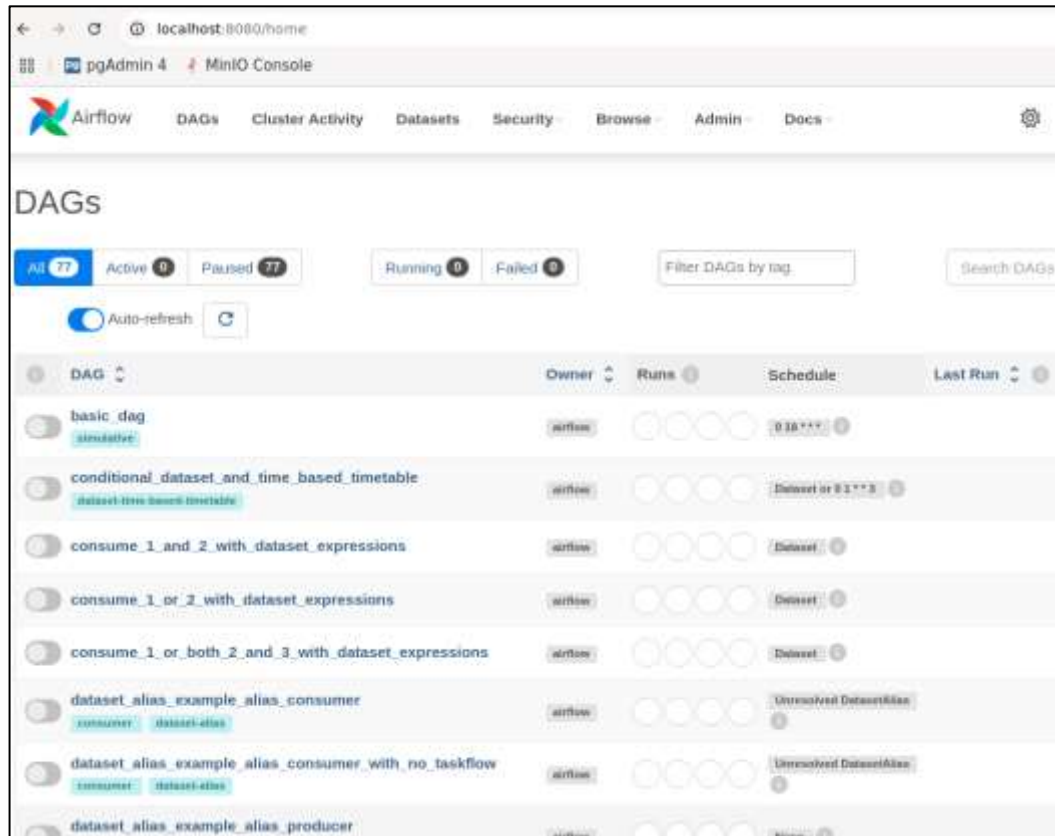


Рисунок 12 – Доступность Airflow

Запустим DAG для проверки доступности серверов:

