Система сбора и хранения данных о просмотрах каналов (роликов) Youtube

Цели и задачи

Цель системы состоит в регулярном сборе и хранение значимых метрик роликов и каналов Youtube с обеспечением доступа к этой информации с помощью систем визуализации (powerBI) или других систем.
Основной потребитель данных – аналитики, маркетинг.

Технологический стек системы

Python 3.10, PostgreSQL, Selenium
(отладка и разработка jupyter notebook)

Структура и краткое описание системы

Для хранения данных используется БД PostgreSQL. Она состоит из 4х таблиц. Текущее состояние «частично» нормализованная по 2ой форме. В таблицах хранится:

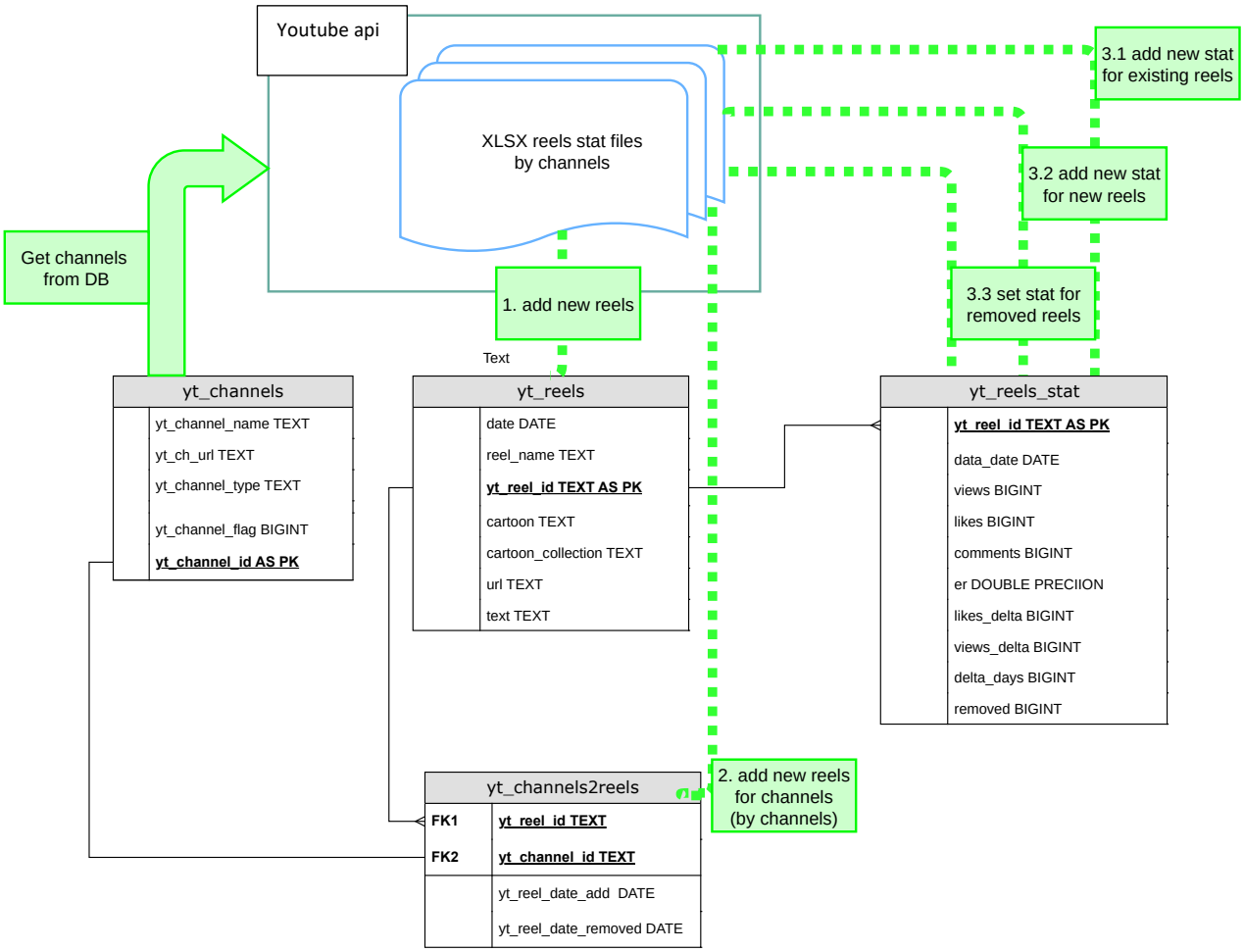
1. список youtube каналов (yt_channels)
2. список и описание youtube роликов (yt_reels)
3. принадлежность роликов к каналам (yt_channels2reels)
4. статистика по роликам (yt_reels_stat).

Выгрузка данных о просмотрах осуществляется с помощью youtube API в файлы xlsx с данными по роликам на канале (имя xlsx файла – ID канала). Список каналов скрипт получает из таблицы БД. Также скрипт генерит два файла - report_urls.csv – файл, в котором описывается привязка xlsx файлов и каналов, по которым они содержат информаицю и файл failed_yt.csv – список каналов, по которым статистика (по каким-либо причинам) не была получена (выгрузка данных не произошла).
Вторым скриптом – yt_parser_loader.py – производится парсинг xlsx файлов r и их последовательная обработка. В ходе которой производятся следующие операции:

1. получение списка роликов и обновление таблицы роликов
2. получение списка роликов на данном канале и обновление таблицы yt_channels2reels.
3. Обновление таблицы со статистикой по роликам yt_reels_stat

После обработки всех xlsx файлов (всех данных собранных по каналам) производится обновление таблицы yt_reels_stat – дублирование последних данных статистики на данную дату сбора информации и разметка удаленных роликов.

