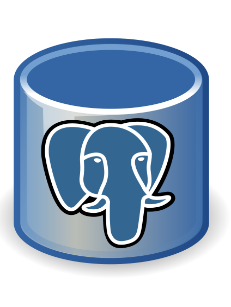
Архитектура решения

Схема архитектуры

Etl\_db\_4

ODS-слой

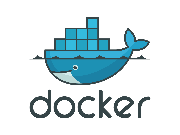
DM-слой

DDS-слой

intermediate-слой

Источник данных

Source



dashboards

Broken\_data



Слои обработки и управления

Описание компонентов

В приложении Docker поднят контейнер с Apache Airflow. Airflow обеспечивает оркестрацию ETL процесса. **Обработка данных происходит с помощью Python скриптов с использованием sql запросов и библиотеки pandas. Для визуализации используется** Luxms BI.

Источник данных:

**Исходные Данные хранятся в PostgreSQL в схеме source\_data.**

Слои обработки и управления:

1. ODS-слой: слой хранения данных из источника в «сыром», необработанном виде

Описание таблиц и атрибутов:

1. Intermediate-слой: слой хранения обработанных и приведённых к нужным форматам данных;
2. DDS-слой: слой хранения данных с созданными связями между таблицами и определенными ключевыми атрибутами;
3. DM-слой: слой специализированных витрин данных, сгруппированных в форме схемы «звезда».

Дополнительно: схема broken\_data. Создана для хранения «битых» данных из источника. Заполняется в два этапа:

* во время формирования intermediate-слоя (если в записи есть пропуск по определенному атрибуту);
* во время формирования DDS-слоя (при отсутствии ключа в таблице «сотрудники\_дар» на который ссылается запись в промежуточной таблице)

Описание ETL потока

1. Загрузка данных из источника данных в схему ODS происходит с помощью скрипта init\_ods.py. Данный скрип состоит из трех тасков:

create\_schema >> create\_tables\_in\_schema >> transfer\_data()

* create\_schema – создание схемы;
* create\_tables\_in\_schema – создание таблиц в схеме с необходимыми атрибутами;
* transfer\_data() – перенос данных (строк) из источника данных в созданную схему;

1. Скрипт init\_intermediate\_schema.py производит обработку данных и состоит из следующих тасков:

create\_intermediate\_schema >> create\_broken\_schema >> migrate\_sotrudniki >> migrate\_end >> migrate\_intermediate

* create\_intermediate\_schema, create\_broken\_schema – создание intermediate схемы и схемы с «поломанными» данными;
* migrate\_sotrudniki – отдельная обработка таблицы «сотрудники\_дар», как основной таблицы;
* migrate\_end – обработка справочных таблиц;
* migrate\_intermediate – обработка таблиц, которые связывают сотрудников с навыками и уровнями владения навыков;

1. Скрипт init\_dds\_schema.py отвечает за создание схемы с определенными связями:

create\_schema >> migrate\_sotrudniki >> create\_end\_tables >> transfer\_data\_end\_tables() >> create\_intermediate\_tables >> transfer\_data\_intermediate\_tables()

* create\_schema – создание схемы;
* migrate\_sotrudniki – создание таблицы «сотрудники\_дар» с ключевыми атрибутами и перенос данных для этой таблицы;
* create\_end\_tables – создание справочных таблиц с ключевыми атрибутами;
* transfer\_data\_end\_tables() – перенос данных для справочных таблиц;
* create\_intermediate\_tables – создание промежуточных таблиц с необходимыми внешними ключами;
* transfer\_data\_intermediate\_tables() - перенос данных для промежуточных таблиц;

1. Скрипт init\_dm\_schema.py отвечает за формирование витрины данных в схеме DM:

create\_schema\_and\_tables >> migrate\_knowledge\_levels >> migrate\_users >> migrate\_fact\_table

* create\_schema\_and\_tables – создание схемы и таблиц с необходимыми связями;
* migrate\_knowledge\_levels, migrate\_users – перенос данных в таблицу справочник;
* migrate\_fact\_table – перенос данных в таблицу фактов

Схема витрины данных

