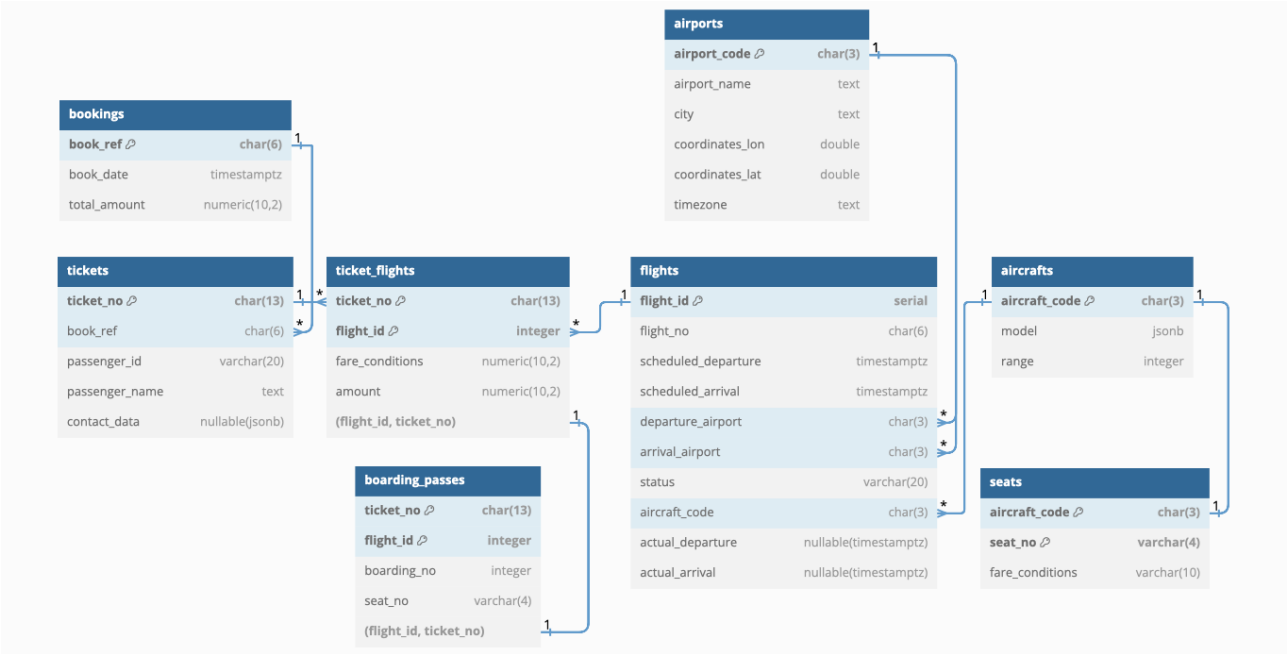
Домашнее задание 3

Important notice

1) **К ДЗ №3 можно приступить, не сдав ДЗ №2 полностью.** Если у вас нет данных в DDS-слое DWH, вы можете использовать для сборки ETL сырые данные (напрямую из исходных таблиц). Складывать витрины можно будет не в аналитическую БД, а в отдельную схему на проде.
2) **Это задание выполняется и оценивается индивидуально или в группах до 4 человек!** 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 (группе), 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 поисковик дешевых авиабилетов.
БД системы, которая содержит информацию о забронированных с его помощью билетах, 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**: frequent flyer'ами в авиации называют людей, которые часто пользуются услугами авиакомпании для деловых и/или личных поездок (и, соотвественно, приносят наибольшую выручку как индивидуальные покупатели)
- **Нужно**: собрать витрину, в которой будут следующие данные:
  - created_at - момент (timestamp) обновления
  - passenger_id - ID пассажира
  - passenger_name - Имя пассажира
  - flights_number - Общее кол-во совершенных перелетов
  - purchase_sum - Сумма денег, которую пассажир потратил на перелеты
  - home_airport - код аэропорта, из которого/в который пассажир совершил наибольшее кол-во поездок (если таких несколько - выберите первый по алфавиту)
  - 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 (или продовой) БД

—

- **Описание**: витрина для подсчета пассажиропотока из аэропорта/в аэропорт по дням
- **Нужно**: собрать витрину, в которой будут следующие данные:
  - created_at - момент (timestamp) обновления
  - flight_date - дата по actual_arrival / actual_departure
  - airport_code - рассматриваемый аэропорт
  - linked_airport_code - второй аэропорт в паре
  - flights_in - количество прилетевших рейсов из linked_airport_code в airport_code
  - flights_out - количество прилетевших рейсов из airport_code в linked_airport_code
  - passengers_in - количество прилетевших пассажиров из linked_airport_code в airport_code
  - passengers_out - количество прилетевших пассажиров из airport_code в linked_airport_code
- **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 (или продовой) БД

Бонусные задания:

1. Использовать для ETL dbt

Сроки

1. Дедлайн на 100% - 2 недели - **23.02.2024 23:59:59 включительно**
2. Дедлайн на 75% - до дедлайна следующего ДЗ
3. Дедлайн на 50% - до конца курса - **30.03.2024 23:59:59 включительно**

Как сдавать ДЗ?

- Готовое ДЗ загружается на GitHub (в приватный репо + выдать мне доступ - @mgcrp)
- Домашнее задание №3 можно продолжать делать в том же репозитории, что и домашнее задание №2 и №1 (можно для связанности отвести отдельную ветку)
- К репо должен быть приложен README.md с описанием того, что вы сделали и как это запустить
- Задание сдается в форму: https://forms.gle/vebgRRu44imsVuVr8

Критерии оценки

Балл	**Критерий**
4	Поднят AirFlow
7	Реализован ETL для одной из витрин
10	Реализован ETL для обоих из витрин
+4 балла	Для реализации ETL использован dbt

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