

# **Анализ публикуемых новостей**

создание ETL-процесса формирования витрин данных  
для анализа публикаций новостей

# Общее описание проекта

Общая задача: создать ETL-процесс формирования витрин данных для анализа публикаций новостей с трех информационных источников:

- <https://lenta.ru/rss/>
- <https://www.vedomosti.ru/rss/news>
- <https://tass.ru/rss/v2.xml>

Подробное описание задачи:

1. Разработать скрипты загрузки данных в 2-х режимах:
  - о Инициализирующий – загрузка полного слепка данных источника
  - о Инкрементальный – загрузка дельты данных за прошедшие сутки
2. Организовать правильную структуру хранения данных
  - о Сырой слой данных
  - о Промежуточный слой
  - о Слой витрин

# Цели проекта и требуемый результат

В качестве результата работы программного продукта необходимо написать скрипт, который формирует витрину данных следующего содержания:

- Суррогатный ключ категории
- Название категории
- Общее количество новостей из всех источников по данной категории за все время
- Количество новостей данной категории для каждого из источников за все время
- Общее количество новостей из всех источников по данной категории за последние сутки
- Количество новостей данной категории для каждого из источников за последние сутки
- Среднее количество публикаций по данной категории в сутки
- День, в который было сделано максимальное количество публикаций по данной категории
- Количество публикаций новостей данной категории по дням недели

Дополнение: необходимо привести названия категорий в разных источниках к единому виду.

# План реализации

Настройка  
окружения



Создание модуля  
сбора сырых  
данных



Создание  
хранилища  
данных



Отладка  
процедур работы  
с хранилищем



Настройка  
оркестратора

# Используемые технологии

- Docker
- Python
  - datetime, pytz – для работы с временными данными
  - pandas – обработка данных
  - sqlalchemy – работа с SQL
  - requests, bs4 – загрузка web страниц
  - os – получение списка загруженных файлов и времени последней новости из их названий
  - re – выражения для дополнительного разбора страниц и анализа списка файлов.
- PostgreSQL
  - Materialized view – для витрины данных
- Airflow
  - DAG – для их создания
  - Variable – для хранения названия подключения
  - PythonOperator – запуск сбора и обработки данных
  - PostgresHook – подключение к PostgreSQL внутри кода Python
  - PostgresOperator – выполнение обособленных действий с PostgreSQL

# Схемы/архитектуры

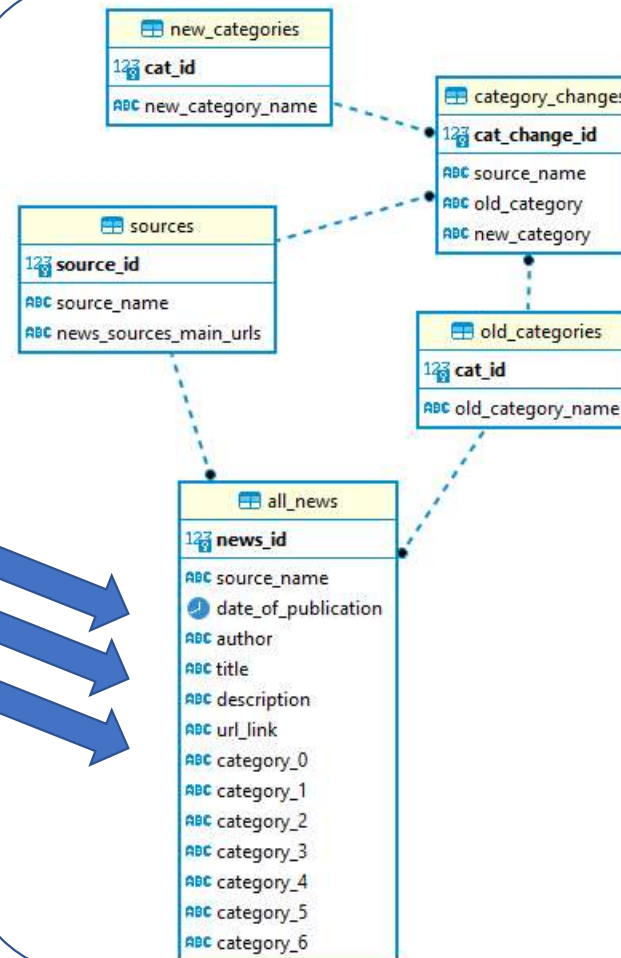
Python  
(web -> csv)

Временные таблицы  
(для избежания конфликтов –  
отдельная для каждого источника)

tmp_news0
ABC source_name
ABC date_of_publication
ABC author
ABC title
ABC description
ABC url_link
ABC category_0
ABC category_1
ABC category_2
ABC category_3
ABC category_4
ABC category_5
ABC category_6

tmp_news1
ABC source_name
ABC date_of_publication
ABC author
ABC title
ABC description
ABC url_link
ABC category_0
ABC category_1
ABC category_2
ABC category_3
ABC category_4
ABC category_5
ABC category_6

tmp_news2
ABC source_name
ABC date_of_publication
ABC author
ABC title
ABC description
ABC url_link
ABC category_0
ABC category_1
ABC category_2
ABC category_3
ABC category_4
ABC category_5
ABC category_6



view_news_summary
ABC category
ABC old_category
123 total
123 total_lenta
123 total_vedomosti
123 total_tass
123 last_total
123 last_total_lenta
123 last_total_vedomosti
123 last_total_tass
123 average_news_per_day
ABC max_date
123 total_mon
123 total_tue
123 total_wed
123 total_thu
123 total_fri
123 total_sat
123 total_sun

# Результаты разработки и выводы

На базе докер-контейнера airflow разработан программный продукт, позволяющий осуществлять сбор данных из трех новостных источников, сохранять их в виде .csv.gz файлов, загружать в PostgreSQL и формировать по расписанию витрину данных в виде материализованного представления.

В тестовом режиме разработана процедура предсказания категории (на базе kNN). В докере пока не удалось реализовать, т.к. по неизвестной причине sklearn не хочет подгружаться в докер (в отличие от других пакетов в requirements.txt)