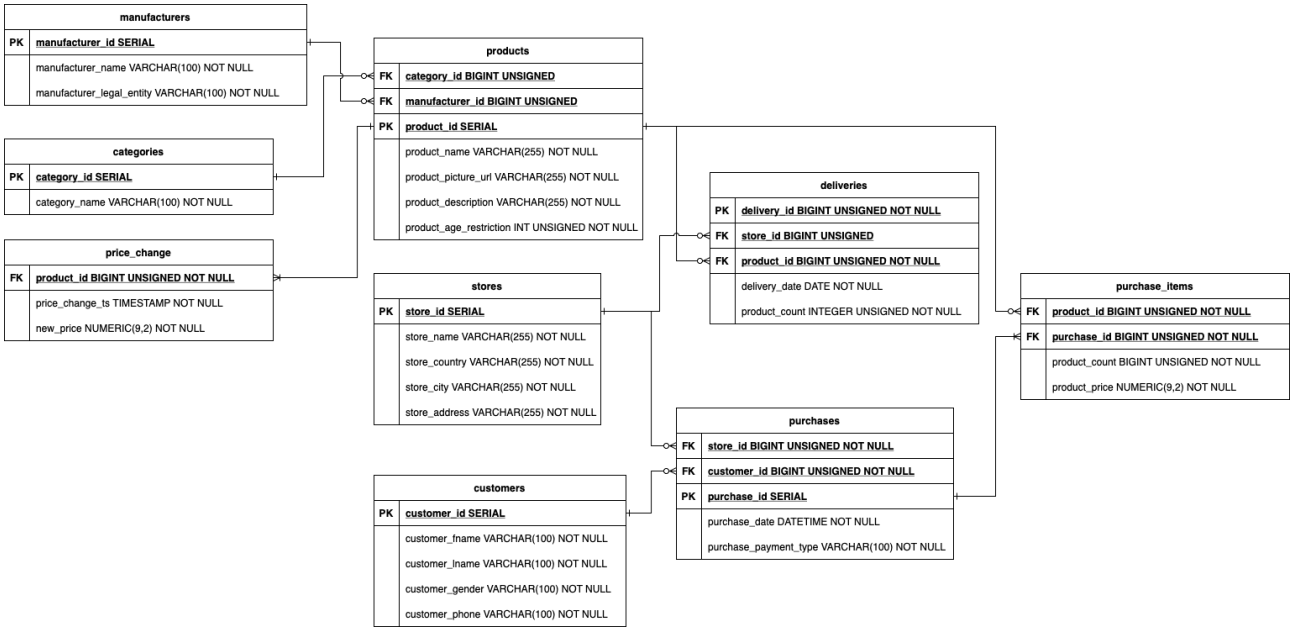
Домашнее задание 3

Important notice

1) **Вы не сможете приступить к выполнению ДЗ №3, пока не выполните хотя бы минимум из ДЗ №2**. Поэтому если вы в ДЗ №2 не перевезли данные в аналитический контур - делать задание №3 нет никакого смысла.
2) **Это задание выполняется и оценивается индивидуально!** Если вы используете код из открытых источников (репозитории ваших одногруппников таковыми не считаются) - пожалуйста, указывайте ссылки на них (можно в readme вести лог всех источников, откуда вы берете код). Находить готовые рецепты в интернете - хорошо, списывать - плохо.
3) **При обнаружении списывания (одинаковый код в двух репозиториях без указания внешнего источника) оценка будет выставляться студенту, чей коммит с решением был первый**. Остальным - 0 баллов и докладная в УО.
4) Чтобы исключить возможность списывания, **рекомендуется сделать ваш репозиторий с домашним заданием приватным**.

Формулировка

У нас есть сеть магазинов.
БД системы, которая обеспечивает её работу, выглядит следующим образом:



Это мы уже с вами видели в ДЗ №1 и №2.
Сейчас мы имеем:
* master-хост с БД
* async-replica, на которую копируются данные для аналитической нагрузки
* пайплайн доставки данных в аналитическую БД
* детальный слой DWH в аналитической БД

Задача:

1. **Поднять Apache Airflow в docker-compose.** Идеально - если все приложения будут запускаться одной командой docker-compose up (можно притащить нужные сервисы руками, или использовать конструкцию extends - **https://docs.docker.com/compose/multiple-compose-files/extends/**)
2. Создайте DAG для Airflow, который используя данные детального слоя DWH собирает следующие витрины:

- **Описание**: китами в e-commerce называют пользователей, которые совершили больше всего покупок (и, соотвественно, приносят наибольшую выручку как индивидуальные покупатели)
- **Нужно**: собрать витрину, в которой будут следующие данные:
  - created_at - момент (timestamp) обновления
  - customer_id - ID клиента
  - customer_gmv - сумма покупок клиента за предыдущие дни (не включая created_at)
  - customer_category - категория товаров, которую этот клиент покупает больше всего (по сумме потраченных денег)
  - customer_group -
    - 5 - если покупатель входит в топ-5% по gmv
    - 10 - … в топ-10% …
    - 25 - … в топ-25% …
    - 50 - … в топ-50% …
    - 50+ - остальные
- **Автоматизировать** процесс обновления витрины с помощью Airflow; Витрина должна полностью обновляться 1 раз в день (в любое время);
- Витрина должна лежать в схеме **presentation** в вашей аналитической БД;

—

- **Описание**: витрина для подсчета GMV в разрезе товарных категорий по дням.
- **Нужно**: собрать витрину, в которой будут следующие данные:
  - created_at - момент (timestamp) обновления
  - business_date - дата, за которую собраны продажи
  - category_name - название товарной категории
  - category_gmv - gmv по категории за этот день. Если продаж по категории за этот день не было - должна быть строка со значением 0;
- **Автоматизировать** процесс обновления витрины с помощью Airflow; Витрина должна обновлять данные о продажах за business_date на следующий за ним день (каждый день обновляем за вчера). При перезапуске расчета за уже существующий день предыдущие данные должны удаляться, чтобы избежать дублей;
- Витрина должна лежать в схеме **presentation** в вашей аналитической БД;

Бонусные задания:
1. Использовать для ETL dbt

Сроки

1. Дедлайн на 100% - 2 недели - **03.12.2023 23:59:59 включительно**
2. Дедлайн на 75% - до дедлайна следующего ДЗ - **17.12.2023 23:59:59 включительно**
3. Дедлайн на 50% - до конца курса - **20.12.2023 23:59:59 включительно**
4. После **20.12.2023** работы **не принимаются**

Как сдавать ДЗ?

- Готовое ДЗ загружается на GitHub (приватный репозиторий, для проверки предоставить доступ @mgcrp)
- Домашнее задание №3 можно продолжать делать в том же репозитории, что и домашнее задание №1 и №2
- К репозиторию должен быть приложен README с описанием того, что вы сделали и как это запустить
- Задание сдается в форму: https://forms.gle/ZNZ2mQks3tKhfXqq6

Критерии оценки

Балл	**Критерий**
4	В структуру данных из ДЗ №1 добавлены новые поля
7	

10	DDL для детального слоя DWH ER-диаграмма Поднят новый инстанс PostgreSQL для DWH
+4 балла	Реализовать DMP как один универсальный класс и yaml-конфиги вместо отдельных классов для каждой таблицы

Максимальный балл за ДЗ - 14

Как это будет проверяться?

1) **Запуск системы по инструкции** из вашего README
2) **Проверка наличия и работоспособности DAG'ов** в Airflow
3) **Проверка наличия и корректности** данных в витринах в CDM-слое аналитической БД