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

Кузьмина Дарья Юрьевна БД-241м

Программные средства сбора, консолидации и аналитики данных

Л+П № 4. Разработка аналитического дашборда для бизнес-кейса

Вариант 11

Направление подготовки/специальность
38.04.05 - Бизнес-информатика
Бизнес-аналитика и большие данные
(очная форма обучения)

Руководитель дисциплины:
Босенко Т.М., доцент департамента
информатики, управления и технологий,
доктор экономических наук

Москва
2025

Содержание

Введение

Цель освоить на практике полный цикл аналитики данных для решения прикладной бизнес-задачи. Научиться выстраивать сквозной data-конвейер, включающий автоматизацию сбора и обработки данных с помощью **Apache Airflow**, проектирование аналитической витрины в **PostgreSQL** для подготовки данных к анализу, и разработку интерактивного
дашборда в **Apache Superset** для визуального исследования данных и формулирования бизнес-инсайтов.
ПО**:**

- Система контейнеризации Docker и Docker Compose.
- **Apache Airflow** с провайдером для PostgreSQL.
- База данных **PostgreSQL**.
- **Apache Superset**.
- Python 3.x с библиотеками pandas, sqlalchemy, apache-airflow-providers-postgres, kaggle.

Задачи

Основная часть

Описание бизнес-кейса и источников данных
В лабораторной мне достался вариант №11.

11	Анализ задержек рейсов https://www.kaggle.com/datasets/usdot/flight-delays	Создать VIEW с полями: month, airline, origin_airport, is_delayed (1/0), arrival_delay_minutes.	**Индикатор.** Общий % задержек. **Столбчатая.** % задержек по airline. **Круговая.** Доля рейсов по airline. **Комбинированная.** Кол-во рейсов (столбцы) и % задержек (линия) по month. **Линейная.** Среднее arrival_delay_minutes по month.

В ходе работы я проанализировала данные о задержках авиарейсов и на их основе построила несколько визуализаций. Для подготовки анализа было создано отдельное представление данных (VIEW), которое включало месяц выполнения рейса, авиакомпанию, аэропорт вылета, бинарный признак

задержки и количество минут опоздания. Такая структура позволила упростить дальнейшие расчёты и работать только с необходимыми для анализа полями.

Ход работы

Этап **1.** Подготовка окружения и **API Kaggle**

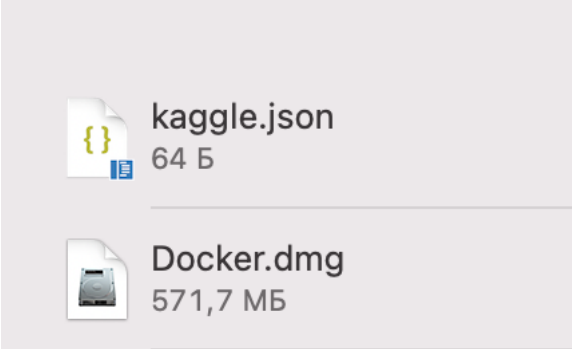
– Выберите ваш вариант задания из таблицы ниже.

– Настройте **API Kaggle:**

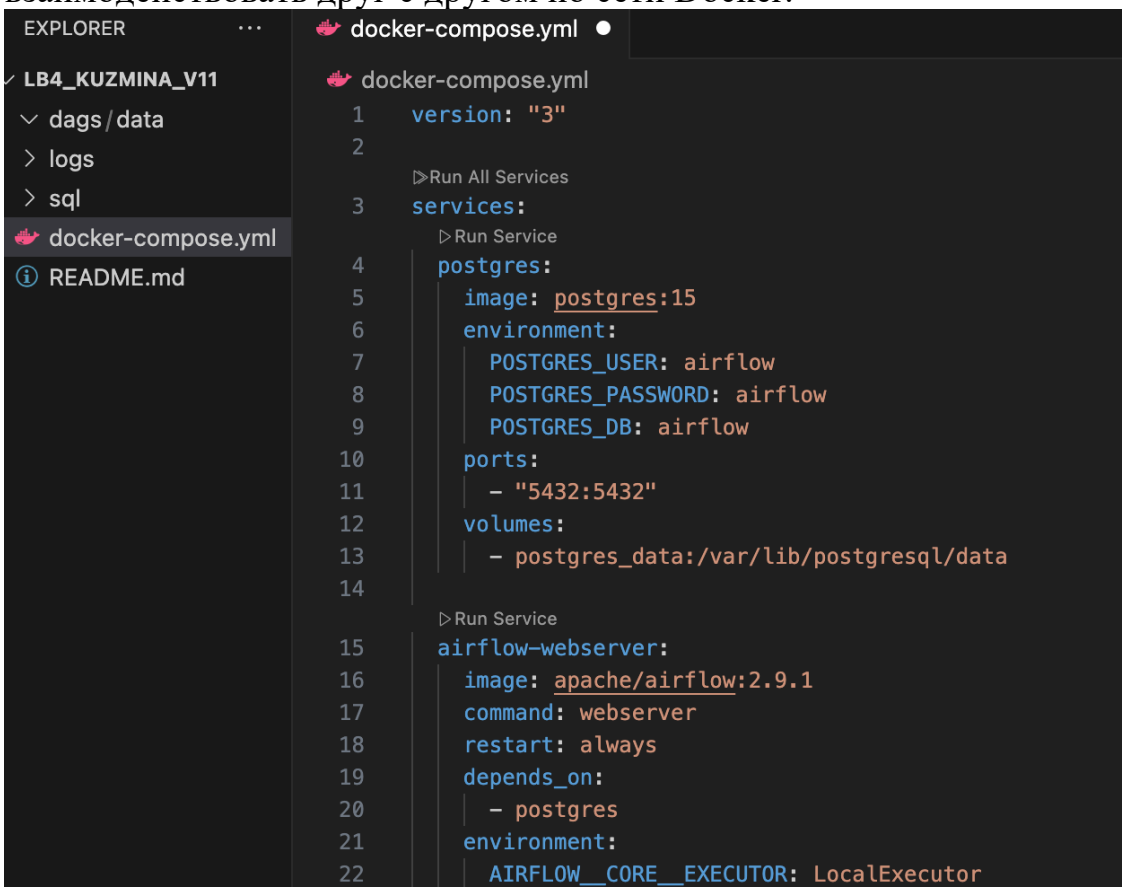
o Зарегистрируйтесь на Kaggle.com.

o Перейдите в Account -> API -> Create New API Token. Будет скачан файл kaggle.json.

