

Департамент образования города Москвы

Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»

Институт цифрового образования
Департамент информатики, управления и технологий

Лабораторная работа 3
по дисциплине «Проектный практикум по разработке ETL-решений»

Тема: «Практическая работа на вебинаре»

Направление подготовки 38.03.05 – бизнес-информатика
Профиль подготовки «Аналитика данных и эффективное управление»
(очная форма обучения)

Выполнила:
St_88

Москва
2025

ЗАДАНИЕ 1.

Заполнение таблицы и анализ данных в PostgreSQL с визуализацией

Цель: заполнить таблицу person в базе данных PostgreSQL фейковыми данными не менее 100 записей и провести анализ этих данных с использованием SQL. Также создать визуализации для полученных результатов.

Задачи:

1. Заполнение таблицы фейковыми данными.
 - a. Используйте библиотеку Faker для генерации фейковых данных.
 - b. Вставьте сгенерированные данные в таблицу person в базе данных PostgreSQL.
2. Анализ возраста.
 - a. Найдите средний, минимальный и максимальный возраст людей в таблице person.
3. Анализ распределения по городам.
 - a. Определите топ-5 городов, в которых проживает наибольшее количество людей.
4. Анализ регистрации.
 - a. Найдите количество регистраций в каждом месяце за последний год.
5. Визуализация данных.
 - a. Создайте графики для визуализации результатов анализа:
 - b. Гистограмма распределения возраста.
 - c. Диаграмма топ-5 городов по количеству проживающих.
 - d. Линейный график количества регистраций по месяцам.

ХОД РАБОТЫ

1. Подготовка к работе: проверка наличия запущенных контейнеров, в случае если такие есть, то их остановка и удаление:

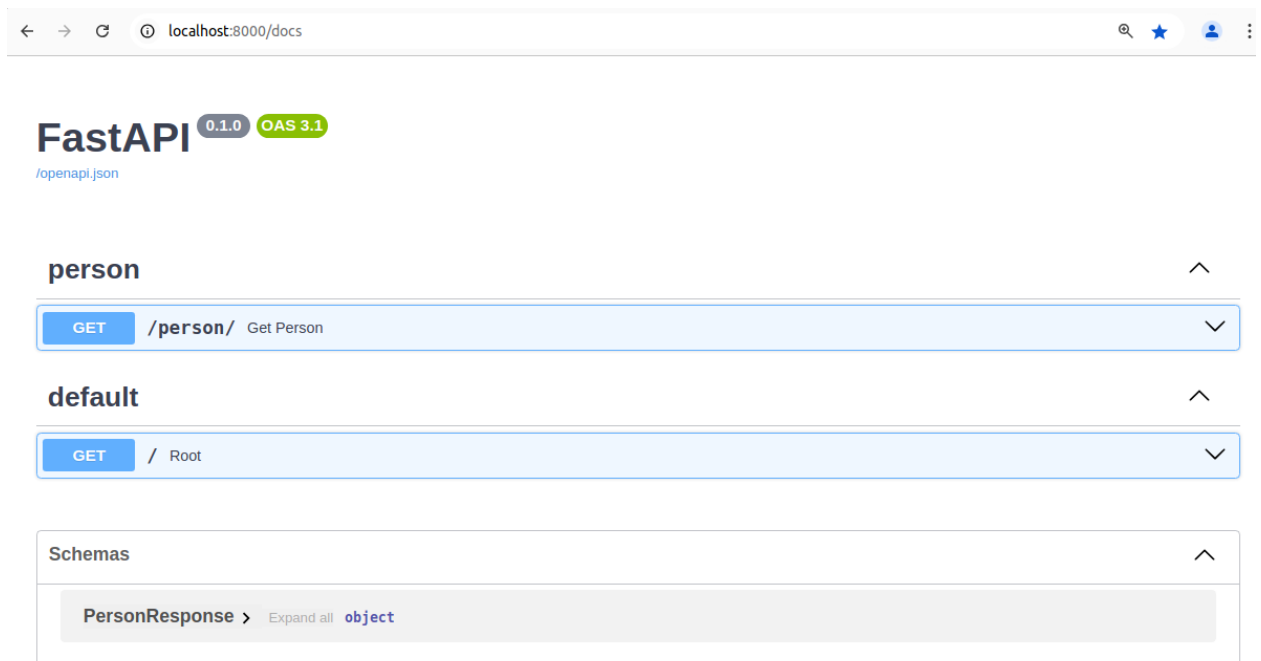
```
● dev@dev-vm:~/Downloads/lab_etl/data_for_labs/lab_airflow/lab_0_webinar$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
a5bb7206afbd   postgres:16                        "docker-entrypoint.s..." 2 weeks ago    Up 3 minutes  0.0.0.0:5432->5432
/tcp, [::]:5432->5432/tcp
ec4919b750da   dpape/pgadmin4:latest             "/entrypoint.sh"         2 weeks ago    Up 3 minutes  0.0.0.0:80->80/tcp
, [::]:80->80/tcp, 443/tcp
pgadmin
● dev@dev-vm:~/Downloads/lab_etl/data_for_labs/lab_airflow/lab_0_webinar$ docker kill $(docker ps -q)
a5bb7206afbd
ec4919b750da
● dev@dev-vm:~/Downloads/lab_etl/data_for_labs/lab_airflow/lab_0_webinar$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
```

