# Отчёт по тестовому заданию для практикума «От SQL до DWH»

В качестве источника данных был выбран csv файл, доступный по http. Выбран файл <https://cloud.mail.ru/public/L1xB/nvgHGYJz5> и загружен на гугл диск. Приложение читает копию файла с гугл диска.

## Схема объектов БД

Поля исходного csv файла TICKER, PER, TIME не будут загружаться в postgres, так как все значения одинаковы, следовательно, информации не несут.

Поле DATE – содержит уникальные отсортированные значения дат. Поэтому логично назначить его первичным ключом для таблицы в postgres.

Поля OPEN, HIGH, LOW, CLOSE, VOL – содержат числовые показатели для каждой даты.

В postgres создадим одну таблицу

|  |
| --- |
| CREATE TABLE IF NOT EXISTS dataset (  dt date PRIMARY KEY,  open NUMERIC(10, 2) NOT NULL,  high NUMERIC(10, 2) NOT NULL,  low NUMERIC(10, 2) NOT NULL,  close NUMERIC(10, 2) NOT NULL,  volume bigint not null  ); |

## Программный код Python

Состоит из модулей:

- datasouce – содержит реализацию извлечения данных из удалённого файла по http и конвертацию их в нужный вид.

- uploader – содержит реализацию загрузки поступающих данных в postgres. Загружает строки небольшими пакетами (количество строк в пакете устанавливается переменной окружения BATCH\_SIZE). Каждый пакет загружается в отдельной транзакции.

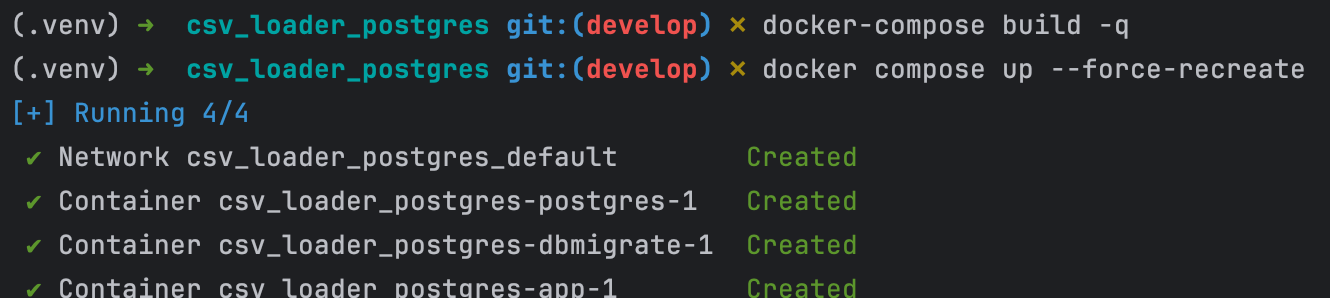
- main.py – запускает и инициализирует приложение

## Конфигурационные файлы для развёртывания докеров

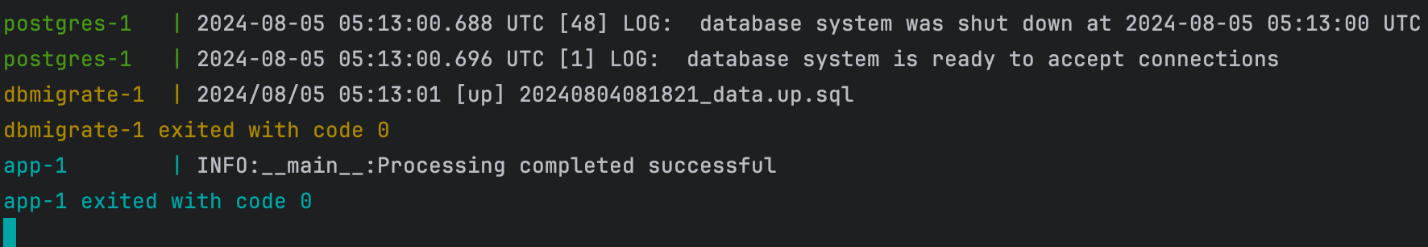
Для демонстрации работы подготовлен docker-compose файл с необходимым окружением. Он использует значения переменных, описанных в файле .env.

