

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 6.1. «Разработка полного ETL-процесса.  
Оркестровка конвейера данных»**

**по дисциплине «Проектный практикум по разработке ETL-решений»**

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

Выполнила:  
St\_88

Москва  
2025

# ВАРИАНТ 1

Номер Задание	Описание
1	Получить данные за последний час для сайта <b>Yandex</b>
	Скачайте данные за последний час для страницы Yandex и сохраните их в базе данных. После этого, напишите SQL-запрос для подсчета среднего числа просмотров по часам и визуализируйте данные на графике.

## ХОД РАБОТЫ

### 1. Предварительная работа

1.1. По условию задачи необходимо отобразить данные за последний час. В ходе предварительного просмотра данных с использованием терминальной строки и `wget` было выяснено, что новейший доступный период — это 17:00 4 апреля 2025:

```
dev@dev-vm:~/Downloads$ wget https://dumps.wikimedia.org/other/pageviews/2025/2025-04/pageviews-20250404-190000.gz
--2025-04-04 21:54:35-- https://dumps.wikimedia.org/other/pageviews/2025/2025-04/pageviews-20250404-190000.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
2025-04-04 21:54:35 ERROR 404: Not Found.

dev@dev-vm:~/Downloads$ wget https://dumps.wikimedia.org/other/pageviews/2025/2025-04/pageviews-20250404-180000.gz
--2025-04-04 21:55:25-- https://dumps.wikimedia.org/other/pageviews/2025/2025-04/pageviews-20250404-180000.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
2025-04-04 21:55:25 ERROR 404: Not Found.
```

На скриншотах выше видно, что данных за 19:00 и 18:00 пока еще нет.

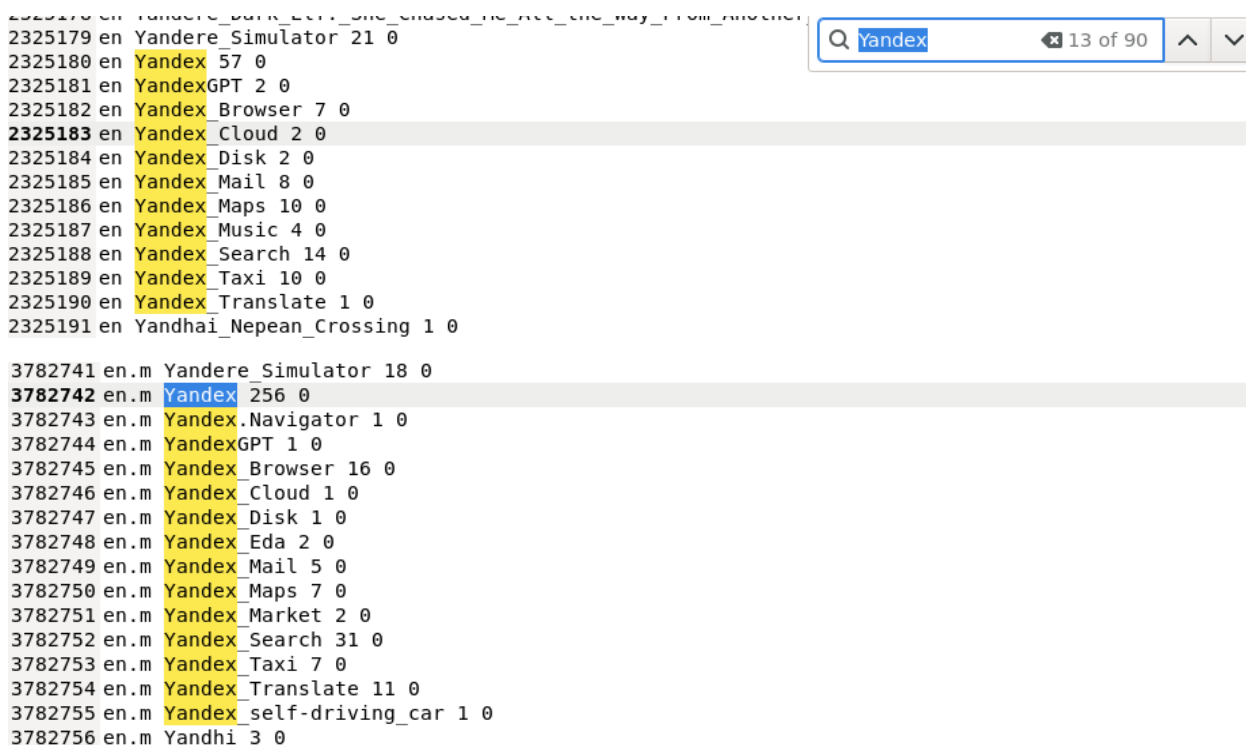
```
dev@dev-vm:~/Downloads$ wget https://dumps.wikimedia.org/other/pageviews/2025/20
25-04/pageviews-20250404-170000.gz
--2025-04-04 22:24:41-- https://dumps.wikimedia.org/other/pageviews/2025/2025-0
4/pageviews-20250404-170000.gz
Resolving dumps.wikimedia.org (dumps.wikimedia.org)... 208.80.154.71, 2620:0:861
:3:208:80:154:71
Connecting to dumps.wikimedia.org (dumps.wikimedia.org)|208.80.154.71|:443... co
nnected.
HTTP request sent, awaiting response... 200 OK
Length: 63802094 (61M) [application/octet-stream]
Saving to: 'pageviews-20250404-170000.gz'

pageviews-20250404- 100%[=====>] 60,85M 3,59MB/s in 72s

2025-04-04 22:25:53 (870 KB/s) - 'pageviews-20250404-170000.gz' saved [63802094/
63802094]
```