2. Запуск контейнеров всех необходимых сервисов, а именно:

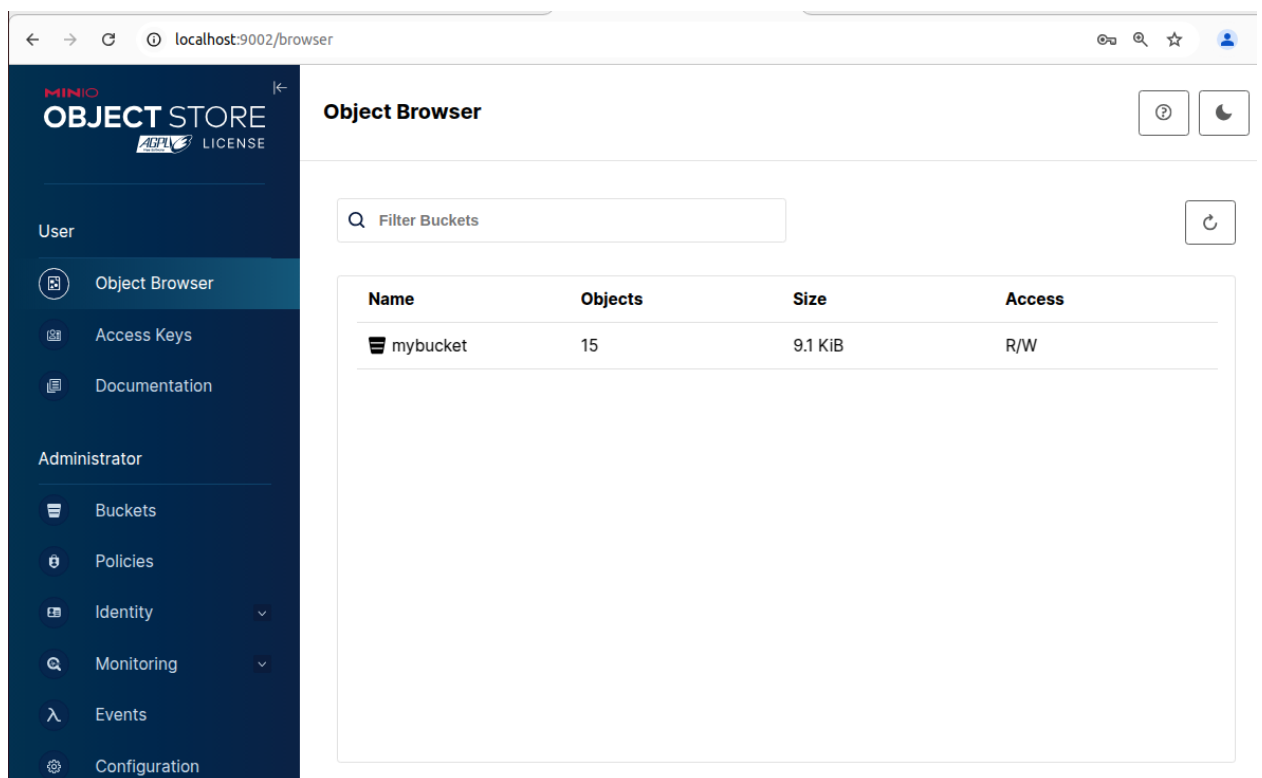
```
● dev@dev-vm:~/Downloads/lab_etl/data_for_labs/lab_airflow/lab_0_webinar$ make up-services
sudo docker compose -f docker-compose-services.yaml up -d --build
[sudo] password for dev:
Compose now can delegate build to bake for better performances
Just set COMPOSE_BAKE=true
[+] Building 4.1s (11/11) FINISHED                                docker:default
=> [faker-api internal] load build definition from Dockerfile      0.1s
=> => transferring dockerfile: 232B                                0.0s
=> [faker-api internal] load metadata for docker.io/library/python:3.12-slim 3.5s
=> [faker-api internal] load .dockerignore                        0.0s
=> => transferring context: 2B                                       0.0s
=> [faker-api 1/5] FROM docker.io/library/python:3.12-slim@sha256:aaa3f8cb64dd64e5f8cb6e58346bdcfa410a1 0.0s
=> [faker-api internal] load build context                        0.1s
=> => transferring context: 265B                                     0.0s

[+] Running 8/8
✓ faker-api                               Built                                0.0s
✓ Network lab_0_webinar_default           Created                             0.3s
✓ Volume "lab_0_webinar_minio_data"       Created                             0.0s
✓ Container zookeeper                     Started                             2.2s
✓ Container lab_0_webinar-minio-1         Started                             2.8s
✓ Container pg                             Healthy                            14.9s
✓ Container ch                             Healthy                            35.8s
✓ Container lab_0_webinar-faker-api-1     Started                             36.4s
```