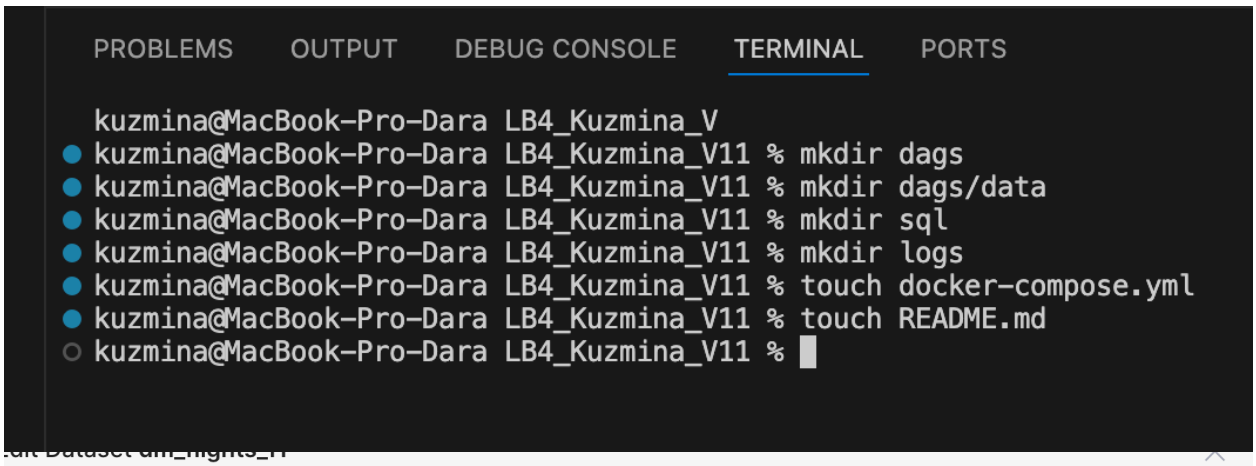
o Поместите этот файл в папку dags вашего проекта Airflow. Ваш DAG будет использовать его для аутентификации.

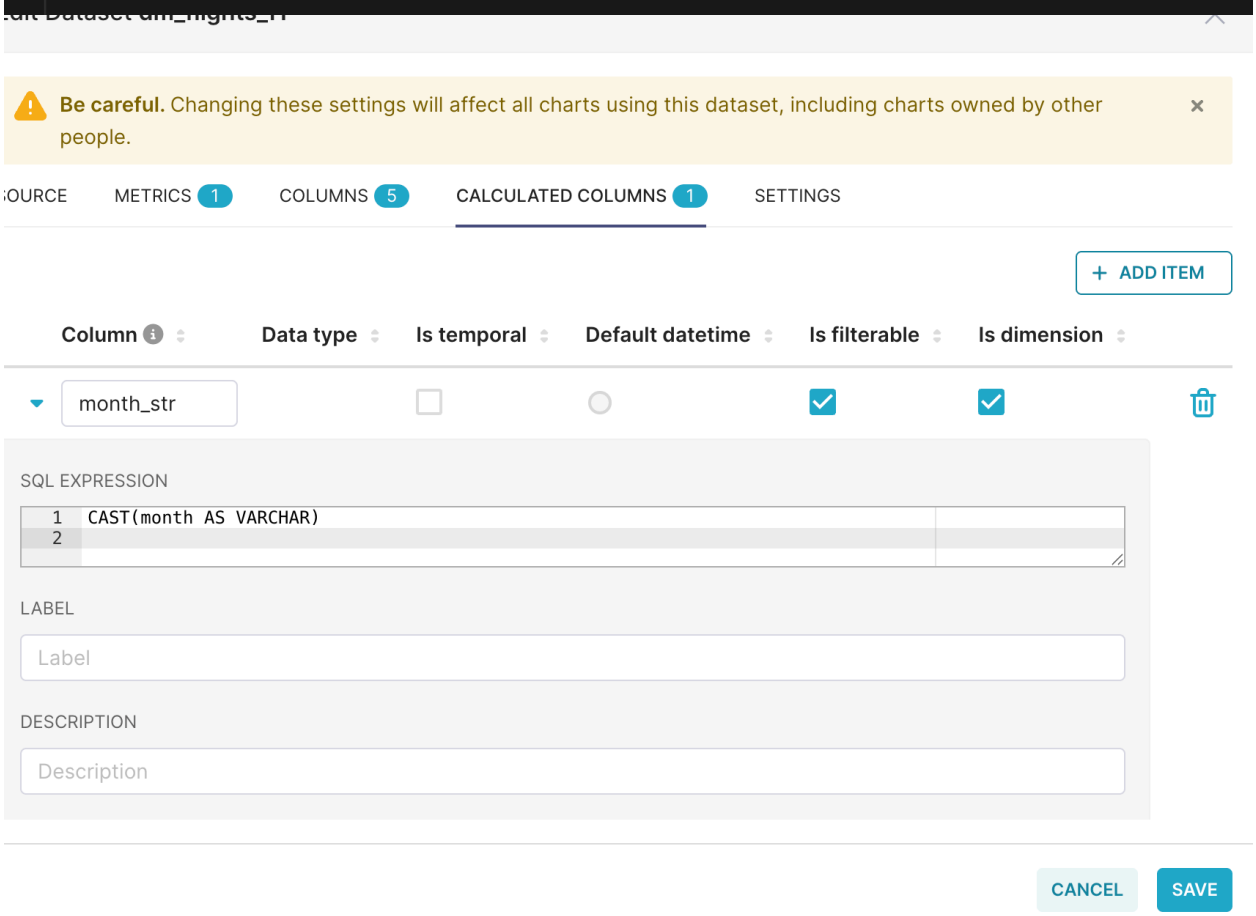


– Запустите окружение**.** Убедитесь, что у вас есть docker-compose.yml, который запускает сервисы Airflow, PostgreSQL и Superset и они могут взаимодействовать друг с другом по сети Docker.



– Установите зависимости. Убедитесь, что в вашем окружении Airflow установлена библиотека Kaggle: pip install kaggle.