А вот при попытке выгрузить данные за 17:00 – они имеются. Поэтому далее при выполнении работы будет взят именно этот период.

1.2. Также при предварительном изучении данных было выяснено, что связанных с Яндексом страниц довольно много.



2325179 en Yandere\_Simulator 21 0  
2325180 en Yandex 57 0  
2325181 en YandexGPT 2 0  
2325182 en Yandex Browser 7 0  
2325183 en Yandex Cloud 2 0  
2325184 en Yandex Disk 2 0  
2325185 en Yandex Mail 8 0  
2325186 en Yandex Maps 10 0  
2325187 en Yandex Music 4 0  
2325188 en Yandex Search 14 0  
2325189 en Yandex Taxi 10 0  
2325190 en Yandex Translate 1 0  
2325191 en Yandhai\_Nepean\_Crossing 1 0

3782741 en.m Yandere\_Simulator 18 0  
3782742 en.m Yandex 256 0  
3782743 en.m Yandex.Navigator 1 0  
3782744 en.m YandexGPT 1 0  
3782745 en.m Yandex Browser 16 0  
3782746 en.m Yandex Cloud 1 0  
3782747 en.m Yandex Disk 1 0  
3782748 en.m Yandex Eda 2 0  
3782749 en.m Yandex Mail 5 0  
3782750 en.m Yandex Maps 7 0  
3782751 en.m Yandex Market 2 0  
3782752 en.m Yandex Search 31 0  
3782753 en.m Yandex Taxi 7 0  
3782754 en.m Yandex Translate 11 0  
3782755 en.m Yandex\_self-driving\_car 1 0  
3782756 en.m Yandhi 3 0

Отберем наиболее знакомые («на слуху») сервисы Яндекса, а именно:

— Яндекс;

- ЯндексГПТ;
- Яндекс Карты;
- Яндекс Такси;
- Яндекс Маркет;
- Поиск Яндекса

## 2. Изменение структуры DAGa

2.1. Указываем дату, за которую нам необходимы данные по условию индивидуального задания – 4 апреля 2025 года. Далее, т.к. выгрузив данные за один час, будет сложно провести какой-либо анализ, то прописываем условие, чтобы данные выгружались за весь день 4 апреля 2025 года крайнего доступного часа (до 17:00):

```
def generate_get_data_tasks(dag):  
    year = 2025  
    month = 4  
    day = 4  
  
    tasks = []  
    for hour in range(17):  
        output_path = f"/tmp/wikipeviews-{hour:0>2}.gz"  
        task = PythonOperator(  
            task_id=f"get_data_{hour:0>2}",  
            python_callable=_get_data,  
            op_kwargs={  
                "year": year,  
                "month": month,  
                "day": day,  
                "hour": hour,  
                "output_path": output_path,  
            },  
            dag=dag,  
            retries=3,  
            retry_delay=timedelta(minutes=2),  
            retry_exponential_backoff=True,  
        )  
        tasks.append(task)  
    return tasks
```

```
get_data_tasks = generate_get_data_tasks(dag)
```

Как видно, был применен цикл, то есть далее будут создаваться отдельные таски для выгрузки данных за каждый час из периода (т.к. по-другому не сработало).

2.2. Был изменен таск агрегации данных так, чтобы, во-первых, выводились данные только по англоязычным и русскоязычным доменам, и, во-вторых, чтобы в целевую таблицу также записывались домен страницы и не дата записи данных в таблицу, а именно час и день, за который представлены данные.

