Домашнее задание

Дисциплина	Python для инженерии данных
Тема	Итоговое домашнее задание
Форма проверки	**Проверка преподавателем**
Имя преподавателя	Дмитрий Клабуков
Время выполнения	10 часов
Цель задания	Проверить знания и умения по дисциплине «Python для инженерии данных»
Инструменты для выполнения ДЗ	Airflow, Spark, Kafka, БД (например, PostgreSQL и MySQL)
Правила приёма работы	Прикрепите в LMS ссылку на выполненное задание в Google Colab или GitHub (если вы использовали Jupyter Notebook или IDE). **Важно:** убедитесь, что по ссылке есть доступ в Google Colab, так как иногда там закрыт доступ для другого логина
Критерии оценивания	**Максимальное количество баллов за итоговое задание — 10.** **Критерии оценивания итогового домашнего задания:** • развёрнуты 2 базы данных — 1 балл; • сгенерированы данные для БД-источника — 1 балл; • сформированы пайплайны для репликации в целевую базу данных с помощью Spark + Airflow — 1 балл; • сформированы пайплайны для формирования аналитических витрин с помощью Spark + Airflow — 1 балл; • аналитические витрины описаны в документации — 0,5 балла; • код Airflow проектов структурирован и читаем — 1 балл; • хранящиеся данные чистые: не имеют дублей, поддаются аналитике — 1 балл; • создано не менее 2 аналитических витрин, если добавляли собственные сущности, или не менее 1, если не добавляли — 1 балл; • предлагаемое решение дополнительно реализует необходимые функции или предлагает эффективное и творческое решение, демонстрирующее глубокое понимание инструментов обработки данных — 1,5 балла;

	Критерии оценивания дополнительного задания: • реализована генерация данных через брокер сообщений Kafka — 1 балл. **Итоговое домашнее задание не выполнено, если:** • файл с заданием не прикреплён или к нему нет доступа по ссылке; • код выдаёт ошибку или неправильный ответ
Дедлайн	**31.12.2024**

Задание

I. Генерация данных.

- Создайте реляционную базу данных, например PostgreSQL, и заполните её данными.

Примеры сущностей в базе данных

1. Users, пользователи:
   - user_id — уникальный идентификатор пользователя;
   - first_name — имя пользователя;
   - last_name — фамилия пользователя;
   - email — электронная почта;
   - phone — номер телефона;
   - registration_date — дата регистрации пользователя;
   - loyalty_status — статус лояльности: Gold, Silver и пр.
2. Products, товары:
   - product_id — уникальный идентификатор товара;
   - name — название товара;
   - description — описание товара;
   - category_id — идентификатор категории товара;
   - price — цена товара;
   - stock_quantity — количество товара на складе;
   - creation_date — дата добавления товара.
3. Orders, заказы:
   - order_id — уникальный идентификатор заказа;
   - user_id — идентификатор пользователя, который сделал заказ;
   - order_date — дата и время создания заказа;
   - total_amount — общая сумма заказа;
   - status — статус заказа: Pending, Completed и т. д.;
   - delivery_date — дата доставки заказа.
4. OrderDetails, детали заказов:
   - order_detail_id — уникальный идентификатор детали заказа;
   - order_id — идентификатор заказа;

- product_id — идентификатор товара;
- quantity — количество товара в заказе;
- price_per_unit — цена за единицу товара;
- total_price — общая стоимость товара в заказе, количество товаров, умноженное на цену единицы товара.

5. ProductCategories, категории товаров:

   - category_id — уникальный идентификатор категории;
   - name — название категории;
   - parent_category_id — идентификатор родительской категории, его может не быть.

II. Репликация данных.

Настройте репликацию сгенерированных данных из PostgreSQL в MySQL с использованием Airflow.

III. Построение аналитических витрин.

На основе данных из MySQL создайте минимум одну аналитическую витрину, если добавляете собственные сущности, не менее двух аналитических витрин.

Опишите в документации эти аналитические витрины, их поля и предназначение.

Пример витрины:

- витрина активности пользователей для анализа поведения пользователя сервиса;

IV.Использование Airflow.

Настройте пайплайны для репликации данных и формирования аналитических витрин.

Дополнительное задание (выполняется по желанию)

- Настройте Spark для чтения данных из Kafka, их обработки и записи в базу данных PostgreSQL.