Структура системы и БД (ERD)



Работа скриптов

Скрипт yt_get_stat.py

Требования: python 3.x, настройки для подключения к БД системы, youtube API v3 token.
Функции в скрипте:
def get_channel_videos(channel_id) – возвращает ID видеороликов на канале channel_id
def get_video_info(v_id) – возвращает статистику по роликам (v_id). На входе список, сбор идет по 50 роликов за запрос. На выходе – список.
def make_table(final,ch_views, ch_subscribers, ch_videos) – формирование финального dataframe, исходя из данных по роликам и по каналу, с которого они взяты.
Работа скрипта – из БД выгружаются ID каналов (которые не закрыты – поле flag_closed) для сбора статистики. По ним формируется список роликов на канале и собирается статистика по каналу и по роликам. Формируется dataframe, с данными по роликам и выгружается в файл с именем ID канала, статистика по всем каналам выгружается в CSV файл.

Скрипт yt_load_ch_stat.py

Скрипт загрузки статистики подписчиков и просмотров по каналам в БД.

Скрипт yt_parse_and_load4API.py

Требования: python 3.x, настройки для подключения к БД системы, файл отчета report_urls.csv, xlsx файлы с выгруженной статистикой, файл ./ref/vocabulary.csv словаря категоризатора роликов
Функции в скрипте:
def iso_to_sec – преобразует формат длительности роликов к привычному.
Функции парсинга названий
def check_items_cartoon(reel_name): функция по названию ролика формирует список обнаруженных вхождений в словарь и возвращает список
def check_len_cartoon(collection_list): просчет числа записей в списке обнаруженных вхождений и возвращает ‘none’ если запись одна или ‘сборник’, если записей несколько
def convert_collect(collection_list): преобразует список записей в одну строку, если записей нет (none) – то возвращает none
def alphanum(element): «очистка» строки от спецсимволов и тп. Остаются буквы и символы []":.,//"
def is_shorts(sec, r_name, ch_name, live) – попытка определись является ли ролик “shorts” (по наличию тега в имени ролика или канала и длительности ролика меньше 61 секунды).

Функции обновления таблиц БД

def proc_reels_to_sql(reels_to_sql): Функция добавления роликов в таблицу yt_reels, которых там нет. На входе dataframe с новыми роликами, которые функция добавляет в таблице yt_reels. В функции происходит парсинг названия и соответствующая категоризация по результатам парсинга.

proc_ch2reels_to_sql(ch2reels_to_sql, ch_id, fdate): Функция добавления роликов в таблицу yt_channels2reels. На входе dataframe с новыми роликами, ch_id - идентификатор канала и дата, которая будет проставлено в поле 'yt_reel_date_add – дата добавления ролика на канал.
proc_ch2reels_update(removed_reels, ch_id, fdate): Функция обновления таблицы yt_channels2reels. Для роликов, которые перестали присутствовать на канале ставится дата «удаления» с канала в поле yt_reel_date_removed. На входе dataframe с удаленными роликами, ch_id – идентификатор канала, fdate – дата.

proc_new_reels_stat(reels, ch_id, fdate): Функция добавления статистики новых роликов в таблицу yt_reels_stat. Reels – dataframe с информацией о роликах, fdate – дата информации.
proc_exist_reels_stat(reels, ch_id, fdate): Функция добавления статистики роликов, которые уже есть в таблице yt_reels_stat. Reels – dataframe с информацией о роликах, fdate – дата информации.

process_file(fpath, ch_id): Функция парсинга XLSX файлов со статистикой и формирования данных для вызова функций обновления таблиц. Считываются данные из файла. После чего происходит последовательное сравнение с данными из таблиц yt_reels (загрузка данных при обработке каждого файла XLSX), yt_channels2reels (загрузка данных только по ch_id при обработке каждого файла XLSX), yt_reels_stat (данные из таблицы загружаются один раз вначале скрипта).
def process_removed_reels(fname) обновление таблицы yt_reel_stat – для роликов, по которым больше нет статистики (считаем что они удалены с youtube каналов) – статическая информация копируется из последних имеющихся данных, дифференциальная устанавливается равной нулю, устанавливается поле removed в значение 1.

Остальной код скрипта строится на, последовательном вызове обработчика файлов XLSX.

План развития системы

Июль 2022 года – запуск пробной версии системы

Декабрь 2022 года – Приведение системы к стандартам: нормализация БД, реализация новой информации по структуре youtube (например, что ролик может быть на нескольких каналах), сохранение статистики по удаленным роликам.

Март 2023 года – отказ от popsters.ru. Использование API youtube для получения данных, в том числе тех, что не выгружаются popsters.ru

Апрель 2023 года – внедрение NLP системы для категоризации роликов (парсинга текстовых описаний)