## Скриншоты, демонстрирующие работу приложения

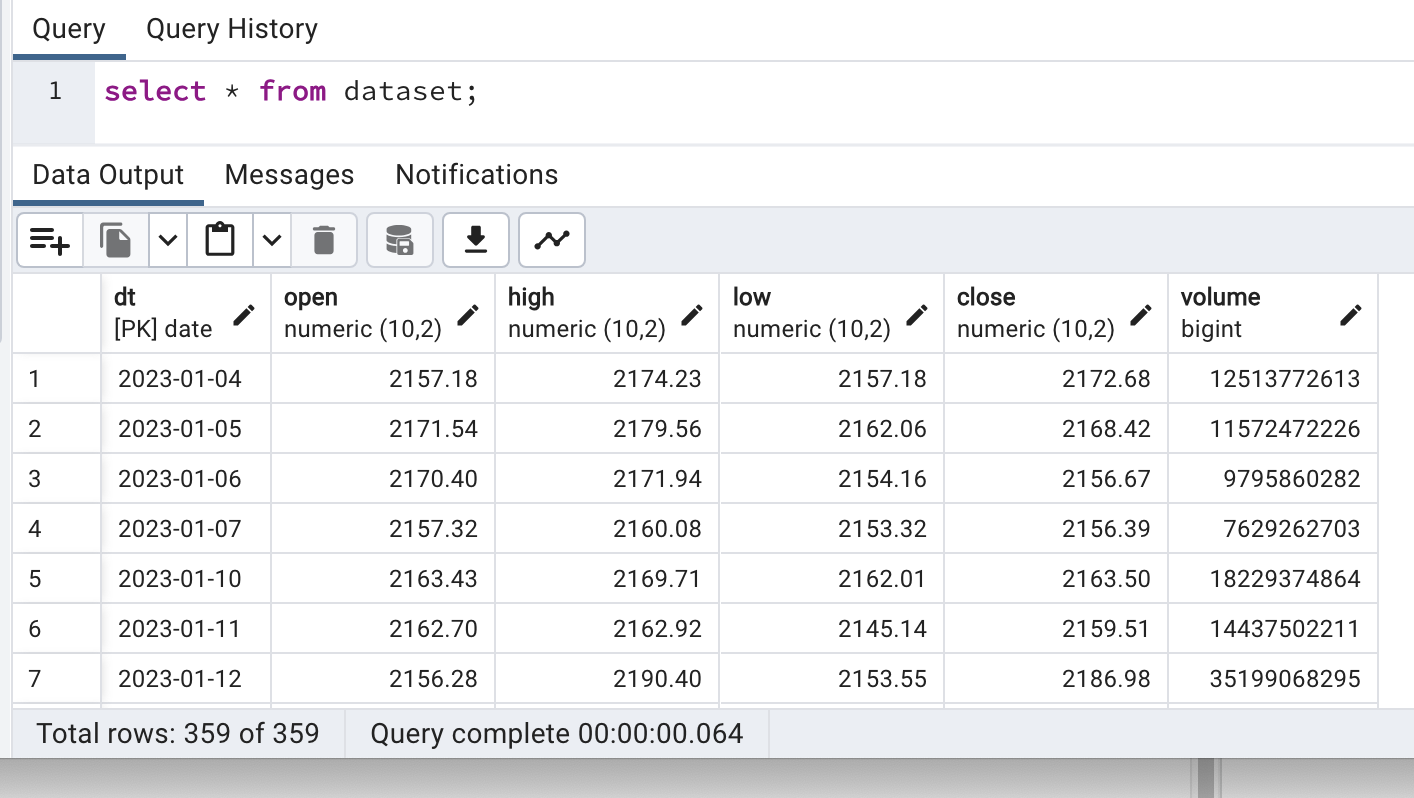
1. Запуск



1. Приложение успешно загрузило данные



1. Проверим их в postgres



В результате видим, что все данные загружены.