В ходе выполнения попыток, также столкнулась с проблемой, что в целевую таблицу данные «перезаписываются», то есть по итогу выполнения ОАГа всегда в результат записывались данные только за последний час из периода. Эта проблема также была решена.

```
def _fetch_pageviews(pagenames, execution_date, **context):
    # Изменяем структуру хранения данных: {(domain, page, hour): views}
    result = defaultdict(int)

    for hour in range(17):
        filename = f"/tmp/wikipageviews-{hour:0>2}"
        try:
            data_datetime = datetime(2025, 4, 4, hour)

            with open(filename, "r") as f:
                for line in f:
                    parts = line.strip().split()
                    if len(parts) < 4:
                        continue

                    domain_code, page_title, view_counts = parts[0], parts[1], parts[2]

                    if domain_code in ["en", "en.m", "ru", "ru.m"] and page_title in pagenames:
                        # Ключ теперь включает час для разделения периодов
                        result[(domain_code, page_title, hour)] += int(view_counts)

        except FileNotFoundError as e:
            print(f"Error processing file {filename}: {e}")
            continue

    with open("/tmp/postgres_query.sql", "w") as f:
        for (domain, pagename, hour), pageviewcount in result.items():
            hour_datetime = datetime(2025, 4, 4, hour)
            f.write(
                f"INSERT INTO pageview_counts (domain_name, page_name, view_count, data_period) VALUES ("
                f"''{domain}'', ''{pagename.replace('\"', '\"')}'', {pageviewcount}, "
                f"''{hour_datetime.isoformat()}::timestamp"
                f"');\n"
            )
```

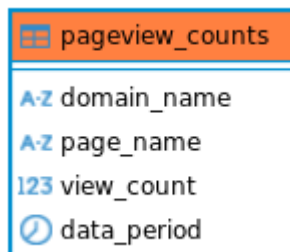
### 2.3. Указываем наименования необходимых страниц:

```
fetch_pageviews = PythonOperator(  
    task_id="fetch_pageviews",  
    python_callable=_fetch_pageviews,  
    op_kwargs={  
        "pagenames": {  
            "Yandex", "YandexGPT", "Yandex_Maps",  
            "Yandex_Taxi", "Yandex_Market", "Yandex_Search"  
        }  
    },  
    provide_context=True,  
    dag=dag,  
    retries=3,  
    retry_delay=timedelta(minutes=1),  
)
```

2.4. Также, т.к. мы меняем структуру целевой таблицы, то был обновлен SQL-запрос для ее создания:

```
scripts > create_table.sql  
1 CREATE TABLE pageview_counts (  
2     domain_name VARCHAR(50) NOT NULL,  
3     page_name VARCHAR(50) NOT NULL,  
4     view_count INT NOT NULL,  
5     data_period TIMESTAMP NOT NULL  
6 );  
7
```

Диаграмма полученной таблицы:



## 3. Запуск контейнеров и выполнение DAGa:

3.1. После сохранения всех изменений билдим и запускаем контейнеры:

