

# PET-Project

Автоматизация аналитики



Анастасия Кузьменко

Анатолий Спесивцев

Кирилл Жуль

Презентация проекта

# Содержание

1

Обзор

5

2

Цели

6

3

Ключевые этапы

7

4

Технические подробности проекта и используемые  
компоненты

8

Презентация проекта

# Содержание

4.1	Описание слоёв в DWH	8
-----	----------------------	---

4.2	Описание данных в выгрузке из 1С	10
-----	----------------------------------	----

4.3	Порядок осуществления выгрузки данных	12
-----	---------------------------------------	----

4.4	Параметры доступа к административной панели AirFlow	14
-----	---	----

Презентация проекта

# Содержание

4.5

Параметры конфигурационного файла `docker-compose.yml`

14

4.6

Вопросы?

Презентация проекта

# Обзор

Данный учебный проект является рабочим прототипом системы автоматизации аналитики в компании “Ингеоком Дистрибьюшен Партнерс”

Презентация проекта

# Цели

- 1) Создание DWH (на базе PostgreSQL)
- 2) Налаживание процесса обработки данных (ELT) с использованием оркестратора Apache AirFlow
- 3) Визуализация агрегированных данных в DataLens согласно установленным правам доступа

# Ключевые этапы

## 1) Создание DWH.

Создание структуры необходимых слоев и таблиц. Установка ограничений и проверок

## 2) Создание ELT-пайплайнов в AirFlow.

Создание пайплайнов загрузки, обработки и транспортировки необходимых данных

## 3) Публикация сервиса в Интернете.

Сервер под управлением Linux Ubuntu 22.04. Сервисы развернуты в контейнерах Docker Compose.

Презентация проекта

# Технические подробности и используемые компоненты

## Описание слоёв в DWH

### 1. STG - слой

STG (Staging) – область подготовки данных. На этом слое данные загружаются “AS IS”.

### 2. DDS - слой

DDS (Data Detail Store) – область детальных данных. На этом слое данные распределяются по словарям и таблицам числовых показателей, с использованием инкрементальной загрузки (где это необходимо). Здесь данные преобразуются из типа данных varchar в нужные форматы типов данных (date, numeric etc.) В этом слое данные хранятся на том же уровне детализации, что и в системе источнике. Данные не агрегируются под нужды пользователей, никаких расчетов не производится.

Презентация проекта



# Технические подробности и используемые компоненты

## Описание слоёв в DWH

### 3. CDM - слой

CDM (Common Data Model) - Унифицированный слой данных, объединяющий данные разных таблиц-источников воедино. Именно данные этого уровня используются для построения ежедневной/еженедельной/ежемесячной отчетности и предоставляются конечным пользователям.

# Технические подробности и используемые компоненты

## Описание данных в выгрузке 1С

Old_sales/Sales	
<del>manager_id</del>	int4 null
manager	var null
client_id	var not null
client	var not null
sales_channel	var null
region	var null
order_date	var null
order_number	var null
realization_date	var null
realization_number	var null
<del>product_id</del>	var not null
item_number	var not null
product_name	var not null
brand	var null
count	int null
price	var null
total_sum	var null
comment	var null

Forecast	
manager	var not null
client_id	var not null
client	var not null
brand	var not null
general_plan	numeric(14, 2) null
week_1	numeric(14, 2) null
week_2	numeric(14, 2) null
week_3	numeric(14, 2) null
week_4	numeric(14, 2) null
week_5	numeric(14, 2) null

marketplaces	
manager	var not null
client_id	var not null
client	var not null
brand	var not null
total_realizations	numeric(14, 2) null

Product_category	
item_number	var not null
product_name	var not null
subbrand	var null
brand	var not null
product_category	var null

\*В блок-схемах зачеркнуты поля, которые выгружаются из 1С, но они не информативны по причине того, что в них присутствуют null, поэтому в работы мы их не использовали.

Презентация проекта

# Технические подробности и используемые компоненты

## Порядок осуществления выгрузки данных

- Первая выгрузка должна выгрузить все данные из 1С за период с 01.01.2016 по 31.12.2022. Эта выгрузка производится один раз.
- Вторая и последующие выгружают все данные из 1С за период с 01.01.2023 по текущую дату и время. Эта выгрузка производится 3 раза в день: в 7:05, 12:05, 16:05.

# Технические подробности и используемые компоненты

Параметры доступа  
к административной  
панели AirFlow

`http://91.107.126.62:8080/  
login: airflow_admin  
password: airflow123!@#`

Презентация проекта

# Технические подробности и используемые компоненты

## Параметры конфигурационного файла docker-compose.yml

```
x-airflow-common: #общие настройки ядра airflow
image: apache/airflow:2.7.1 #использовалась актуальная версия образа airflow
environment:
  AIRFLOW__CORE__LOAD_EXAMPLES: 'false' #отключили базовые примеры
  AIRFLOW__CORE__TEST_CONNECTION: 'Enabled' #включили опцию тестирования соединения п
ри создании Connection в Airflow
volumes: # тут мы делаем байнды к папкам, чтобы могли сохранять и модифицировать
необходимые нам файлы
  - /opt/compose/dags:/opt/airflow/dags
  - /opt/compose/logs:/opt/airflow/logs
  - /opt/compose/config:/opt/airflow/config
  - /opt/compose/plugins:/opt/airflow/plugins #данную папку мы решили использовать как
холдер для папки с общими файлами для межконтейнерного доступа
user: "${AIRFLOW_UID:-50000}:0" #пользователь 50000, он же указан в .env файле
```

Презентация проекта

# Технические подробности и используемые компоненты

## Параметры конфигурационного файла docker-compose.yml

services:

postgres\_database: #база postgresql версии 15 используется для хранилища DWH

image: postgres:15

environment:

POSTGRES\_USER: project\_user

POSTGRES\_PASSWORD: project\_password

POSTGRES\_DB: project\_db

expose:

- "5433" #транслируем порт во внешний мир, для подключения к базе по логин/паролю

ports:

- "5433:5432" #меняем стандартный порт БД на 5433

volumes:

- postgres\_database:/var/lib/postgresql/data:rw #сохраняем данные базы данных в отдельный volume, чтобы они не сбросились в случае рестарта контейнера

- /opt/compose/plugins:/opt:r #байндим общую папку к базе, чтобы мы могли прокидывать общие данные внутри контейнеров

Презентация проекта

# Технические подробности и используемые компоненты

## Параметры конфигурационного файла docker-compose.yml

```
postgres: #стандартные настройки внутренней БД для работы airflow
  image: postgres:13
  environment:
    POSTGRES_USER: airflow
    POSTGRES_PASSWORD: airflow
    POSTGRES_DB: airflow
  ports:
    - "5432:5432"
  volumes:
    - postgres:/var/lib/postgresql/data #сохраняем данные базы данных в отдельный volume,
    чтобы они не сбросились в случае рестарта контейнера
```

# Технические подробности и используемые компоненты

## Параметры конфигурационного файла docker-compose.yaml

airflow-webserver:

volumes: #байндим все нужные нам общие папки

- /opt/compose/dags:/opt/airflow/dags
- /opt/compose/logs:/opt/airflow/logs
- /opt/compose/config:/opt/airflow/config
- /opt/compose/plugins:/opt/airflow/plugins

expose:

- "8080"

ports:

- "8080:8080"

volumes: #создаем

postgres\_database:

postgres:

Презентация проекта



Вопросы?

Презентация проекта

**Спасибо за внимание!**

Презентация проекта