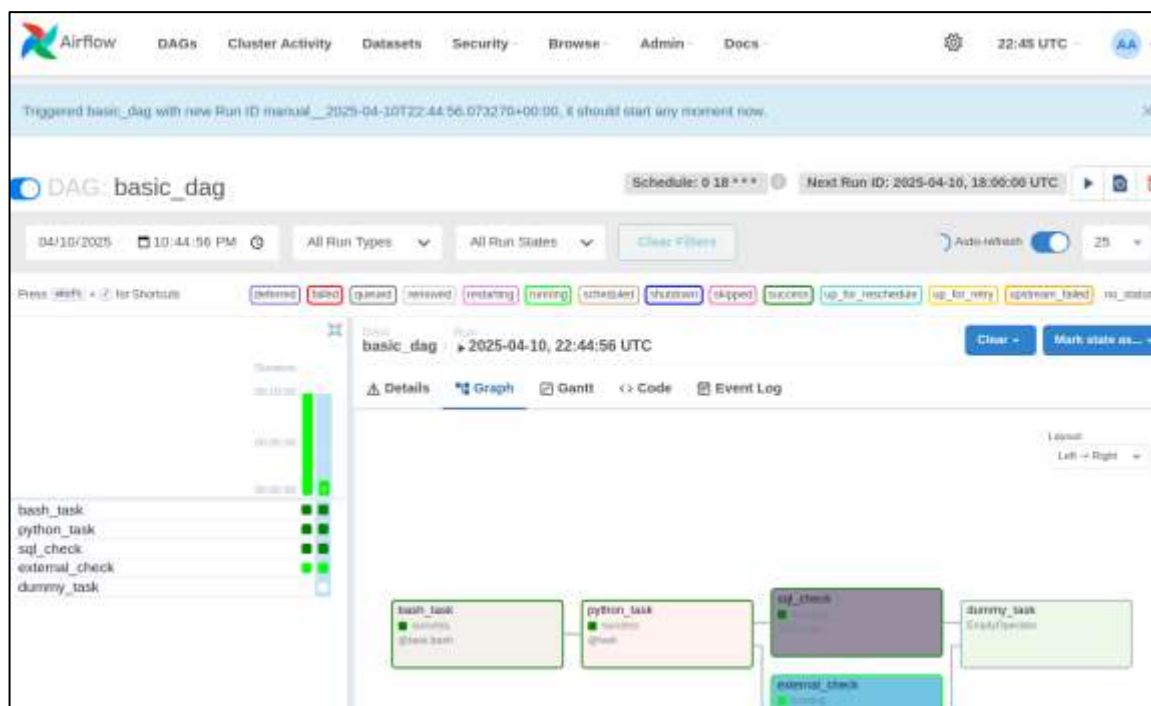


Рисунок 13 – Выполнение basic_dag

Просмотрим полученные данные в MinIO

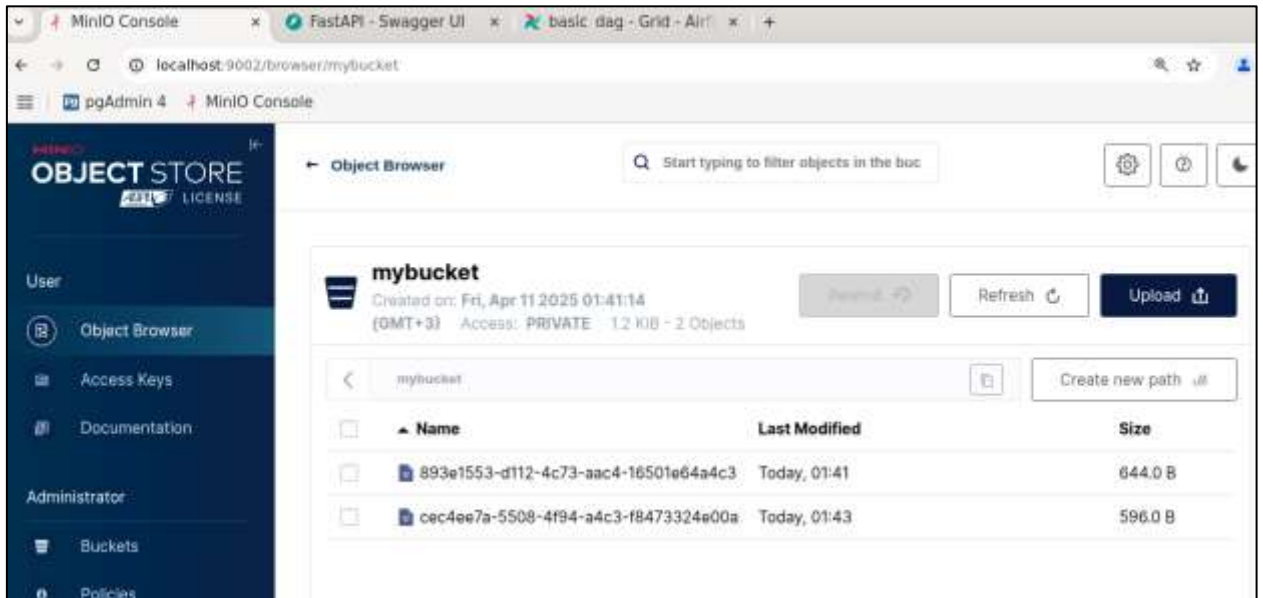


Рисунок 14 – Записанные данные в MinIO

Запросим данные из таблицы postgres:

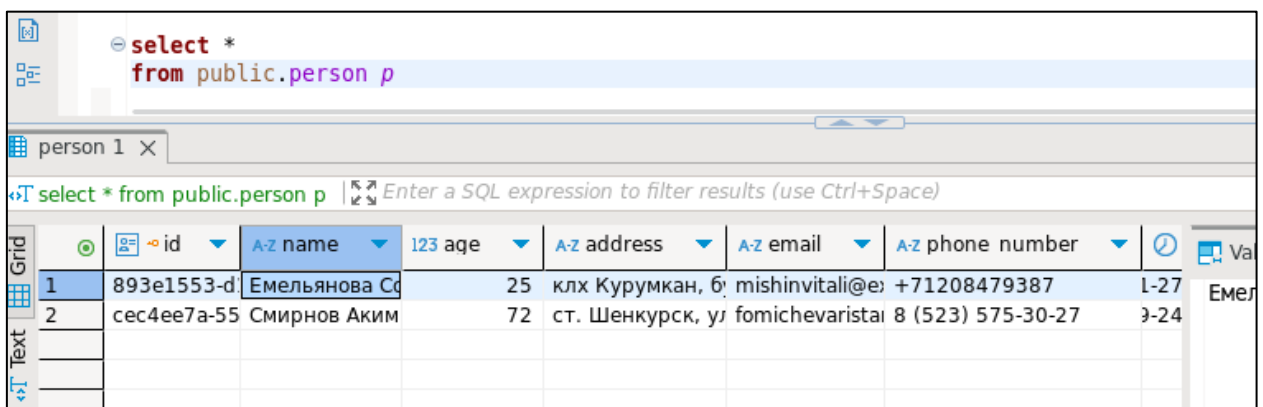


Рисунок 15 – Запрос к БД

Заполним таблицу person в базе данных PostgreSQL фейковыми данными не менее 100 записей

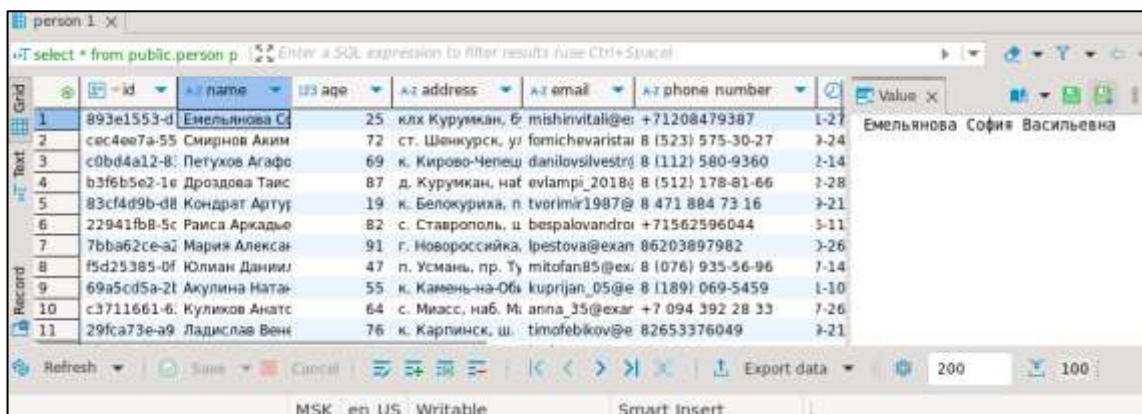
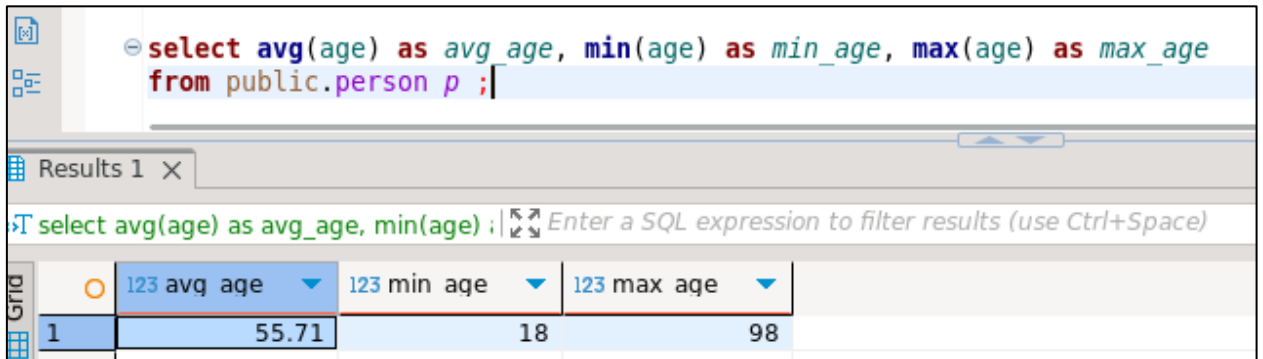