Проверка доступа к Faker (localhost:8000):



Проверка доступа к Minio (localhost:9002):

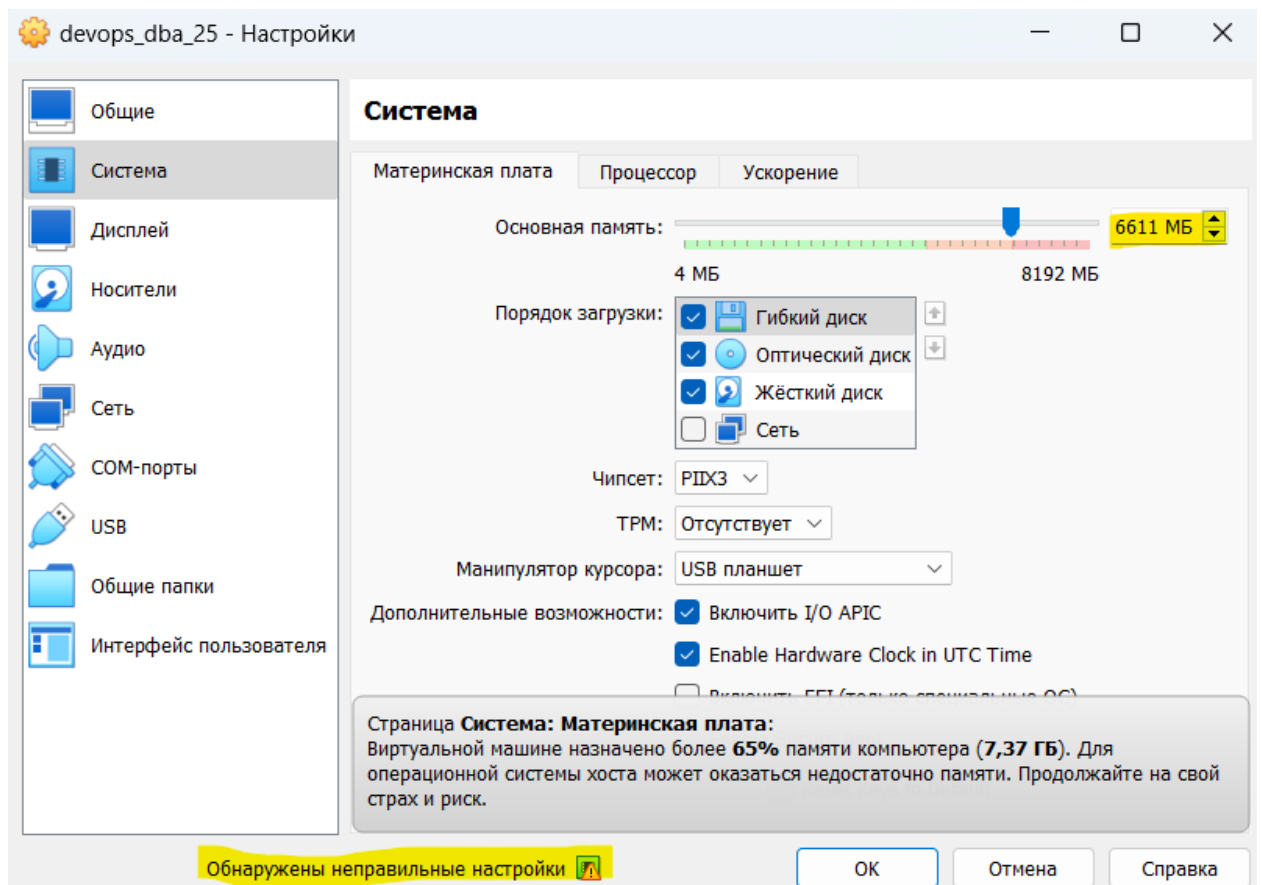


3. Далее по заданию необходимо будет сгенерировать 100 записей. Для того, чтобы сделать это автоматически, перед запуском Airflow необходимо в коде дага, отвечающего за генерацию и запись данных в базу PostgreSQL, установить расписание выполнения дага: раз в 2 минуты:

```
1 import datetime
2
3 from airflow.decorators import task, task_group
4 from airflow.models.dag import DAG
5
6
7 with DAG(
8     dag_id="simulative example basic dag",
9     schedule=datetime.timedelta(minutes=2)),
10 start_date=datetime.datetime(2025, 1, 1),
11 catchup=False,
12 tags=["simulative"],
13 ) as dag:
14
15     @task
```

Изначально, для более частой работы дага было выставлено значение: раз в 18 секунд. Однако, при попытке дальнейшего запуска дага в Airflow у виртуальной машины закончилось ОЗУ, поэтому для, хоть и медленной, но стабильной автоматической генерации было принято решение оставить значение запуска дага: раз в 2 минуты.

Максимально доступные ресурсы на ноутбуке довольно малы:



4. Запуск контейнеров, отвечающих за работу Airflow:

```

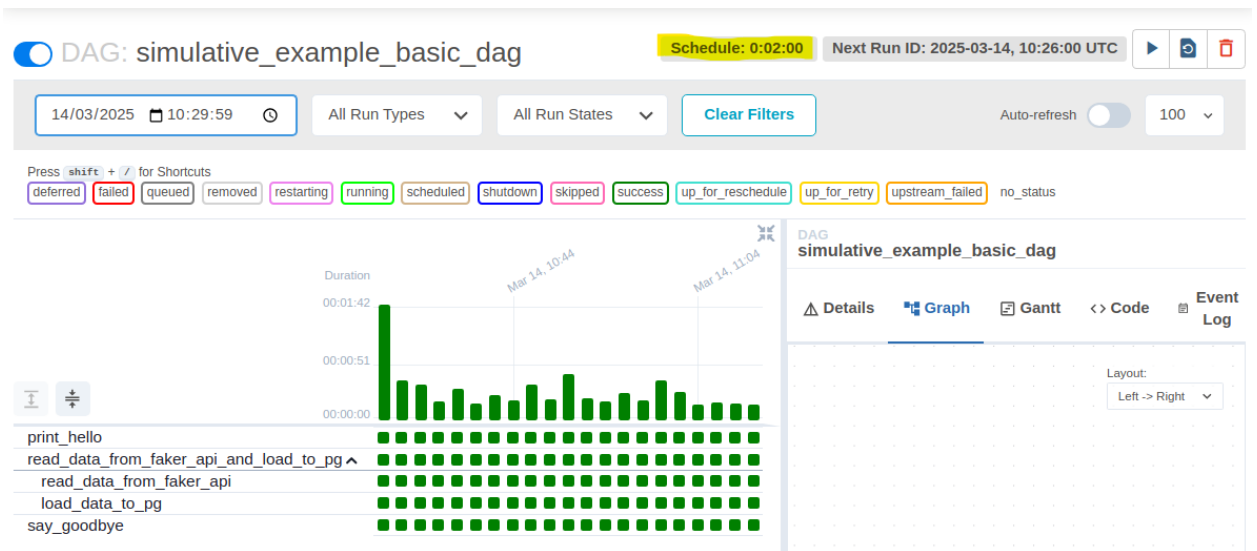
dev@dev-vm:~/Downloads/lab_etl/data_for_labs/lab_airflow/lab_0_webinar$ make up-af
sudo docker compose -f docker-compose-af.yaml up -d --build
[+] Running 48/48
  ✓ postgres Pulled 66.9s
  ✓ airflow-webserver Pulled 125.8s
  ✓ airflow-triggerer Pulled 125.8s
  ✓ airflow-worker Pulled 125.8s
  ✓ airflow-scheduler Pulled 125.8s
  ✓ airflow-init Pulled 125.8s
  ✓ redis Pulled 40.9s
WARN[0126] Found orphan containers ([lab_0_webinar-faker-api-1 ch zookeeper lab_0_webinar-minio-1 pg]) for this
project. If you removed or renamed this service in your compose file, you can run this command with the --remo
ve-orphans flag to clean it up.
[+] Running 8/8
  ✓ Volume "lab_0_webinar_postgres-db-volume" Created 0.0s
  ✓ Container lab_0_webinar-redis-1 Healthy 9.0s
  ✓ Container lab_0_webinar-postgres-1 Healthy 9.0s
  ✓ Container lab_0_webinar-airflow-init-1 Exited 38.8s
  ✓ Container lab_0_webinar-airflow-scheduler-1 Started 40.7s
  ✓ Container lab_0_webinar-airflow-triggerer-1 Started 41.0s
  ✓ Container lab_0_webinar-airflow-worker-1 Started 41.0s
  ✓ Container lab_0_webinar-airflow-webserver-1 Started 41.0s