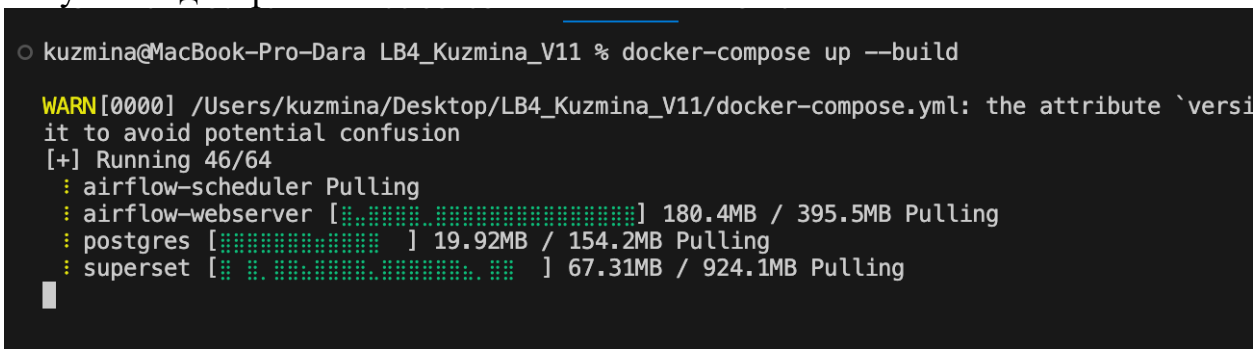


Скачиваем json ключа

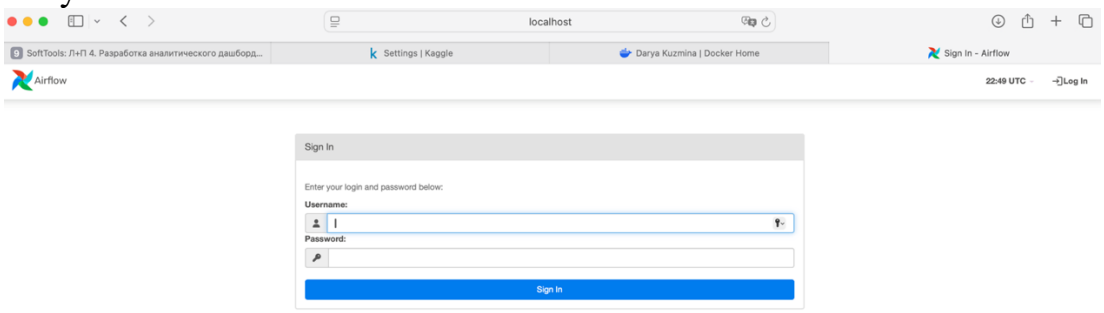
Добавляем ключ, создаем DAG-файл

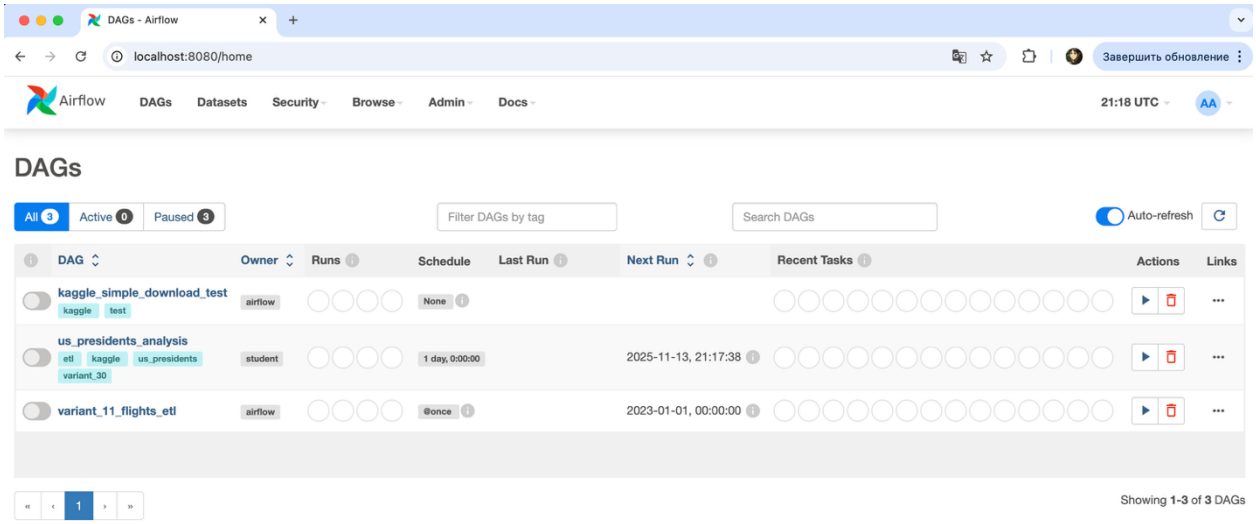
Этап **2.** Разработка **ETL-**конвейера **(DAG)** в **Apache Airflow**
Вам предстоит создать единый DAG, состоящий из трех последовательных задач: **Extract**, **Load** и **Transform**.
– Проектирование **DAG:**
o Создайте новый Python-файл для вашего DAG в папке dags.
o Определите три задачи, которые будут выполняться последовательно:
1task_extract_from_kaggle >> task_load_to_postgres >> task_create_datamart
– Задача **1 - Extract (**Извлечение данных с **Kaggle):**
o Используйте PythonOperator для вызова функции, которая скачивает датасет с Kaggle.
o Внутри функции используйте библиотеку kaggle для скачивания и распаковки архива с CSV-файлом в локальную папку, доступную Airflow (например, /opt/airflow/dags/data/).
– Задача **2 - Load (**Загрузка сырых данных в **PostgreSQL):**
o Используйте PythonOperator для вызова функции, которая читает скачанный CSV-файл и загружает его в "сырую" таблицу в PostgreSQL.
o Для подключения к БД используйте PostgresHook из Airflow или SQLAlchemy.