Рисунок 16 – Заполненная таблица

Рассчитаем средний, минимальный и максимальный возраст:



```
select avg(age) as avg_age, min(age) as min_age, max(age) as max_age
from public.person p ;
```

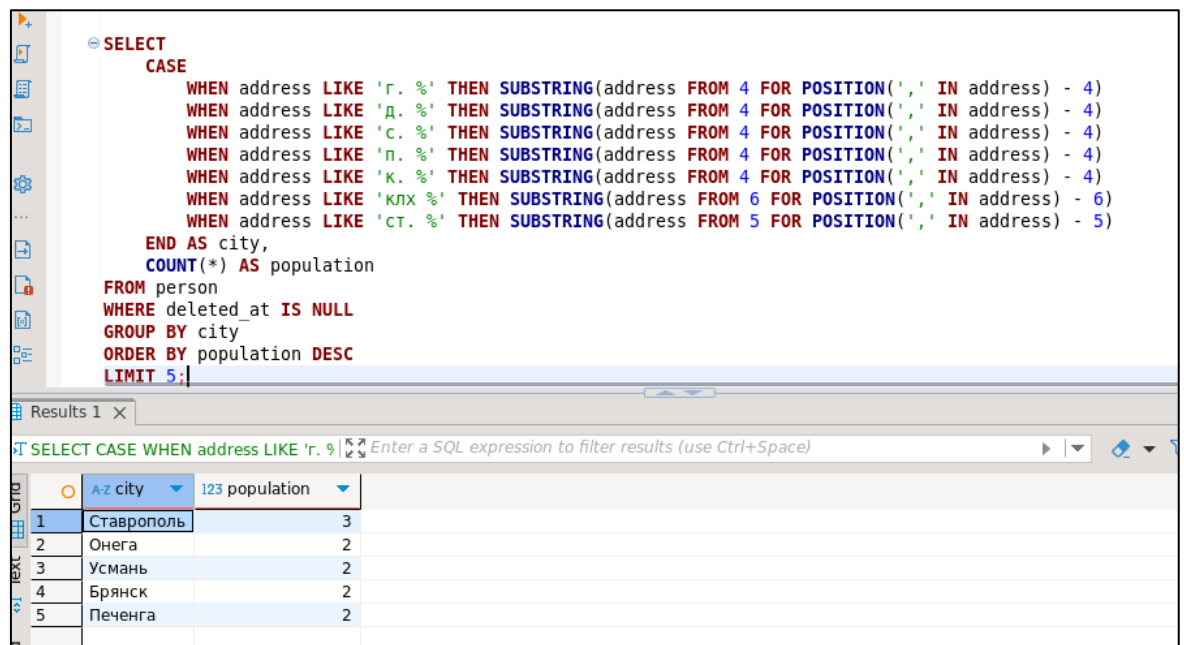
Results 1 X

select avg(age) as avg_age, min(age) ; Enter a SQL expression to filter results (use Ctrl+Space)