```

Проверка доступа к Airflow (localhost:8080):

The screenshot shows the Apache Airflow web interface at localhost:8080. The top navigation bar includes links for DAGs, Cluster Activity, Datasets, Security, Browse, Admin, and Docs. The main section is titled 'DAGs' and features a filter bar with buttons for 'All' (3), 'Active' (1), and 'Paused' (2). Below the filter bar, there are buttons for 'Running' (1) and 'Failed' (0), a search bar, and an 'Auto-refresh' toggle. The DAGs table lists three DAGs: 'basic_dag', 'simulative_example_advanced_dag', and 'simulative_example_basic_dag'. The 'simulative_example_basic_dag' is selected, indicated by a blue toggle switch. The table columns are DAG, Owner, Runs, Schedule, Last Run, Next Run, and Recent Tasks. The 'simulative_example_basic_dag' has a '1' in a green circle in the 'Runs' column, indicating a successful run.

- Запуск дага, отвечающего за генерацию и запись данных в базу PostgreSQL – как видно, автоматическое включение дага раз в 2 минуты отрабатывает:



Проверка успешности записи данных в базу PostgreSQL посредством DBeaver:

select *
from public.person p;

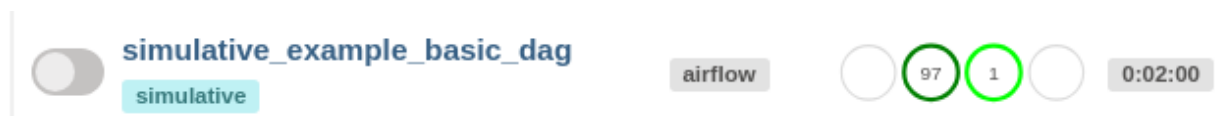
person 1 x

select * from public.person p

	id	name	age	address	email	phone number	registration date	created at
1	076fe4e6-69	Евфросиния Ф	72	к. Луховицы, бул. гавдеева@exa	8 (653) 115-71-88	5-15 02:35:47.895 +0300	5:47:38.155 +0300	13:5
2	bb56aedc-e5	Гусев Автоном	87	п. Рязань, наб. В. bronsislavpavl	82382262903	1-15 17:44:26.278 +0300	3:41:20.080 +0300	13:5
3	205803ea-fe	Вероника Арх	77	п. Каргополь, бу. mefodi52@exa	8 676 866 6753	5-15 21:24:00.001 +0300	3:03:37.353 +0300	13:5
4	3ed76bef-ef	София Леонид	68	клх Троицко-Печ. ivanovkliment	+79677542571	7-23 13:02:27.612 +0300	4:40:10.485 +0300	13:5
5	74b2f075-2f	Стойня Фокич С	32	с. Уфа, наб. Прох. kononovafekla	8 (169) 806-4753	5-11 19:17:38.368 +0300	5:31:25.819 +0300	13:5
6	04036191-cl	Прокофий Вла	92	к. Можайск, наб. agata_2001@e	+7 384 061 79 63	3-25 23:35:43.008 +0300	3:38:59.642 +0300	13:4
7	e329427f-60	Якушев Адриа	79	г. Каменск-Урал. raksenov@exa	+73489180650	1-13 02:04:05.183 +0300	1:36:18.059 +0300	13:4
8	bf1bf19a-6c	Федот Дорофе	83	клх Шелехов, ул. petrovamarg	8 217 185 1570	3-04 08:52:27.529 +0300	7:33:49.614 +0300	13:4
9	b245070a-a	Юлия Тарасов	18	г. Вилюйск, пр. i milan_97@exa	84164907068	5-30 04:53:44.263 +0300	8:27:41.758 +0300	13:4
10	dcea451e-41	Гуляев Селиве	74	с. Тимашевск, ул. ljudmila_1976	8 (506) 406-85-30	3-21 05:44:21.857 +0300	0:25:42.895 +0300	13:4
11	4cb9eccd-8c	Александра Л	65	с. Усть-Ордынский martinovprok	81466389602	1-21 03:48:37.881 +0300	2:32:07.273 +0300	13:5
12	72aeb67a-6c	Симон Вилоро	94	к. Калевала, ш. j jakovlevamarg	+7 638 582 3179	1-01 07:18:20.896 +0300	7:42:44.664 +0300	13:5
13	3cd0a5b6-5f	Евпраксия Ма	95	клх Витим, наб. tatjana_80@ex	+7 897 148 4780	3-23 07:25:15.670 +0300	3:39:48.359 +0300	13:5
14	ad2f52cf-d4	Лев Харламп	96	к. Шилка, ул. He burovandron	+71476672974	2-20 15:00:54.134 +0300	9:30:55.016 +0300	13:5
15	11d9a0ab-8	Виноградов Со	59	ст. Магас, наб. Т. averjanvoronov	8 (042) 267-28-47	3-18 16:00:11.986 +0300	8:16:30.651 +0300	13:5
16	d513ebe8-3	Ермил Власов	79	п. Кедровский, ул. arhipovaregina	+7 437 388 89 63	2-23 17:52:31.599 +0300	0:44:19.471 +0300	14:0
17	46939376-0	Агата Евгение	22	с. Тихвин, ш. Пе. arodionov@exa	8 071 551 83 02	3-12 20:51:37.163 +0300	6:21:15.131 +0300	14:0
18	6d72c917-3	Шашков Корн	85	ст. Кропоткин (r jomina@exam	+7 357 979 5635	1-09 12:45:28.683 +0300	6:02:22.394 +0300	14:0
19	99c8e4eb-2f	Мечислав Вас	41	с. Комсомольск-н. valerifedorov	81452617787	1-16 04:10:09.201 +0300	0:43:41.272 +0300	14:0
20	0a1d1b9d-b	Лобанов Арсен	19	г. Данков, бул. П. savinaljubov	88926288077	2-04 07:20:37.419 +0300	1:10:07.703 +0300	14:0