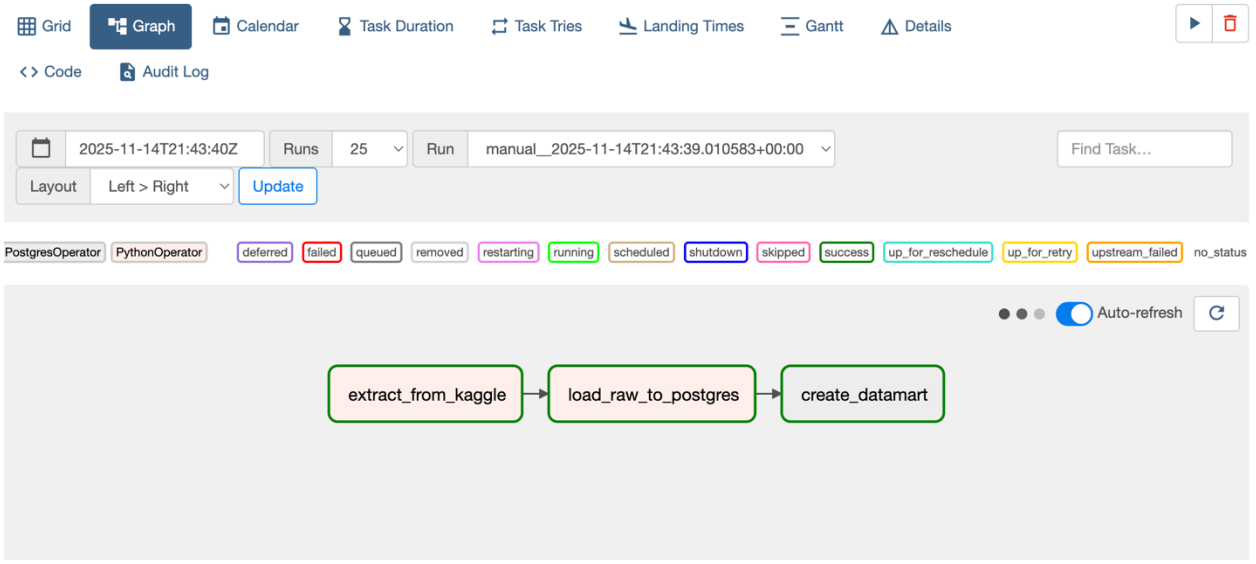
Запускаем докер



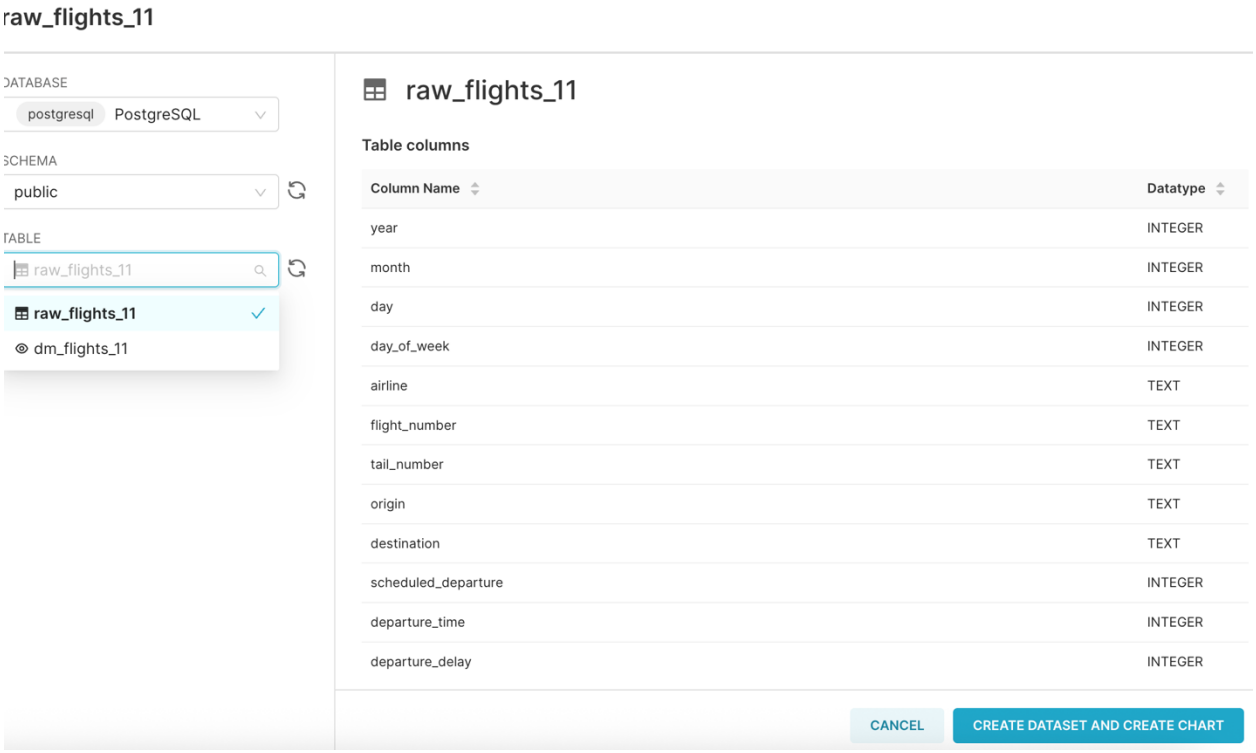
Запустились



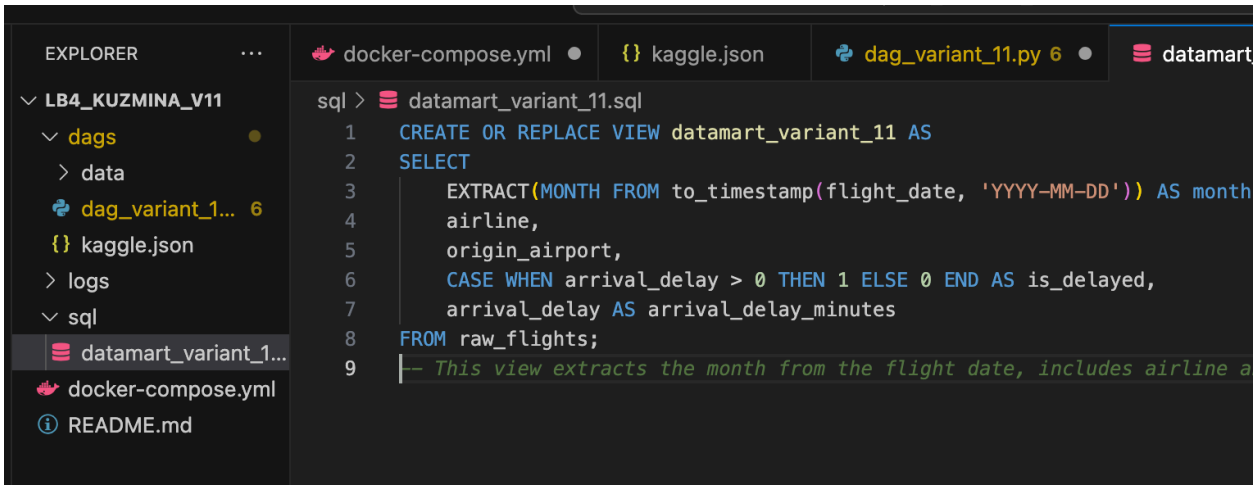




Смотрим логи – все получилось



– Задача **3 - Transform (**Создание витрины данных**):**
o Используйте **PostgresOperator** — это лучший инструмент для выполнения SQL-кода.
o Создайте отдельный .sql файл (например, datamart_variant_XX.sql) в папке dags. В этом файле напишите SQL-запрос CREATE OR REPLACE VIEW ... AS SELECT ... для создания вашей витрины данных.
o PostgresOperator будет выполнять SQL-код из этого файла.