	123 avg_age	123 min_age	123 max_age
1	55.71	18	98

Рисунок 17 – Вывод SQL-запроса

Определим топ-5 городов, в которых проживает наибольшее количество людей:



```
SELECT
CASE
WHEN address LIKE 'г. %' THEN SUBSTRING(address FROM 4 FOR POSITION(',') IN address) - 4)
WHEN address LIKE 'д. %' THEN SUBSTRING(address FROM 4 FOR POSITION(',') IN address) - 4)
WHEN address LIKE 'с. %' THEN SUBSTRING(address FROM 4 FOR POSITION(',') IN address) - 4)
WHEN address LIKE 'п. %' THEN SUBSTRING(address FROM 4 FOR POSITION(',') IN address) - 4)
WHEN address LIKE 'к. %' THEN SUBSTRING(address FROM 4 FOR POSITION(',') IN address) - 4)
WHEN address LIKE 'клх %' THEN SUBSTRING(address FROM 6 FOR POSITION(',') IN address) - 6)
WHEN address LIKE 'ст. %' THEN SUBSTRING(address FROM 5 FOR POSITION(',') IN address) - 5)
END AS city,
COUNT(*) AS population
FROM person
WHERE deleted_at IS NULL
GROUP BY city
ORDER BY population DESC
LIMIT 5;
```

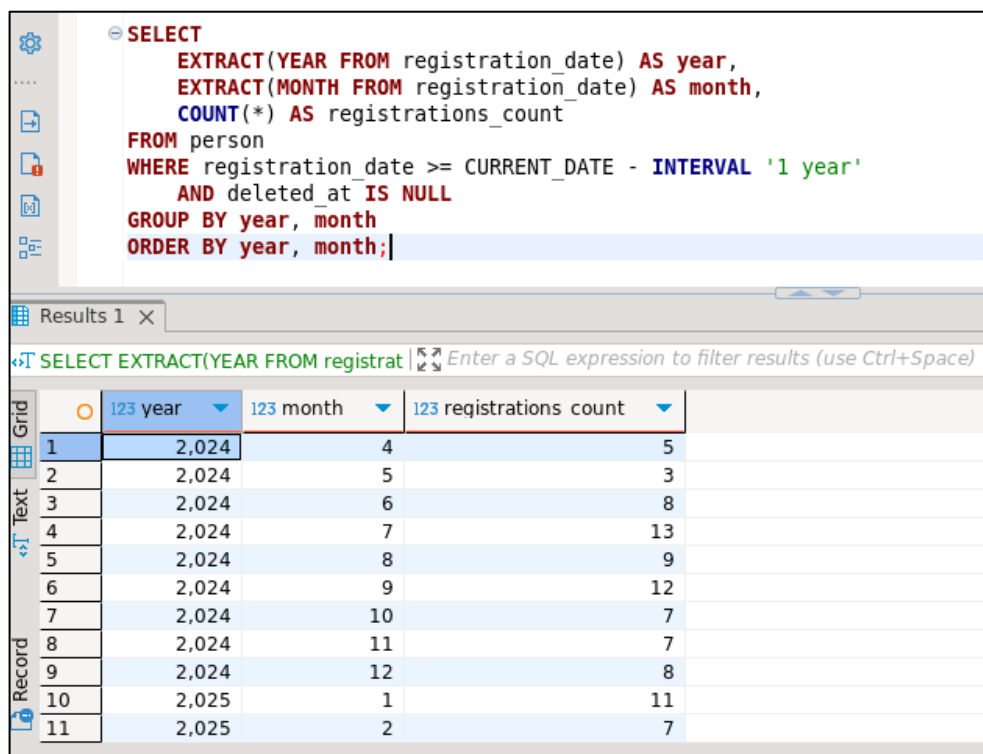
Results 1 X

SELECT CASE WHEN address LIKE 'г. %' ; Enter a SQL expression to filter results (use Ctrl+Space)

	A-Z city	123 population
1	Ставрополь	3
2	Онега	2
3	Усмань	2
4	Брянск	2
5	Печенга	2

Рисунок 18 – Вывод SQL-запроса

Найдите количество регистраций в каждом месяце за последний год.