22 row(s) fetched - 0.156s (0.075s fetch), on 2025-03-14 at 14:12:32

6. 97 записей было занесено в таблицу, что составило 194 минуты работы дага

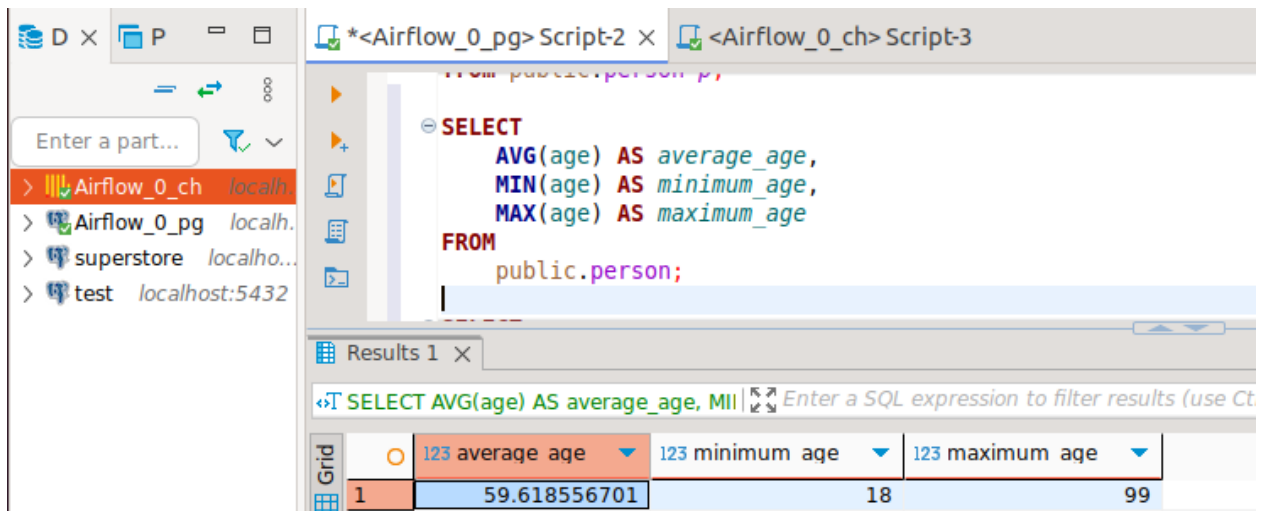


7. На сгенерированных данных необходимо найти средний, минимальный и максимальный возраст людей в таблице person. Скрипт SQL:

```

SELECT
    AVG(age) AS average_age,
    MIN(age) AS minimum_age,
    MAX(age) AS maximum_age
FROM
    public.person;