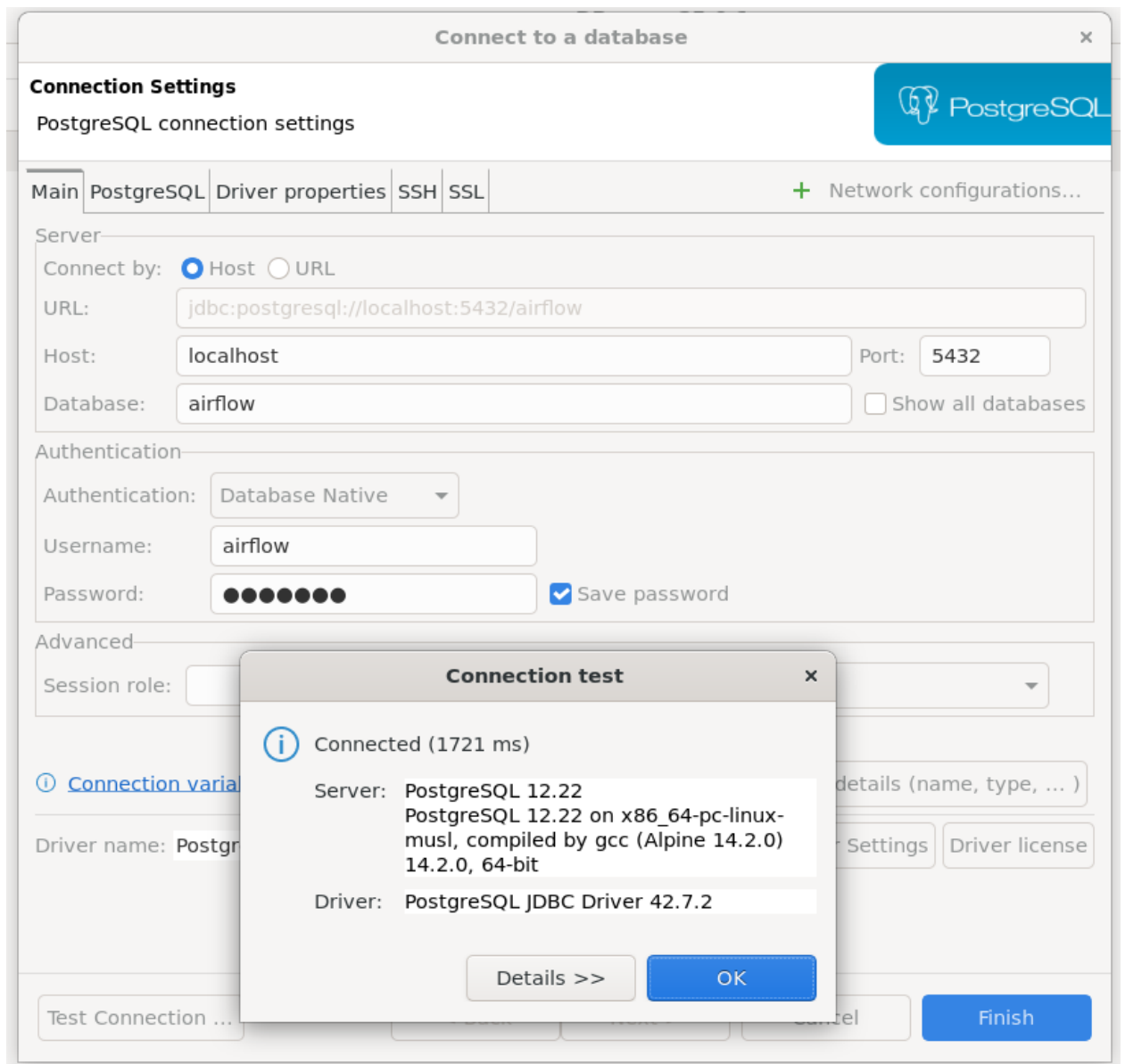
```

● dev@dev-vm:~/business_case_stocksense_25$ sudo docker build -t custom-airflow:slim-2.8.1-python3.1
1 .
[sudo] password for dev:
2025/04/04 22:45:41 in: []string{}
2025/04/04 22:45:41 Parsed entitlements: []
[+] Building 5.2s (7/7) FINISHED

                                                                    docker:default
                                                                    -----
○ dev@dev-vm:~/business_case_stocksense_25$ sudo docker compose up --build
[+] Running 13/13
  ✓ wiki_results Pulled                                         28.5s
  ✓ 1f3e46996e29 Pull complete                                  4.4s
  ✓ 47e20ba03731 Pull complete                                  4.8s
  ✓ 101b82465a4f Pull complete                                  5.6s
  ✓ 319529a7ccb0 Pull complete                                  5.9s
  ✓ c2f9392cfd4c Pull complete                                  6.3s
  ✓ 4e04446ce95d Pull complete                                  21.0s
  ✓ 47bfe778b869 Pull complete                                  21.0s

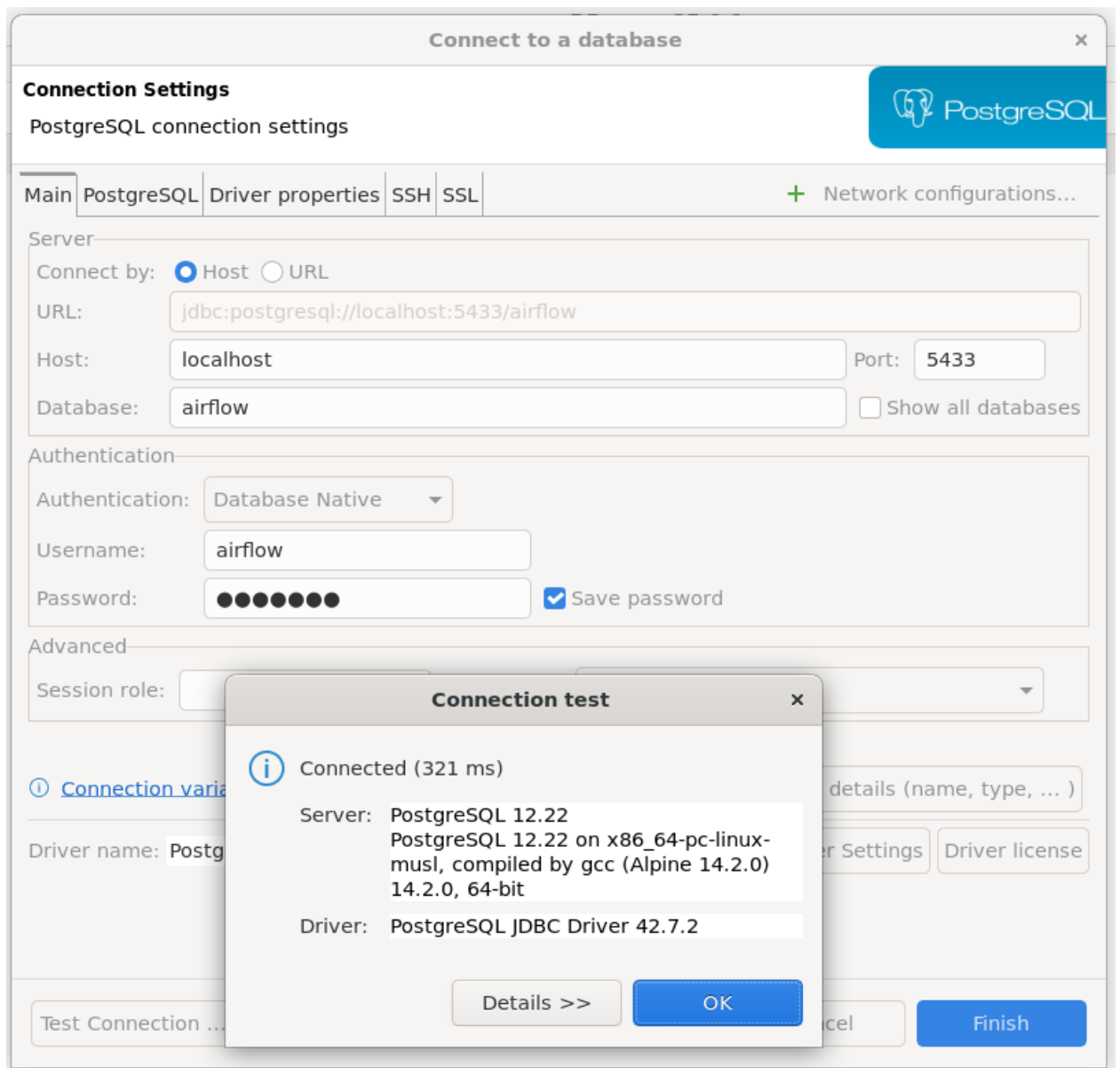
```