The screenshot shows a database query editor with the following SQL query:

```
SELECT  
    EXTRACT(YEAR FROM registration_date) AS year,  
    EXTRACT(MONTH FROM registration_date) AS month,  
    COUNT(*) AS registrations_count  
FROM person  
WHERE registration_date >= CURRENT_DATE - INTERVAL '1 year'  
    AND deleted_at IS NULL  
GROUP BY year, month  
ORDER BY year, month;
```

Below the query editor, the results are displayed in a table grid. The table has 4 columns: an index column, and columns for year, month, and registrations count. The results show data for the years 2024 and 2025.

	year	month	registrations count
1	2,024	4	5
2	2,024	5	3
3	2,024	6	8
4	2,024	7	13
5	2,024	8	9
6	2,024	9	12
7	2,024	10	7
8	2,024	11	7
9	2,024	12	8
10	2,025	1	11
11	2,025	2	7

Рисунок 19 – Вывод SQL-запроса

Создадим графики для визуализации результатов анализа:

- Гистограмма распределения возраста.

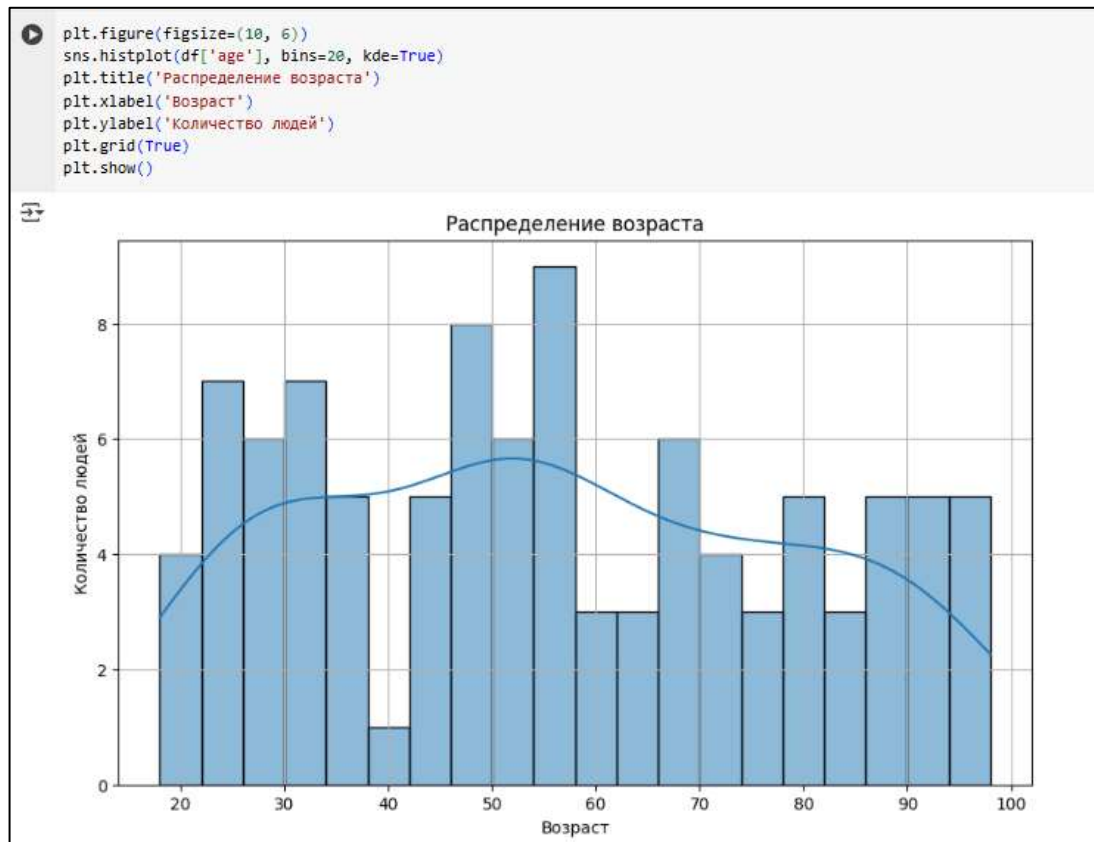


Рисунок 20 – Гистограмма распределения возраста

- Диаграмма топ-5 городов по количеству проживающих.

