Тестовое задание

Data Engineer

Цель

Требуется:

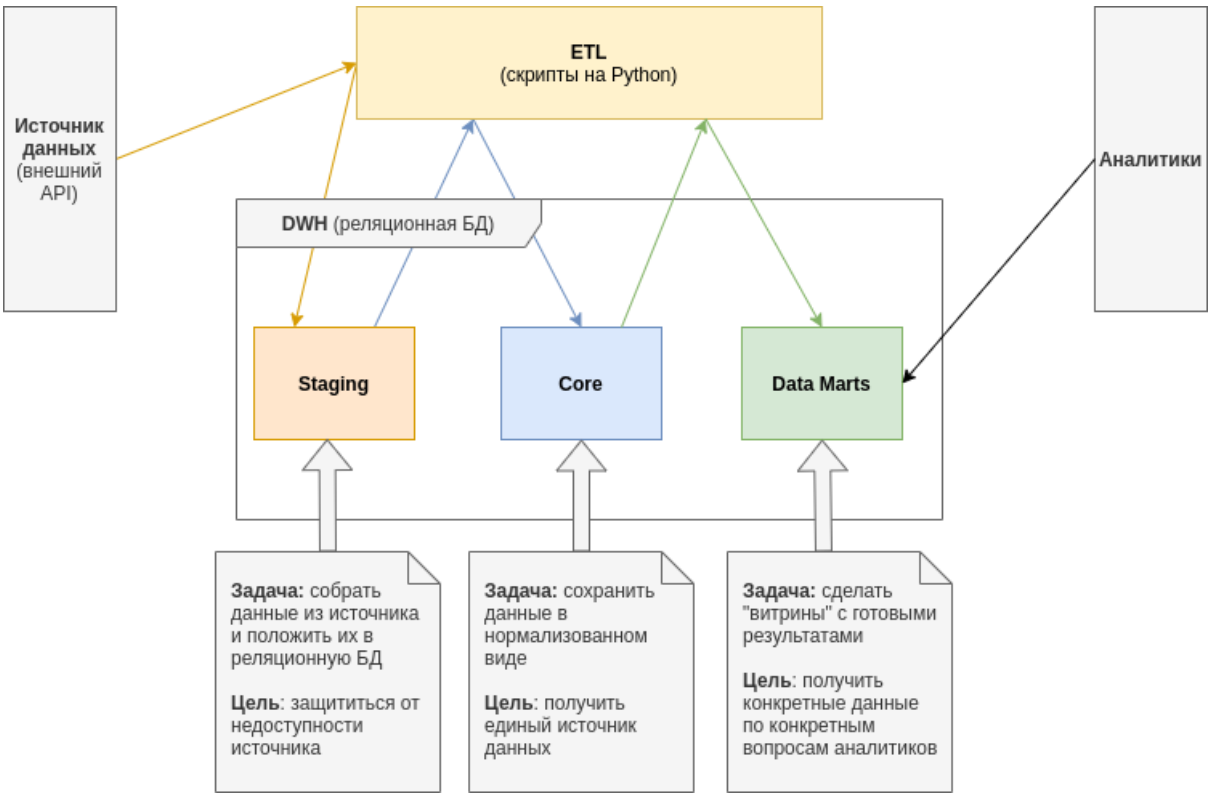
1. Спроектировать Data Warehouse (DWH) для хранения информации о курсах конвертации фиксированного списка валют
2. Написать на Python код, выполняющий ETL-процессы

Для проектирования нужно ознакомиться с моделями проектирования DWH и затем с помощью app.diagrams.net или другого редактора диаграмм нарисовать схему таблиц и связей в DWH.

Готовый проект вместе с рендером диаграммы в PNG нужно выложить на Github или другой хостинг для Git.

Что такое DWH и ETL

- DWH - это единое хранилище данных, например, для команды аналитики
- DWH может состоять из нескольких слоёв, чтобы достичь одновременно трёх целей:
    - обеспечить защиту от изменений в источниках данных
    - реализовать долговременное хранение данных
    - предоставить аналитикам удобные агрегаты, виртуальные таблицы и другие инструменты получения готовых данных
- ETL - это некий процесс, выполняющий по расписанию перенос данных из источников в DWH



Функциональные требования к DWH

DWH обеспечивает данные по курсам конвертации для фиксированного списка валют:

- Доллар (USD)
- Евро (EUR)
- Рубль (RUB)
- Юань (CNY)

Помимо этого, DWH предоставляет данные о названиях этих валют на четырёх языках: английский, немецкий, русский, китайский.

Слой Data Marts должен позволять простыми SQL запросами без JOIN с таблицами, не входящими в Data Marts, делать следующее:

1. получить курсы конвертации в рубли для всех валют (включая сам рубль) вместе с названиями валют на русском языке
2. получить курсы конвертации в доллары для всех валют вместе с названиями валют на английском языке
3. получить историю конвертации одной валюты в другую за выбранный период времени

Нефункциональные требования к DWH и ETL

Требования к DWH

1. СУБД выбирается разработчиком из двух вариантов: MySQL либо MS SQL
2. Все таблицы DWH находятся в одной схеме (database) выбранной СУБД

3. Уровень Staging может не быть вовсе, если же он есть, он проектируется произвольно
4. Уровень Core проектируется как набор таблиц с минимальным дублированием информации (некоторый разумный уровень дублирования допускается), при желании его можно проектировать по методологии Data Vault
    - См. Обзор гибких методологий проектирования DWH
    - См. Введение в Data Vault
    - См. Data Vault Cheatsheet
5. Уровень Data Marts на усмотрение разработчика может проектироваться по Star Schema, Snowflake Schema или в виде набора независимых таблиц, при этом можно использовать как физические таблицы (TABLE), так и виртуальные (VIEW)
6. Именование всех таблиц и полей должно соблюдать coding conventions, выбор конкретных coding conventions - на усмотрение разработчика
    - См. Learn SQL: Naming Conventions

Требования к ETL:

1. ETL разрабатывается в виде скриптов на Python 3
2. Для запуска задач по расписанию можно использовать Apache Airflow, Celery либо Cron
3. За счёт декомпозиции на функции и делегирования нужно добиться того, чтобы каждая функция на Python непосредственно (т.е. без учёта делегирования другим функциям) отвечала не более чем за одну из следующих задач:
    a. Взаимодействие с СУБД
    b. Взаимодействие с внешним API
    c. Преобразование данных
4. Для обработки ошибок следует использовать исключения

Требования к стилю кода:

1. Код на Python должен соблюдать правила PEP8 для именования (a.k.a. “pep8 naming”)
2. Код должен быть читаемым за счёт понятного именования и комментариев, при этом
    - Баланс именования/комментариев разработчик определяет сам
    - Все комментарии пишутся либо на русском, либо на английском языке (разработчик выбирает сам)
    - Все идентификаторы в коде пишутся на английском языке без транслита и сокращений (кроме общепринятых аббревиатур)