3.2. В DBeaver создаем новое подключение к внутренней базе данных Airflow на порту 5432:



3.3. Далее создаем второе подключение к базе данных на порту 5433 – для выгрузки конечного итогового результата:





3.4. Далее переходим к работе в Airflow. Проверяем, что интерфейс доступен на порту 8080. И сразу создаем новое подключение к Postgres, чтобы при отработке дага данные успешно записывались в целевую таблицы базы данных:

localhost:8080/connection/add

Airflow DAGs Cluster Activity Datasets Security Browse Admin Docs

20:06 UTC AU

Connection Id *	my_postgres
Connection Type *	Postgres Connection Type missing? Make sure you've installed the corresponding Airflow Provider Package.
Description	
Host	business_case_stocksense_25-wiki_results-1
Schema	airflow
Login	airflow
Password	
Port	
	<pre>{}</pre>

3.5. Запускаем даг. Как было сказано ранее, то для парсинга данных за каждый час из периода с использованием цикла были созданы отдельные задачи. Это может пугать, но это работает!

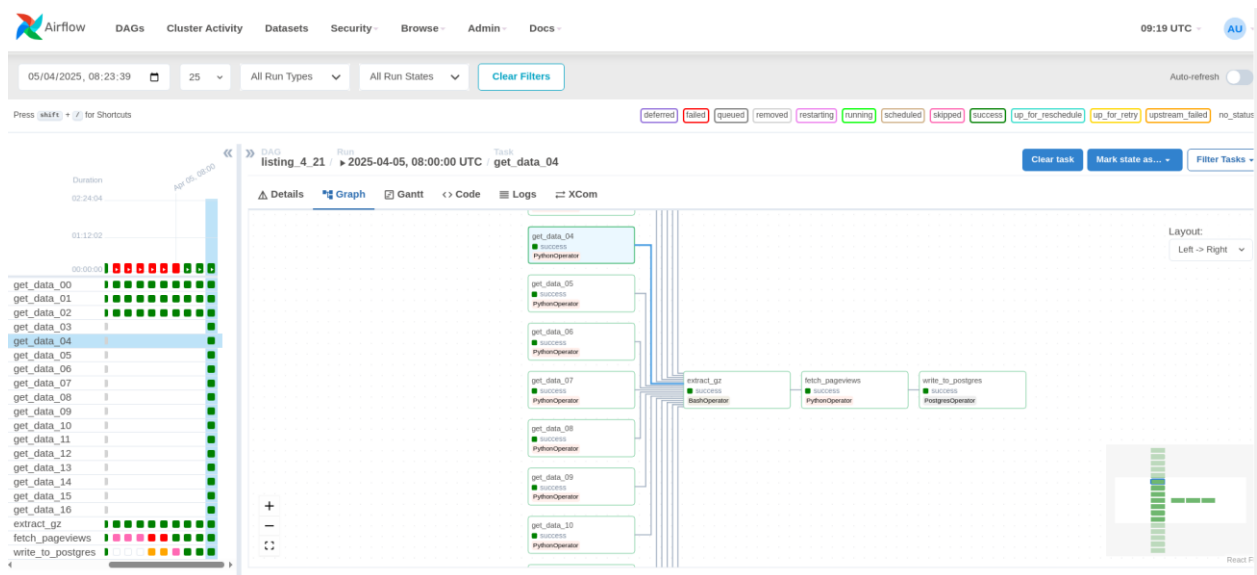


Диаграмма Ганта полученного DAGa:



Время выполнения всех задач составило 17 минут.