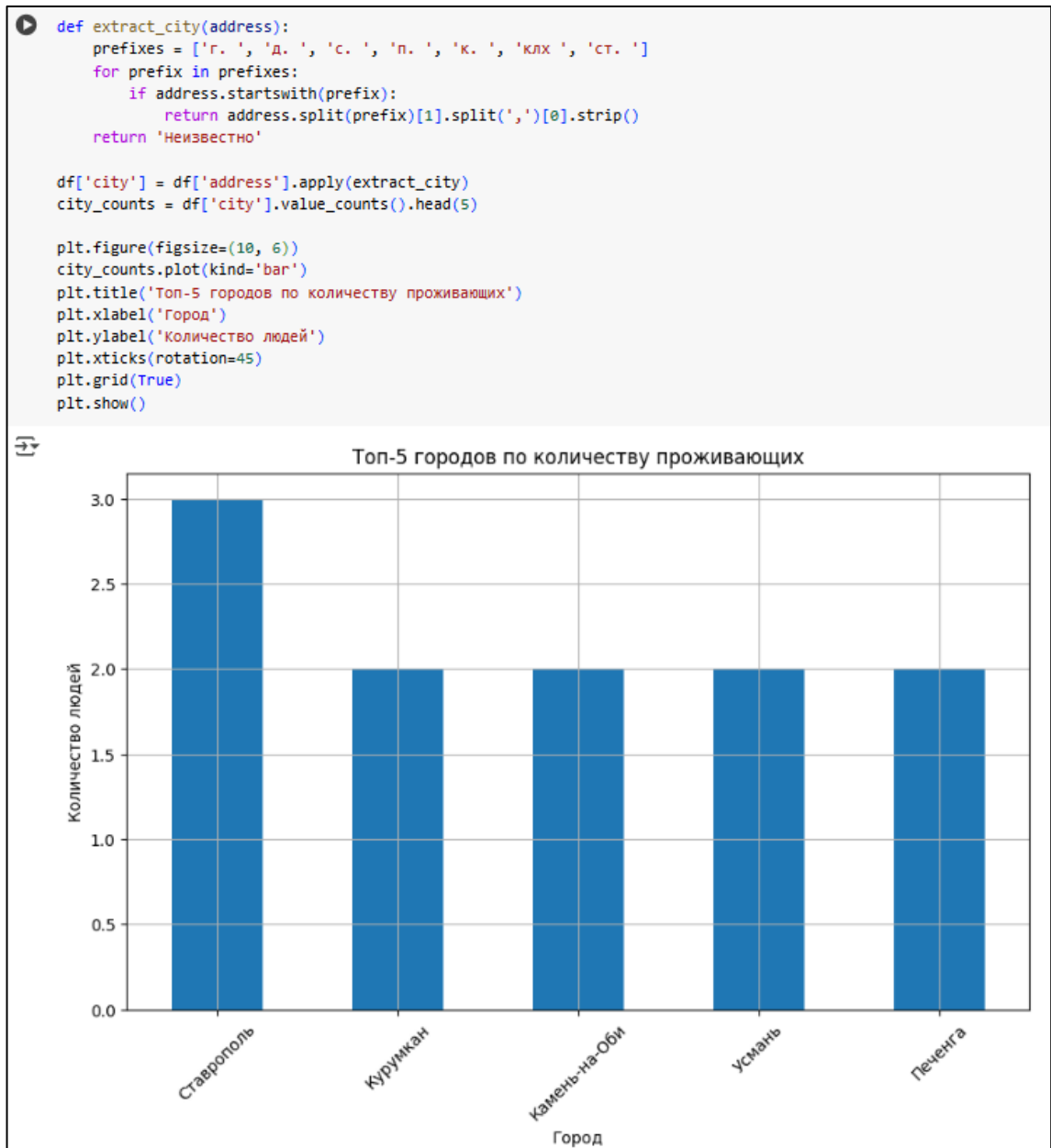


Рисунок 21 – Диаграмма топ-5 городов

- Линейный график количества регистраций по месяцам.