```

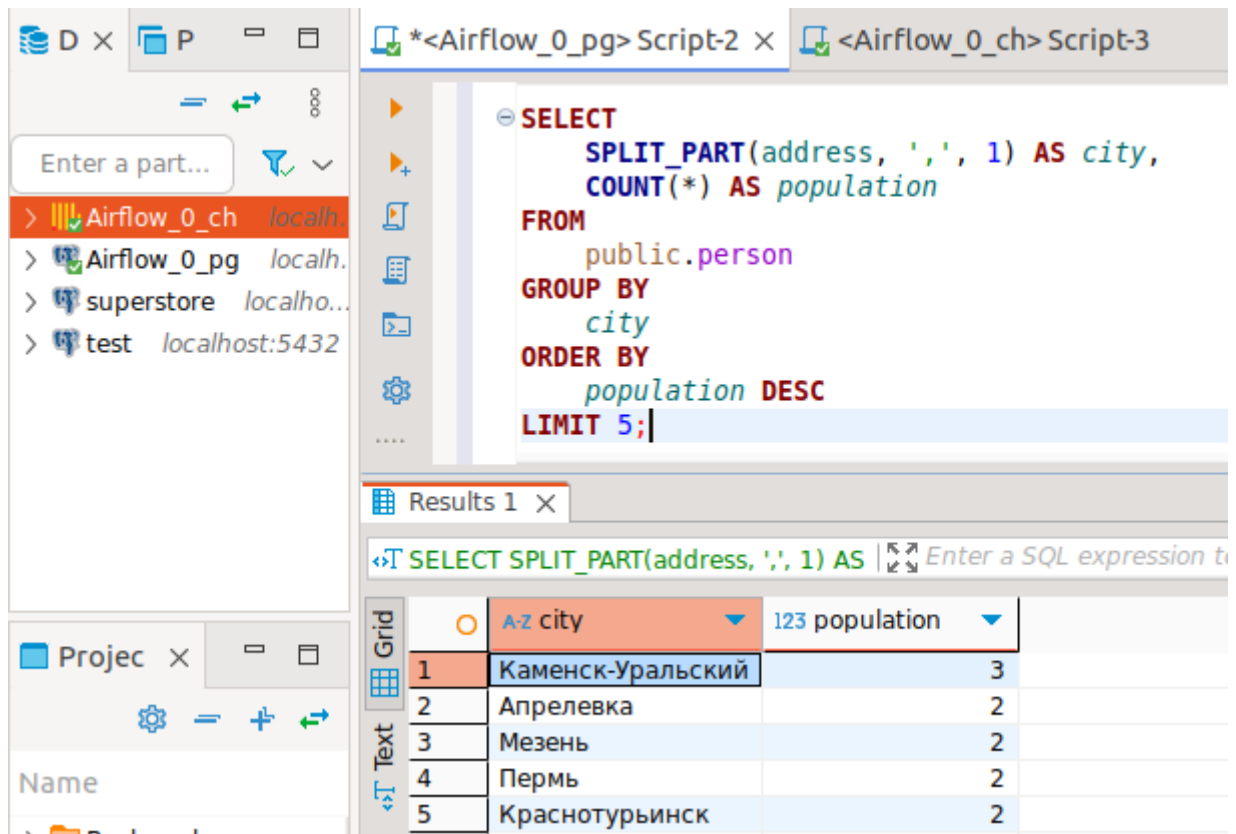


8. Далее необходимо определить топ-5 городов, в которых проживает наибольшее количество людей. Скрипт SQL:

```

SELECT
    TRIM(SPLIT_PART(address, ',', 1), 'к. ') AS city,
    COUNT(*) AS population
FROM
    public.person
GROUP BY
    city
ORDER BY
    population DESC
LIMIT 5;

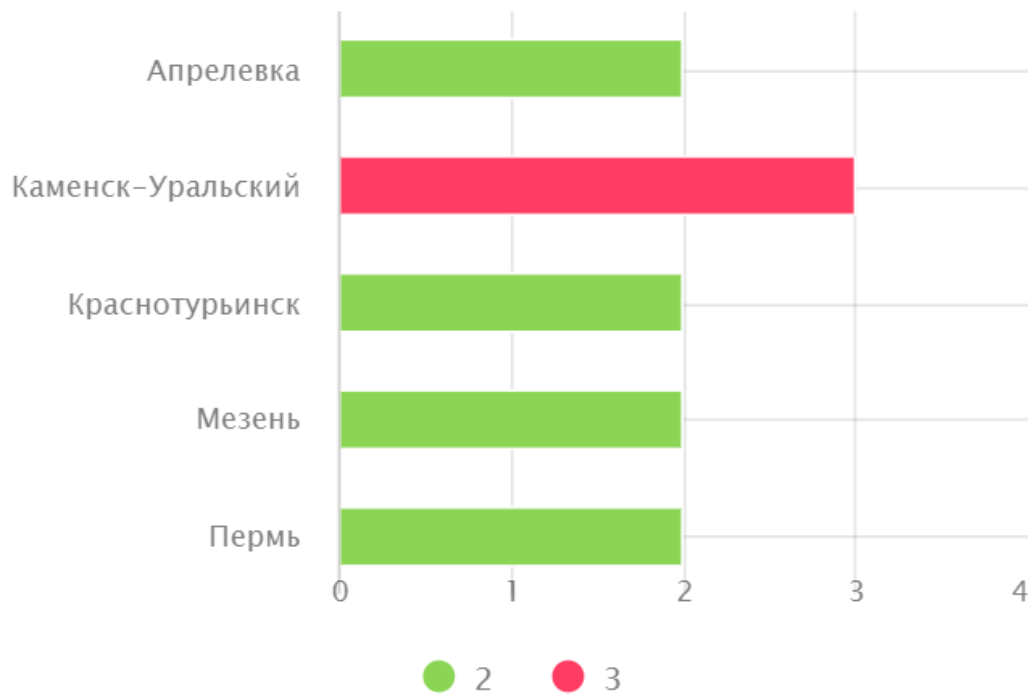
```

9. Определение количества регистраций в каждом месяце за последний год. Скрипт SQL:

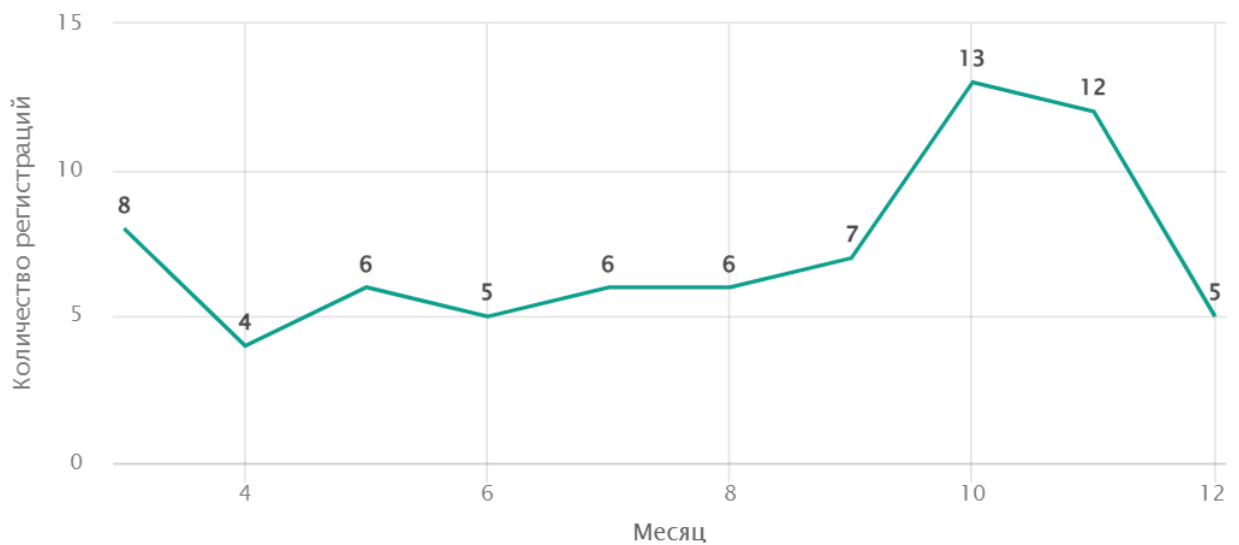
```
SELECT
    EXTRACT(YEAR FROM registration_date) AS year,
    EXTRACT(MONTH FROM registration_date) AS month,
    COUNT(*) AS registration_count
FROM
    public.person
WHERE
    EXTRACT(YEAR FROM registration_date) = '2024'
GROUP BY
    year, month
ORDER BY
    month;
```


Топ-5 городов по числу проживающих ...



- Линейный график количества регистраций по месяцам.

Количество регистраций по месяцам за последний год (2024) ...



Для этого выгрузим файлы результатов обработки каждого скрипта и визуализируем их с помощью YandexDataLens, потому что там быстро (а время ограничено) и удобно.

Ссылка на дашборд: <https://datalens.yandex/25z0zok7axh8o>