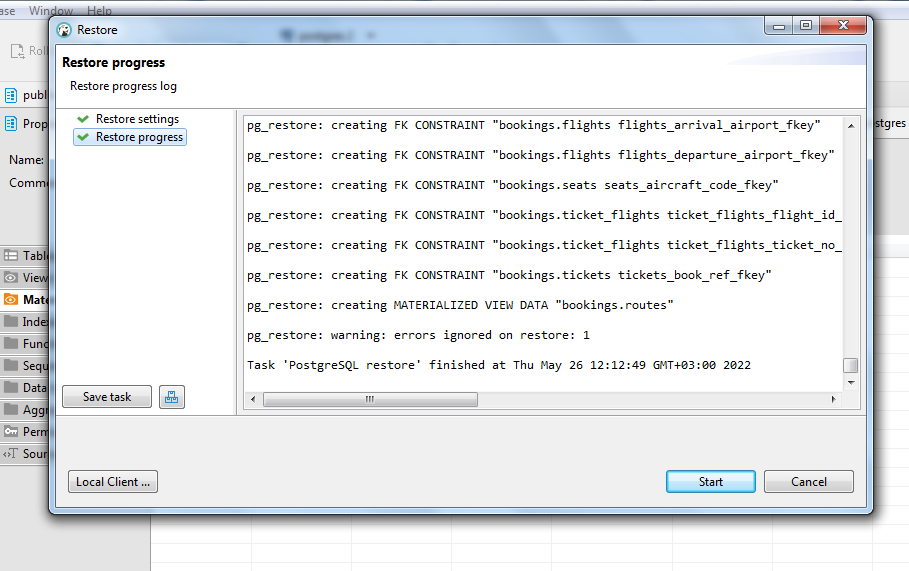
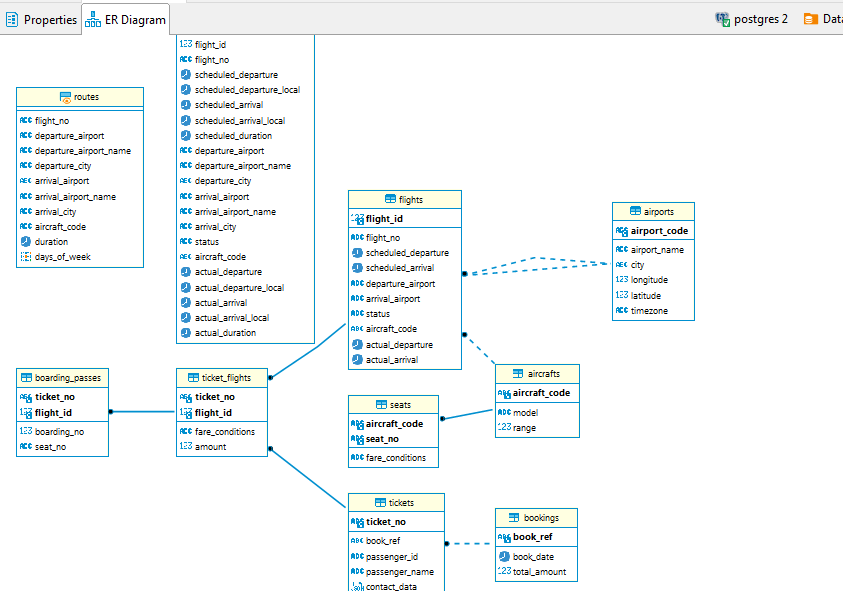
1. В работе использовался локальный тип подключения.

* если база была развернута из \*.sql или \*.backup файла, необходимо приложить скриншот успешного импорта или восстановления



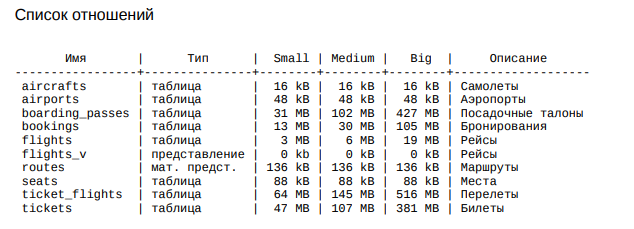
**Restore Complete**

1. Скриншот ER-диаграммы из DBeaver`a согласно Вашего подключения.



1. Краткое описание БД - из каких таблиц и представлений состоит.

База данных состоит из восьми таблиц и двух представлений. В качестве предметной области выбраны авиаперевозки по России.



1. Развернутый анализ БД - описание таблиц, логики, связей и бизнес области (частично можно взять из описания базы данных, оформленной в виде анализа базы данных).

* **Таблица bookings.aircrafts**

Каждая модель воздушного судна идентифицируется своим трехзначным кодом (aircraft\_code). Название модели (model). Максимальная дальность полета в километрах (range)

* **Таблица bookings.airports**

Трехбуквенный код аэропорта (airport\_code), имя (airport\_name). Город, в котором аэропорт расположен (city) . Широта (longitude), долгота (latitude) и часовой пояс (timezone).