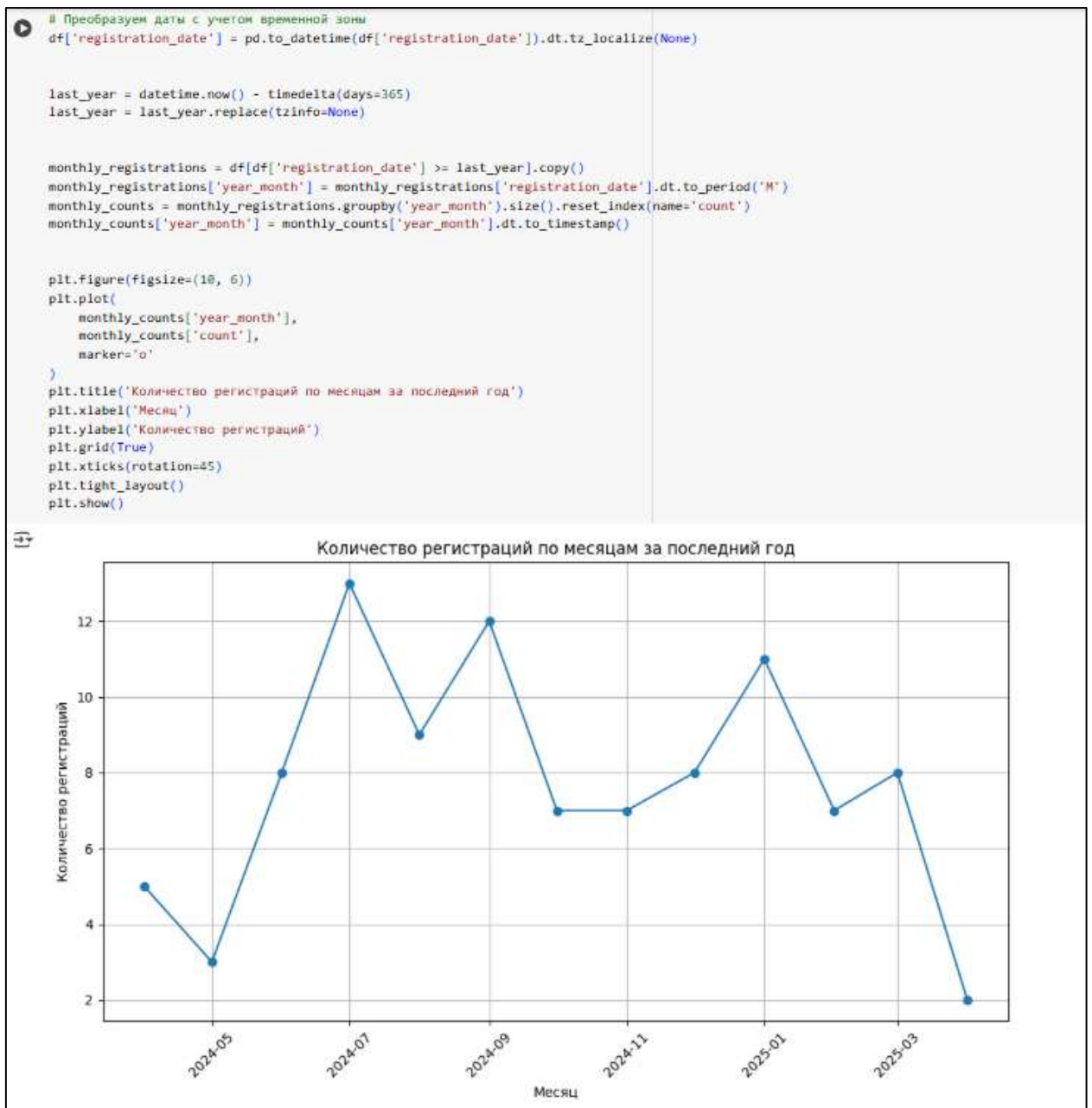


Рисунок 22 – Линейный график количества регистраций по месяцам