3.6. Проверяем в DBeaver, что данные успешно записались в целевую таблицу:

The screenshot shows a Database Navigator on the left with the 'airflow' database selected, showing a tree view of schemas and tables. The 'pageview\_counts' table is highlighted. On the right, the 'Data' tab shows a table view of the 'pageview\_counts' table with columns: domain name, page name, view count, and data period. The table contains 20 rows of data, showing various domains and page names with their respective view counts and timestamps.

	domain name	page name	view count	data period
170	ru	YandexGPT	6	5-04-04 13:00:00.000
171	ru.m	YandexGPT	3	5-04-04 13:00:00.000
172	en	Yandex	33	5-04-04 14:00:00.000
173	en	YandexGPT	2	5-04-04 14:00:00.000
174	en	Yandex_Maps	13	5-04-04 14:00:00.000
175	en	Yandex_Market	1	5-04-04 14:00:00.000
176	en	Yandex_Search	9	5-04-04 14:00:00.000
177	en.m	Yandex	142	5-04-04 14:00:00.000
178	en.m	Yandex_Maps	8	5-04-04 14:00:00.000
179	en.m	Yandex_Market	1	5-04-04 14:00:00.000
180	en.m	Yandex_Search	26	5-04-04 14:00:00.000
181	en.m	Yandex_Taxi	2	5-04-04 14:00:00.000
182	ru	YandexGPT	5	5-04-04 14:00:00.000
183	ru.m	Yandex	1	5-04-04 14:00:00.000
184	ru.m	YandexGPT	6	5-04-04 14:00:00.000
185	en	Yandex	61	5-04-04 15:00:00.000
186	en	YandexGPT	1	5-04-04 15:00:00.000
187	en	Yandex_Maps	15	5-04-04 15:00:00.000
188	en	Yandex_Market	3	5-04-04 15:00:00.000
189	en	Yandex_Search	5	5-04-04 15:00:00.000
190	en	Yandex_Taxi	4	5-04-04 15:00:00.000
191	en.m	Yandex	200	5-04-04 15:00:00.000
192	en.m	Yandex_Maps	3	5-04-04 15:00:00.000
193	en.m	Yandex_Market	2	5-04-04 15:00:00.000
194	en.m	Yandex_Search	49	5-04-04 15:00:00.000
195	en.m	Yandex_Taxi	4	5-04-04 15:00:00.000
196	ru	YandexGPT	9	5-04-04 15:00:00.000
197	ru.m	YandexGPT	6	5-04-04 15:00:00.000
198	en	Yandex	41	5-04-04 16:00:00.000
199	en	Yandex_Maps	12	5-04-04 16:00:00.000
200	en	Yandex_Market	2	5-04-04 16:00:00.000
201	en	Yandex_Search	8	5-04-04 16:00:00.000
202	en	Yandex_Taxi	1	5-04-04 16:00:00.000
203	en.m	Yandex	258	5-04-04 16:00:00.000
204	en.m	Yandex_Maps	5	5-04-04 16:00:00.000
205	en.m	Yandex_Market	1	5-04-04 16:00:00.000
206	en.m	Yandex_Search	46	5-04-04 16:00:00.000
207	en.m	Yandex_Taxi	9	5-04-04 16:00:00.000

## 4. Создание SQL-запросов

4.1. По условию индивидуального задания необходимо написать SQL-запрос для подсчета среднего числа просмотров по часам:

```
SELECT
    EXTRACT(HOUR FROM data_period) AS hour_of_day,
    ROUND(AVG(view_count), 2) AS avg_views,
    SUM(view_count) AS total_views
FROM
    pageview_counts
GROUP BY
    hour_of_day
ORDER BY
    avg_views desc
limit 10;
```

Результат выполнения запроса:

```
SELECT
    EXTRACT(HOUR FROM data_period) AS hour_of_day,
    ROUND(AVG(view_count), 2) AS avg_views,
    SUM(view_count) AS total_views
FROM
    pageview_counts
GROUP BY
    hour_of_day
ORDER BY
    avg_views desc
limit 10;
```