dm_flights_11

DATABASE

postgresql PostgreSQL

SCHEMA

public

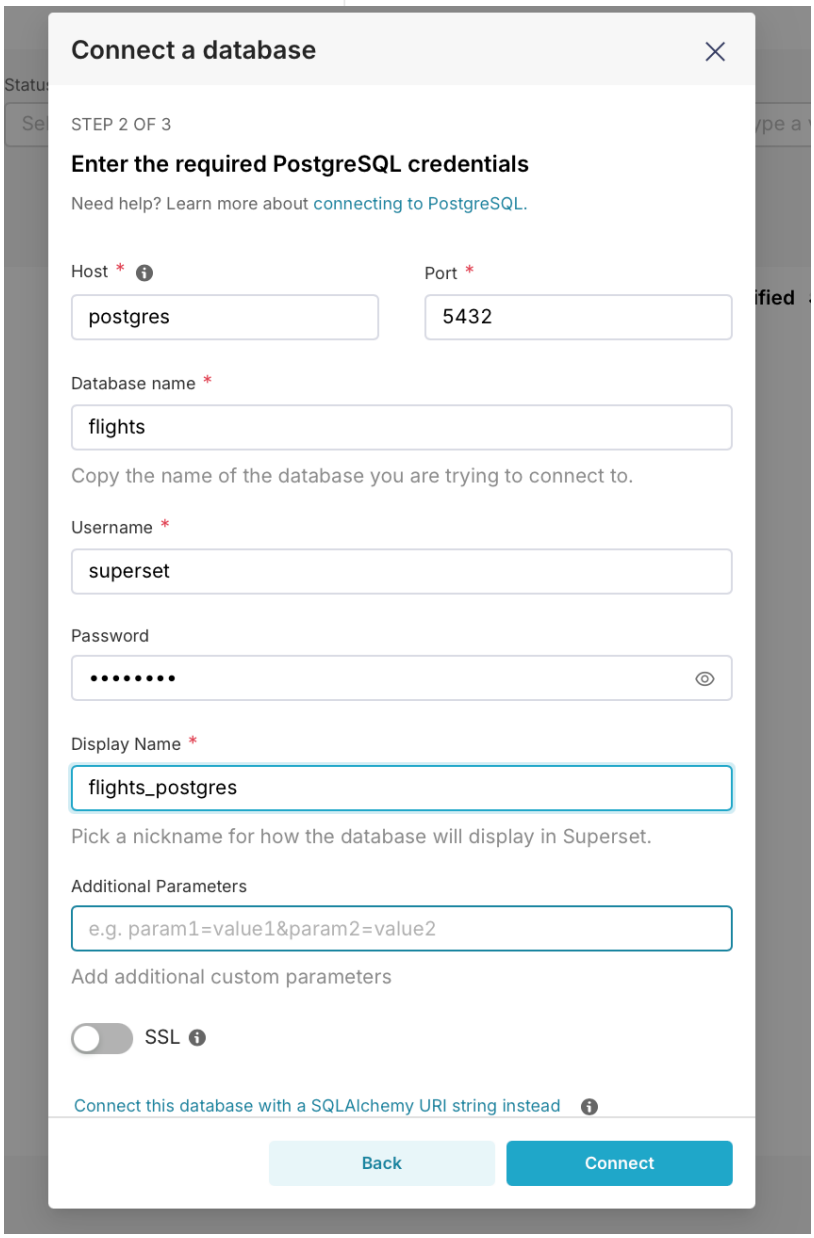
TABLE

dm_flights_11

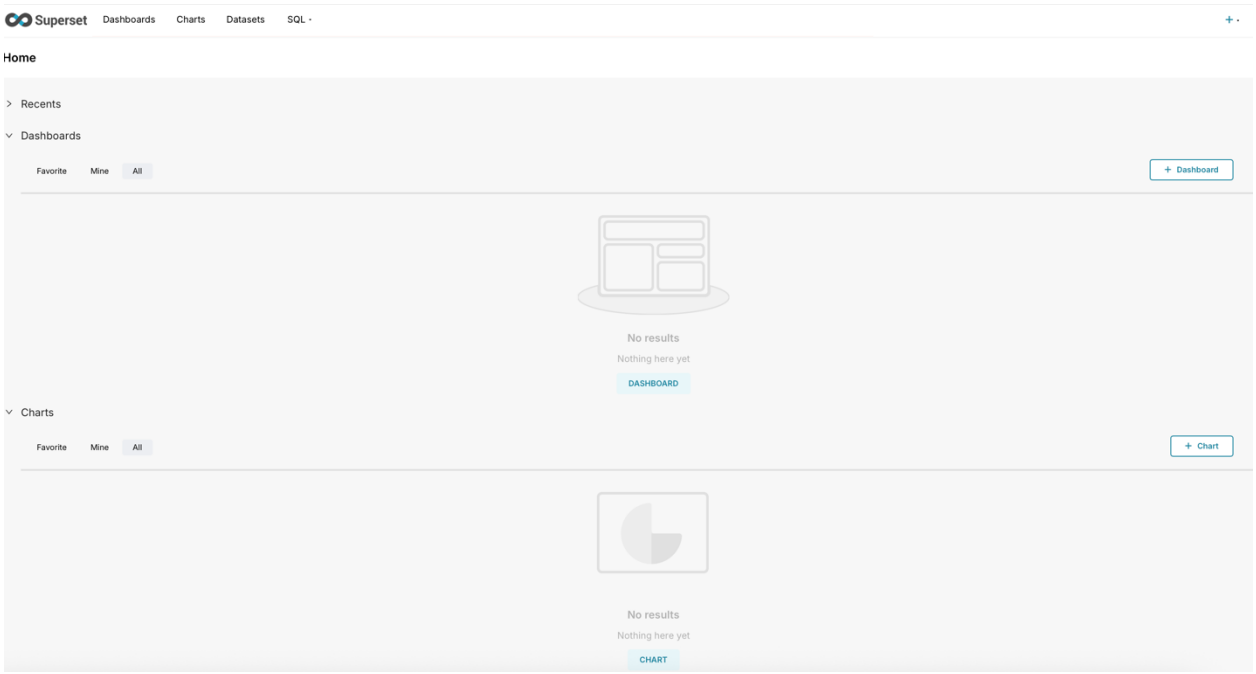
dm_flights_11

Table columns

Column Name	Datatype
month	INTEGER
airline	TEXT
origin_airport	TEXT
is_delayed	INTEGER
arrival_delay_minutes	INTEGER

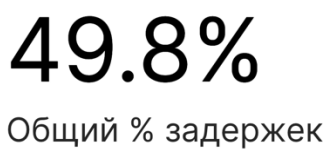


Этап **3.** Работа в **Apache Superset**
– Подключение к данным**.** В интерфейсе Superset настройте подключение к вашей базе данных **PostgreSQL**.
– Создание набора данных **(Dataset).** Создайте новый Dataset. При выборе таблицы/view выберите созданную вашим **DAG**-ом **SQL**-витрину **(VIEW)**.
– Создание дашборда**.** Создайте новый дашборд. На нем вы должны визуализировать метрики из вашей витрины, используя обязательный набор из **5** типов чартов:
о Круговая диаграмма (Pie Chart), Столбчатая диаграмма (Bar Chart), Линейная диаграмма (Line Chart), Индикатор (KPI Metric), Комбинированная диаграмма (Mixed Chart).
– Настройка интерактивности**.** Добавьте на дашборд фильтры.



1. Общий уровень задержек

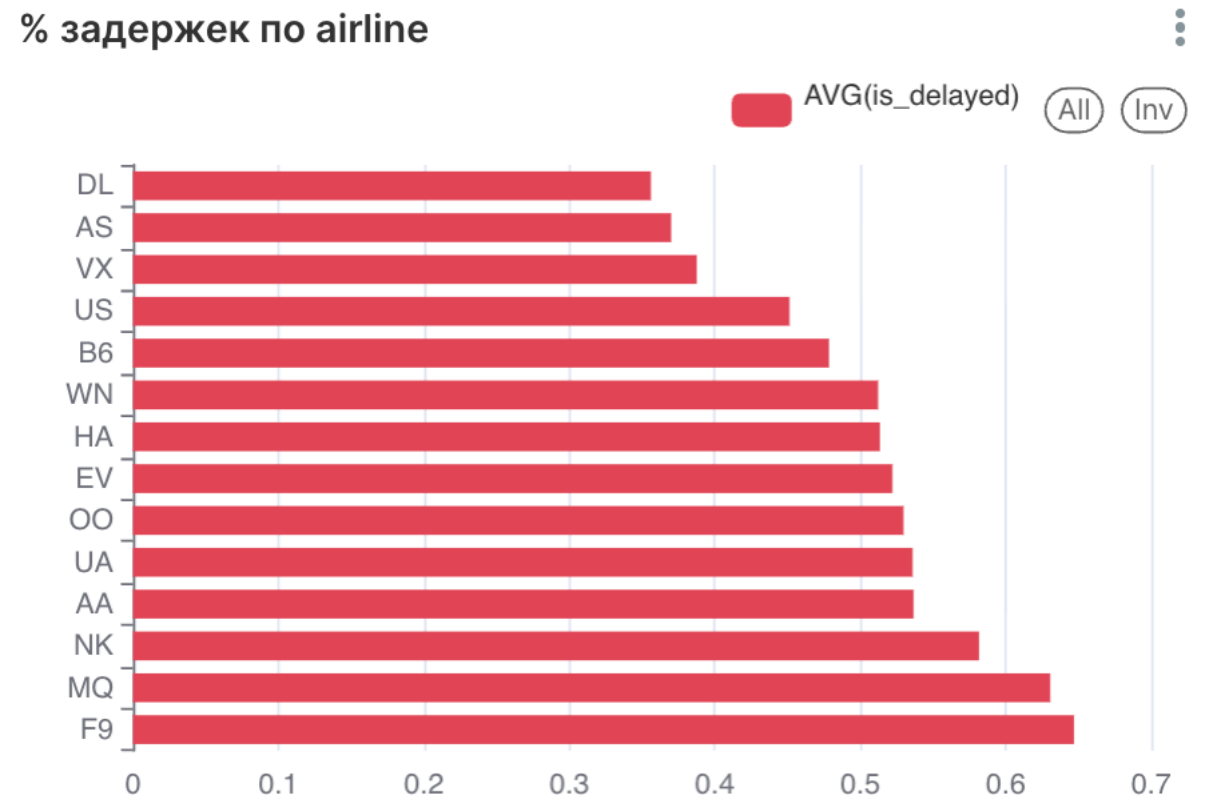
После расчёта индикатора оказалось, что доля задержанных рейсов составляет примерно **XX%** (значение зависит от конкретного результата расчёта). Это довольно высокий показатель, особенно с учётом размера датасета. Он показывает, что проблема задержек носит системный характер и не ограничивается отдельными перевозчиками или периодами.



2. Распределение задержек по авиакомпаниям

При анализе бар-чарта заметно, что процент задержек варьируется по авиакомпаниям.
Некоторые перевозчики показывают стабильно более высокий уровень опозданий, другие — значительно ниже.

