Архитектура решения.

1. Контекст.
2. Требования заказчика
3. Пример желаемого результата
4. Схема с технической архитектурой (MVP)
5. Технические компоненты
6. Допущения и ограничения
7. Список планируемых улучшений для дальнейшей масштабируемости
8. Список тестов для валидации

1. Контекст.

Заказчик - онлайн-магазин.
Сущности - данные о клиентах, товарах, заказах и составах заказов
Данные поступают батчами в csv файлах для разных сущностей (1 день = 1 батч = 1 итерация пайплайна).
данные могут новые записи, так и обновления существующих. Удалений нет.

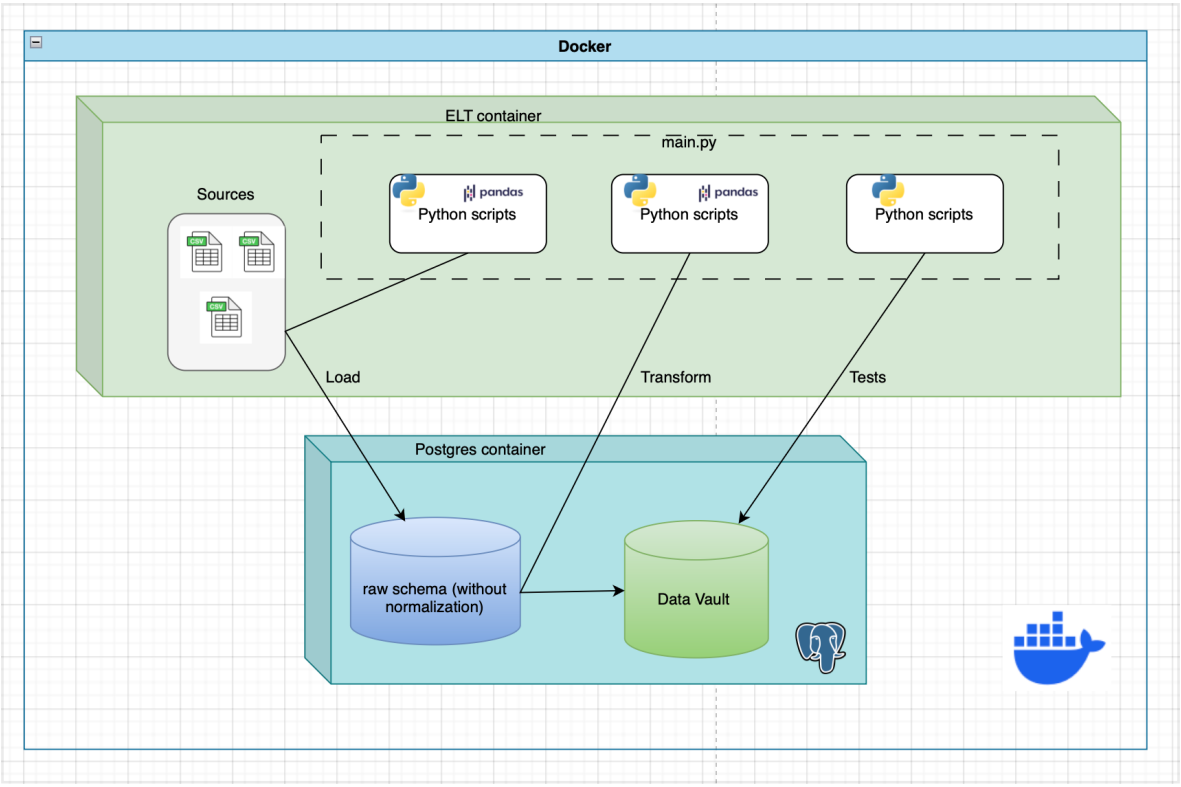
2. Требования заказчика

Система собирает данные из csv файлов в БД с сохранением истории изменений (схема Raw Data Vault с SCD Type 2).
Данные загружаются AS IS (без изменений, очисток) - только необходимая техническая нормализация.

3. Результат

Заполненная БД с данными.

4. Схема с технической архитектурой (MVP)



5. Технические компоненты:
    - Python
    - Docker
    - PostgreSQL
    - Pandas

6. Допущения и ограничения
    - Удаление записей не поддерживается, т.е. Только создание новых и обновление существующих
    - За одну итерацию пайплайн обрабатывает 1 день (не все csv сразу).
    - Бизнес-ключи в источнике уникальны (например, не будет двух разных клиентов с одним customer_id)
    - Групповые заказы не предусмотрены (1 заказ совершается максимум 1 покупателем)
    - Источник данных только csv
    - В исходных таблицах отсутствуют составные PK
    - Товары из заказа удаляться не могут или удаляются через обновление quantity = 0.

7. Список планируемых улучшений для дальнейшей масштабируемости

    - Замена pandas на PySpark
    - Замена Postgres на GreenPlum
    - Добавление визуализации (PowerBI)
    - Оркестрания

- Введение Каталога Данных.
- Расширенный набор тестов
- Добавление шины данных Kafka.
- Хранение csv в томах
- Добавить логирование

8. Список тестов для валидации
    - Сравнение count на источнике и в хранилище