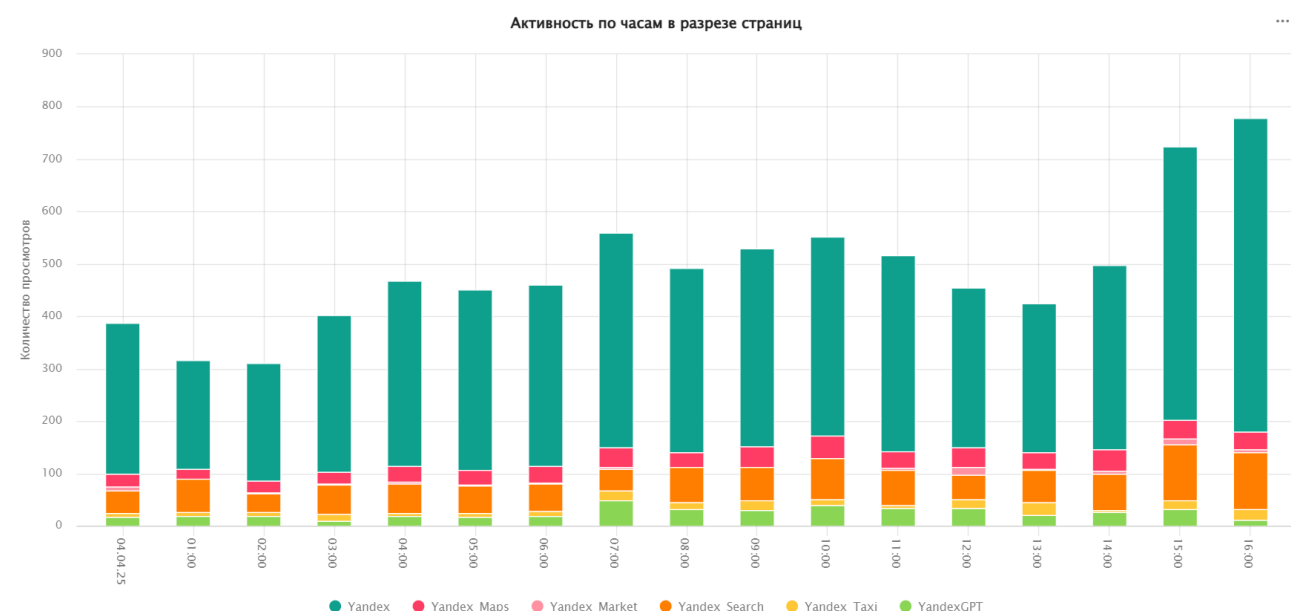
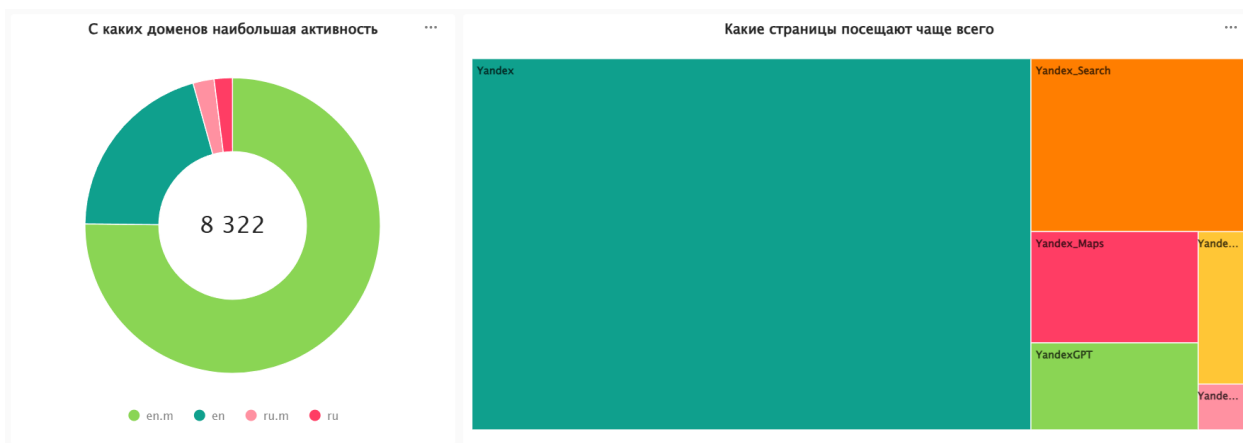
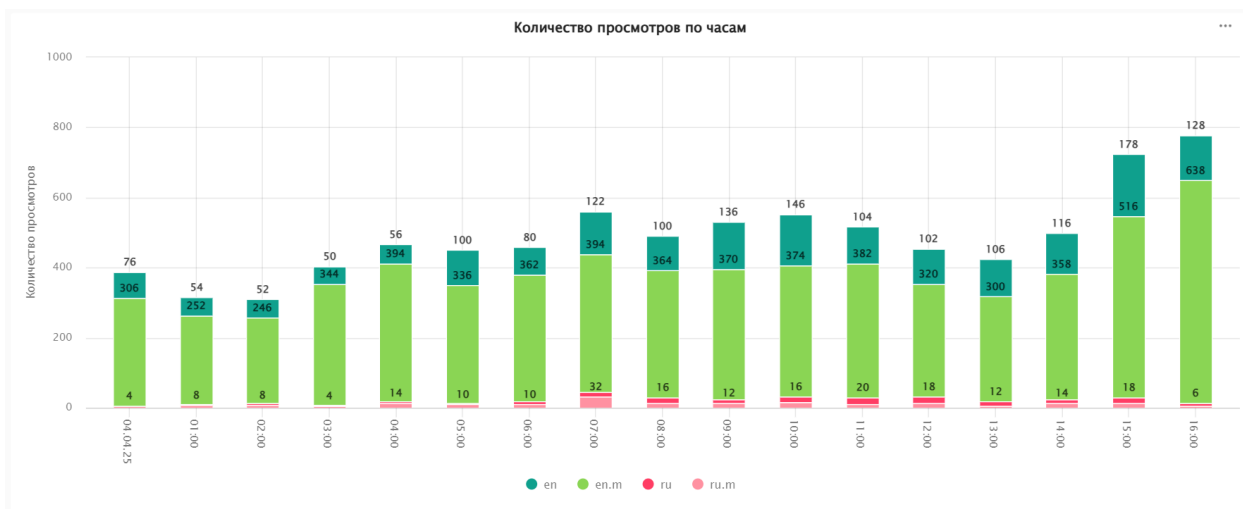
123 hour of day	123 avg views	123 total views
16	32.42	389
15	27.85	362
6	23	230
9	22.08	265
7	21.54	280
4	21.27	234
10	21.23	276
8	20.5	246
11	19.85	258
14	19.15	249