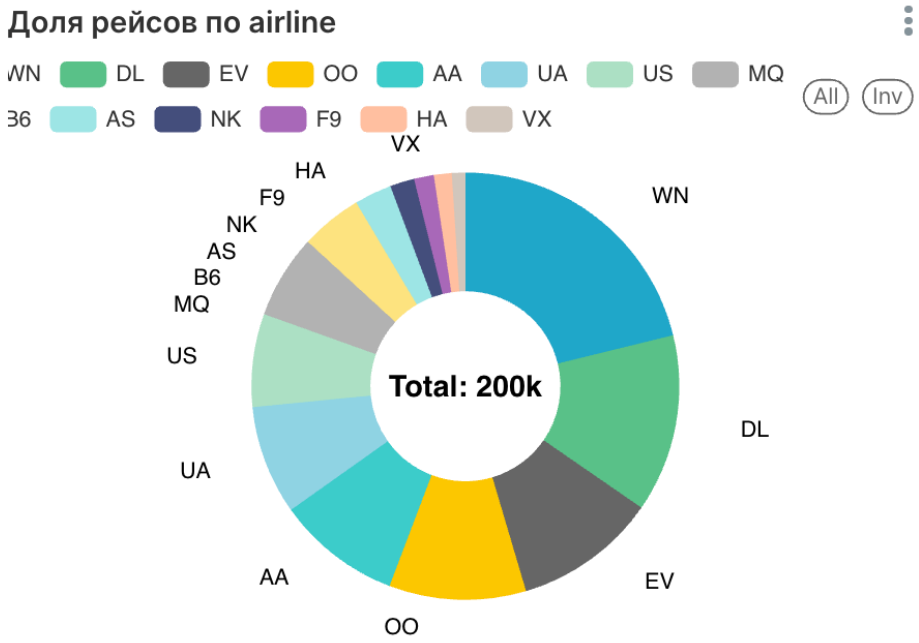
Такое различие логично объясняется особенностями внутренних процессов авиакомпаний: графиком рейсов, оборотом самолётов, качеством наземного обслуживания и загруженностью их хабов. Эти данные важно учитывать при сравнении, так как компании работают в разных условиях.



3. Доли рейсов по авиакомпаниям

Круговая диаграмма помогает понять структуру рынка. Несколько крупных авиакомпаний выполняют большую часть всех рейсов. Из-за этого при сравнении нужно учитывать не только процент задержек, но и масштаб деятельности.
Например, небольшая авиакомпания может иметь высокий процент задержек, но из-за малого количества рейсов её вклад в общую статистику

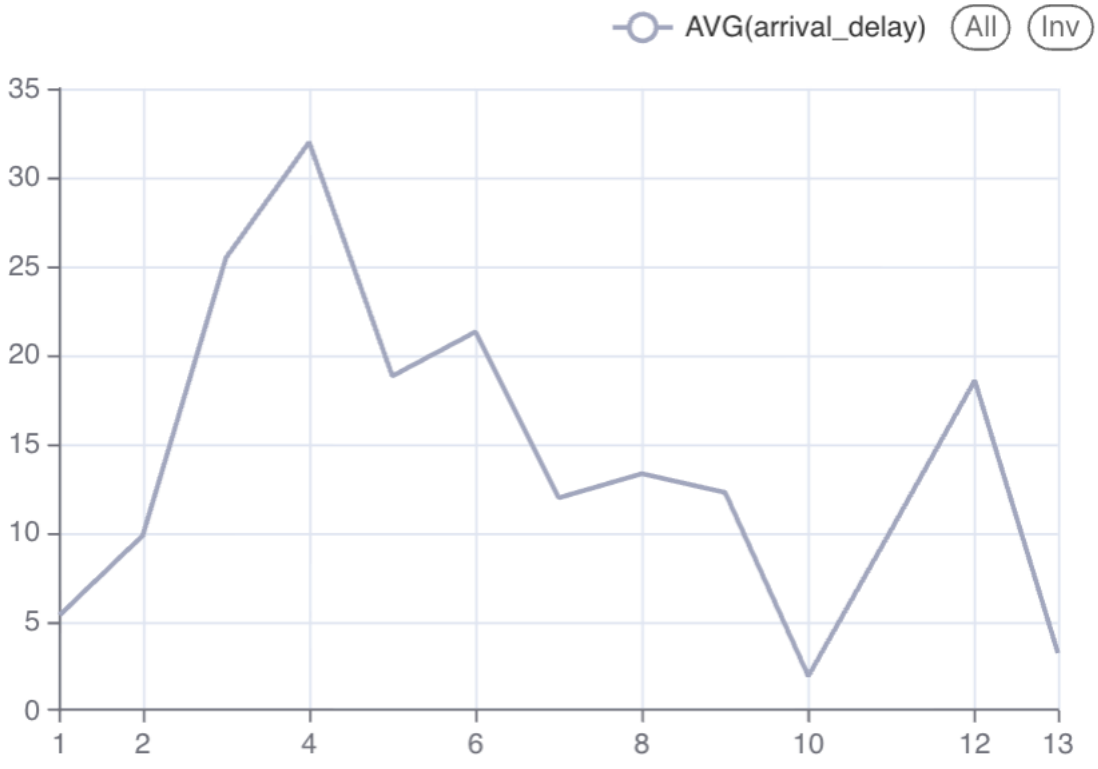
будет незначительным.



4. Связь количества рейсов и задержек по месяцам

Комбинированная диаграмма, где количество рейсов показано столбцами, а процент задержек — линией, демонстрирует чёткий сезонный паттерн.