* **Таблица bookings.boarding\_passes**

При регистрации на рейс, которая возможна за сутки до плановой даты отправления, пассажиру выдается посадочный талон. Он идентифицируется также, как и перелет — номером билета и номером рейса. Посадочным талонам присваиваются последовательные номера (boarding\_no) в порядке регистрации пассажиров на рейс (этот номер будет уникальным только в пределах данного рейса). В посадочном талоне указывается номер места (seat\_no).

* **Таблица bookings.bookings**

Пассажир заранее (book\_date, максимум за месяц до рейса) бронирует билет себе и, возможно, нескольким другим пассажирам. Бронирование идентифицируется номером (book\_ref, шестизначная комбинация букв и цифр). Поле total\_amount хранит общую стоимость включенных в бронирование перелетов всех пассажиров.

* **Таблица bookings.flights**

Естественный ключ таблицы рейсов состоит из двух полей — номера рейса (flight\_no) и даты отправления (scheduled\_departure). Чтобы сделать внешние ключи на эту таблицу компактнее, в качестве первичного используется суррогатный ключ (flight\_id). Рейс всегда соединяет две точки — аэропорты вылета (departure\_airport) и прибытия (arrival\_airport). Такое понятие, как «рейс с пересадками» отсутствует: если из одного аэропорта до другого нет прямого рейса, в билет просто включаются несколько необходимых рейсов. У каждого рейса есть запланированные дата и время вылета (scheduled\_departure) и прибытия (scheduled\_arrival). Реальные время вылета (actual\_departure) и прибытия (actual\_arrival) могут отличаться: обычно не сильно, но иногда и на несколько часов, если рейс задержан. Статус рейса (status) может принимать одно из следующих значений: • Scheduled Рейс доступен для бронирования. Это происходит за месяц до плановой даты вылета; до этого запись о рейсе не существует в базе данных. • On Time Рейс доступен для регистрации (за сутки до плановой даты вылета) и не задержан. • Delayed Рейс доступен для регистрации (за сутки до плановой даты вылета), но задержан. • Departed Самолет уже вылетел и находится в воздухе. Arrived Самолет прибыл в пункт назначения. • Cancelled Рейс отменен

* **Таблица bookings.seats**

Места определяют схему салона каждой модели. Каждое место определяется своим номером (seat\_no) и имеет закрепленный за ним класс обслуживания (fare\_conditions) — Economy, Comfort или Business.

* **Таблица bookings.ticket\_flights**

Перелет соединяет билет с рейсом и идентифицируется их номерами. Для каждого перелета указываются его стоимость (amount) и класс обслуживания (fare\_conditions).

* **Таблица bookings.tickets**

Билет имеет уникальный номер (ticket\_no), состоящий из 13 цифр. Билет содержит идентификатор пассажира (passenger\_id) — номер документа, удостоверяющего личность, — его фамилию и имя (passenger\_name) и контактную информацию (contact\_date). Ни идентификатор пассажира, ни имя не являются постоянными (можно поменять паспорт, можно сменить фамилию), поэтому однозначно найти все билеты одного и того же пассажира невозможно

* **Представление "bookings.flights\_v"**

Содержит дополнительную информацию: • расшифровку данных об аэропорте вылета (departure\_airport, departure\_airport\_name, departure\_city), • расшифровку данных об аэропорте прибытия (arrival\_airport, arrival\_airport\_name, arrival\_city), • местное время вылета (scheduled\_departure\_local, actual\_departure\_local), • местное время прибытия (scheduled\_arrival\_local, actual\_arrival\_local), • продолжительность полета (scheduled\_duration, actual\_duration).

* **Материализованное представление bookings.routes**

Таблица рейсов содержит избыточность: из нее можно было бы выделить информацию о маршруте (номер рейса, аэропорты отправления и назначения), которая не зависит от конкретных дат рейсов. Именно такая информация и составляет материализованное представление routes.

Бизнес задачи, которые можно решить, используя БД.

- Анализ наиболее популярных направлений для путешествий, в зависимости от даты(возможно для настройки таргетированной рекламы или регулирования цен на билеты)

- Получать данные о задержанных рейсах. Выявлять и устранять проблемы.

- Получать данные о загруженности аэропортов для оптимизации пассажиропотоков.

- Имея информацию о моделях самолетов, совершающих перелеты по определенным направлениям, можно сделать выводы о наличии взлетно-посадочных полос, под определенные типы воздушных судов, а также о наличии наземного оборудования для их обслуживания. Так же это позволит в случае возникновения внештатной ситуации, объединить два рейса по одному направлению, к примеру, на малых самолетах Boeing 737 в один рейс с использованием Boeing 777.

1. Список SQL запросов из приложения №2 с описанием логики их выполнения.

