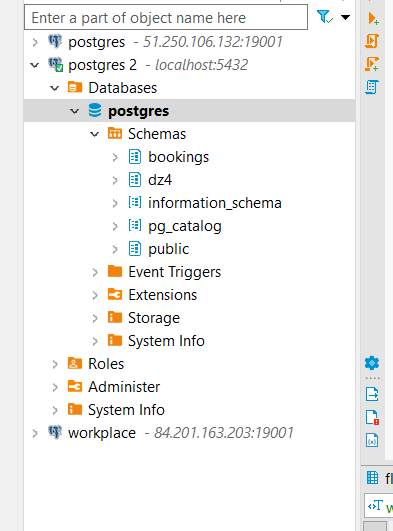
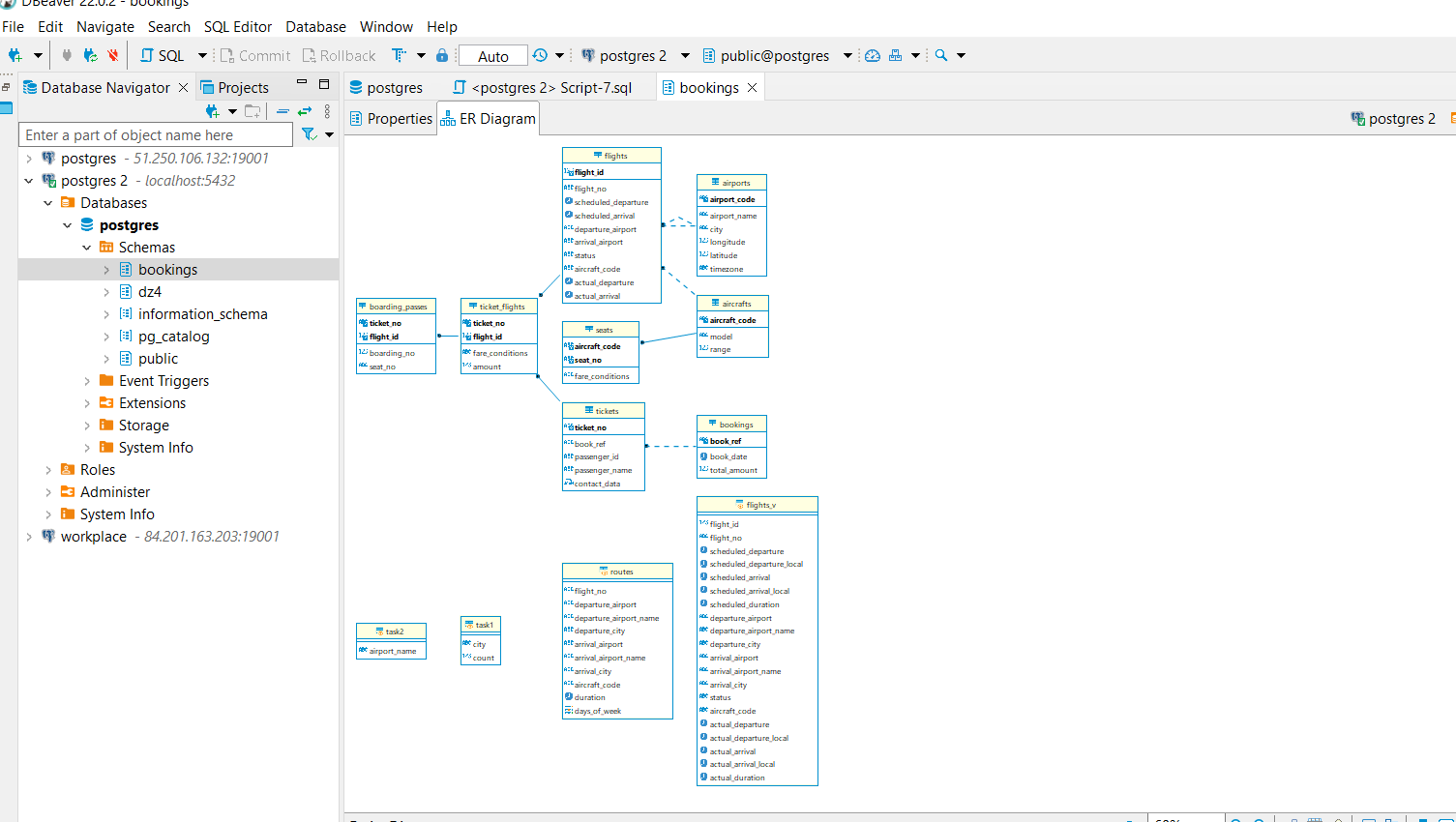
**Итоговая работа**

1. В работе использовался локальный тип подключения.