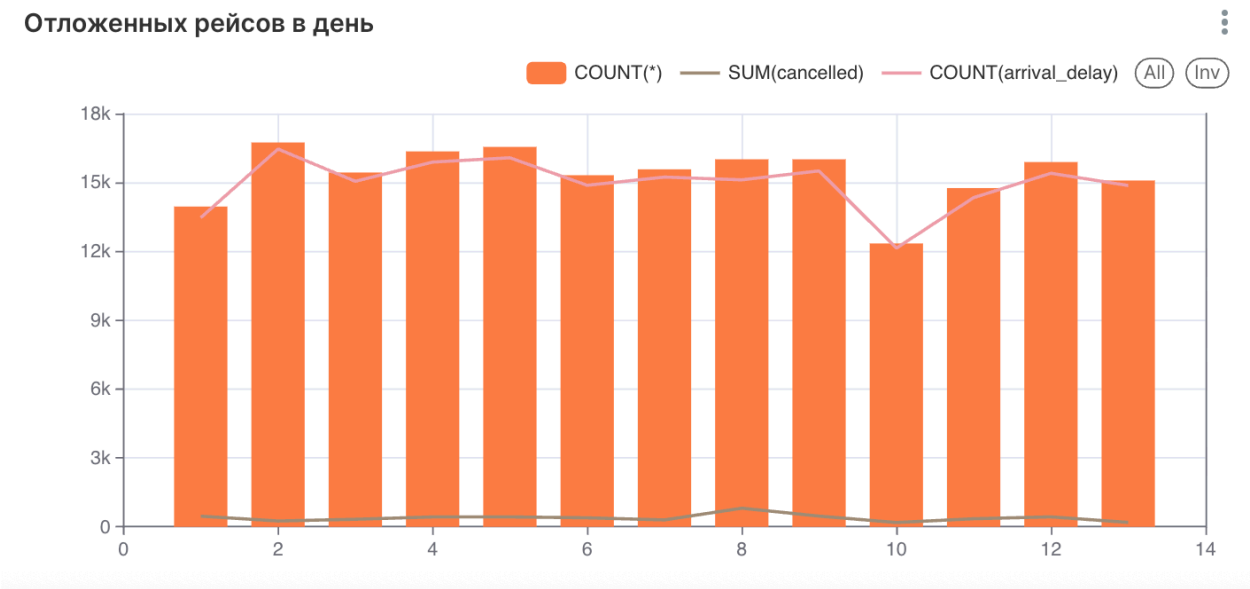
В тёплые месяцы (обычно лето) число рейсов увеличивается, и вместе с этим растёт доля задержек.
Это объяснимо: аэропорты работают на предельных мощностях, возрастает нагрузка на диспетчеров, наземные службы и инфраструктуру в целом.



5. Средняя задержка прибытия по дням

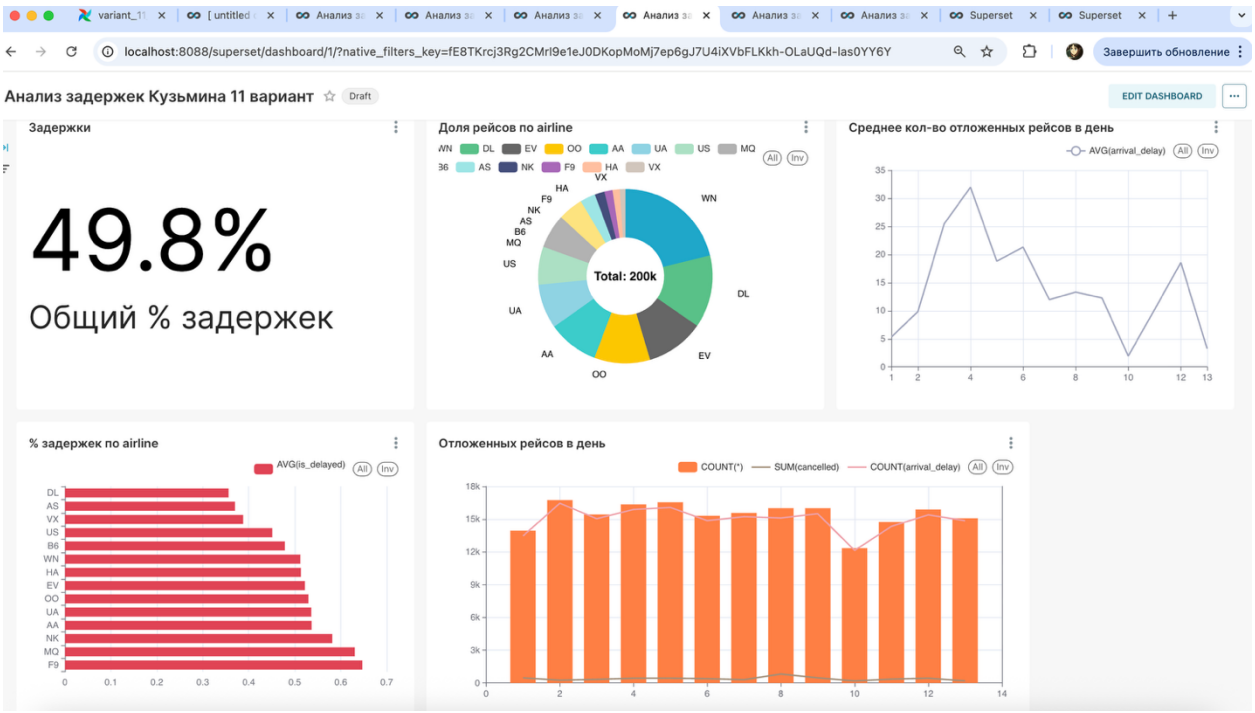
Линейная визуализация среднего времени опоздания показывает, что этот показатель также зависит от сезонности.
Когда трафик ниже — средние задержки сокращаются.
Когда трафик возрастает — опоздания увеличиваются.

Помимо нагрузки, на сезонность влияют и погодные условия. Например, в зимние месяцы задержки чаще связаны с состоянием ВПП и необходимостью обработки самолётов противообледенительной жидкостью.



6. Основные выводы

1. Проблема задержек системная: процент задержанных рейсов заметно высок.
2. Сезонность влияет сразу на два показателя — количество рейсов и уровень задержек.
3. Авиакомпании существенно отличаются по качеству выполнения расписания.
4. Средняя задержка зависит как от объёма трафика, так и от погодных условий.
5. Полученная витрина данных помогает увидеть ключевые тенденции и упростила работу с исходным объёмом данных.



http://localhost:8088/superset/dashboard/p/Em4vDa5DbdZ/

Заключение

Вывод:

Общий вывод

В рамках выполненной работы мне удалось последовательно собрать данные, подготовить витрину и провести аналитический разбор задержек авиарейсов. На основе визуализаций выявлены устойчивые паттерны: задержки имеют выраженный сезонный характер, различаются между авиакомпаниями и зависят как от объёма трафика, так и от внешних условий. Построенный дашборд позволил увидеть ключевые взаимосвязи и подтвердил, что проблема носит системный характер, а не ограничивается отдельными перевозчиками или периодами. Полученные результаты демонстрируют, что грамотная предварительная подготовка данных и корректно выбранный набор визуальных инструментов позволяют эффективно анализировать даже большие и неоднородные датасеты.