Таким образом, можно сделать вывод, что наибольшая активность пользователей к сервисам Яндекса проявляется в 15:00 - 16:00. На втором месте по среднему количеству просмотров в час находится утренний промежуток (6:00 – 10:00). То есть, если стоит задача увеличить охват аудитории или ее интерес к продуктам компании посредством размещения рекламы на сервисах Яндекса, то целесообразно размещать ее именно в утреннем промежутке или вечером.

## 5. Визуализация результата

Все данные целевой таблицы были выгружены в файл CSV. Далее этот файл был обработан в сервисе Yandex Datalens, т.к. это наиболее удобный для меня инструмент (и с широким перечнем возможностей).

Полученный Дашборд: <https://datalens.yandex.cloud/t01y0bhyvop6f>



## Общие выводы из результатов

### 1. Общие тенденции

Популярные сервисы: Наибольшее количество переходов наблюдается на главную страницу Яндекса и сервис поиска, особенно в мобильной версии. Это указывает на высокую активность пользователей в этих разделах.

Мобильный трафик: Количество переходов в мобильной версии значительно выше, чем в десктопной. Например, в 16:00 переходы на en.m/Yandex составили 258, тогда как на en/Yandex — только 41. Это подчеркивает важность мобильного трафика.

Пиковые часы: Активность пользователей возрастает в дневные часы (с 10:00 до 16:00), достигая пика в 16:00. Это оптимальное время для размещения рекламы.

### 2. Анализ по сервисам

Yandex\_Search: Высокий трафик, особенно в мобильной версии (до 49 переходов в 15:00). Рекомендуется размещать рекламу здесь для максимального охвата.

Yandex\_Maps: Стабильный трафик, но менее популярный, чем поиск. Может быть полезен для локального таргетинга.

YandexGPT: Низкий трафик, но стабильный интерес в русскоязычном сегменте (ru и ru.m). Возможно, стоит рассмотреть для нишевой рекламы.

Yandex\_Taxi: Умеренный трафик, с пиками в утренние и вечерние часы. Подходит для рекламы, связанной с транспортом или услугами.

### 3. Особенности аудитории

Англоязычный сегмент (en, en.m): Основной трафик сосредоточен здесь, особенно в мобильной версии. Рекомендуется ориентироваться на эту аудиторию.

Русскоязычный сегмент (ru, ru.m): Меньший трафик, но стабильный интерес к YandexGPT. Если компания ориентирована на русскоязычную аудиторию, стоит рассмотреть этот сегмент.

#### 4. Рекомендации для размещения рекламы

##### Приоритетные сервисы:

- Yandex\_Search (особенно мобильная версия) — для максимального охвата.
- Главная страница Yandex — для широкой аудитории.

##### Время размещения:

- Дневные часы (10:00–16:00), когда активность пользователей наиболее высока.
- Утренние часы (07:00–09:00) для таргетинга на аудиторию, использующую Yandex\_Taxi.