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



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

База данных содержит в себе 8 таблиц:

1. aircrafts;
2. airports;
3. boarding\_passes:
4. bookings;
5. flights;
6. seats;
7. ticket\_flights;
8. tickets.

А также 1 обычное представление flights\_v и 1 материализованное представление routes.

4. Развернутый анализ БД - описание таблиц, логики, связей и бизнес области (частично можно взять из описания базы данных, оформленной в виде анализа базы данных). Бизнес задачи, которые можно решить, используя БД.

Основной сущностью является бронирование (bookings). В одно бронирование можно включить несколько пассажиров, каждому из которых выписывается отдельный билет (tickets). Билет имеет уникальный номер и содержит информацию о пассажире. Как таковой пассажир не является отдельной сущностью. Как имя, так и номер документа пассажира могут меняться с течением времени, так что невозможно однозначно найти все билеты одного человека; для простоты можно считать, что все пассажиры уникальны. Билет включает один или несколько перелетов (ticket\_flights). Несколько перелетов могут включаться в билет в случаях, когда нет нет прямого рейса, соединяющего пункты отправления и назначения (полет с пересадками), либо когда билет взят «туда и обратно». В схеме данных нет жесткого ограничения, но предполагается, что все билеты в одном бронировании имеют одинаковый набор перелетов. Каждый рейс (flights) следует из одного аэропорта (airports) в другой. Рейсы с одним номером имеют одинаковые пункты вылета и назначения, но будут отличаться датой отправления. При регистрации на рейс пассажиру выдается посадочный талон (boarding\_passes), в котором указано место в самолете. Пассажир может зарегистрироваться только на тот рейс, который есть у него в билете. Комбинация рейса и места в самолете должна быть уникальной, чтобы не допустить выдачу двух посадочных талонов на одно место. Количество мест (seats) в самолете и их распределение по классам обслуживания зависит от модели самолета (aircrafts), выполняющего рейс. Предполагается, что каждая модель самолета имеет только одну компоновку салона. Схема данных не контролирует, что места в посадочных талонах соответствуют имеющимся в самолете (такая проверка может быть сделана с использованием табличных триггеров или в приложении).

Список столбцов в рамках каждого отношения представлен в таблице.

|  |  |
| --- | --- |
| **Таблица (отношение)** | **Столбцы** |
| bookings.aircrafts | aircraft\_code  model  range |
| bookings.airports | airport\_code  airport\_name  city  longitude  latitude  timezone |
| bookings.boarding\_passes | ticket\_no  flight\_id  boarding\_no  seat\_no |
| bookings.bookings | book\_ref  book\_date  total\_amount |
| bookings.flights | flight\_id  flight\_no  scheduled\_departure  scheduled\_arrival  departure\_airport  arrival\_airport  status  aircraft\_code  actual\_departure  actual\_arrival |
| bookings.seats | aircraft\_code  seat\_no  fare\_conditions |
| bookings.ticket\_flights | ticket\_no  flight\_id  fare\_conditions  amount |
| bookings.tickets | ticket\_no  book\_ref  passenger\_id  passenger\_name  contact\_data |

Так, каждая модель воздушного судна идентифицируется своим трехзначным кодом (aircraft\_code). Указывается также название модели (model) и максимальная дальность полета в километрах (range). Аэропорт идентифицируется трехбуквенным кодом (airport\_code) и имеет свое имя (airport\_name). Для города не предусмотрено отдельной сущности, но название (city) указывается и может служить для того, чтобы определить аэропорты одного города. Также указывается широта (longitude), долгота (latitude) и часовой пояс (timezone). 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).

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.

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.

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.

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

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.

**Бизнес задачи, которые можно решить, используя БД:**

1) Вычислить загруженность каждого из аэропортов, проходимость аэропортов.

2) Узнать наиболее и наименее популярные маршруты пользователей услуг.

3) Вычислить наиболее часто используемые типы самолетов, их загруженность.

4) Определить выручку на каждый самолет за определенный промежуток времени для дальнейшего расчета с добавлением доп. данных по затратам, какие наиболее окупаемы с точки зрения их обслуживания, например.

5) Определить маршруты, самолеты, города вылета, по которым наибольший процент отмены бронирований.

6) Определить равномерность осуществления посадки пассажиров с учетом требований к лучшей эксплуатации самолета.

**Логика выполнения запросов содержится в самом SQL-файле.**