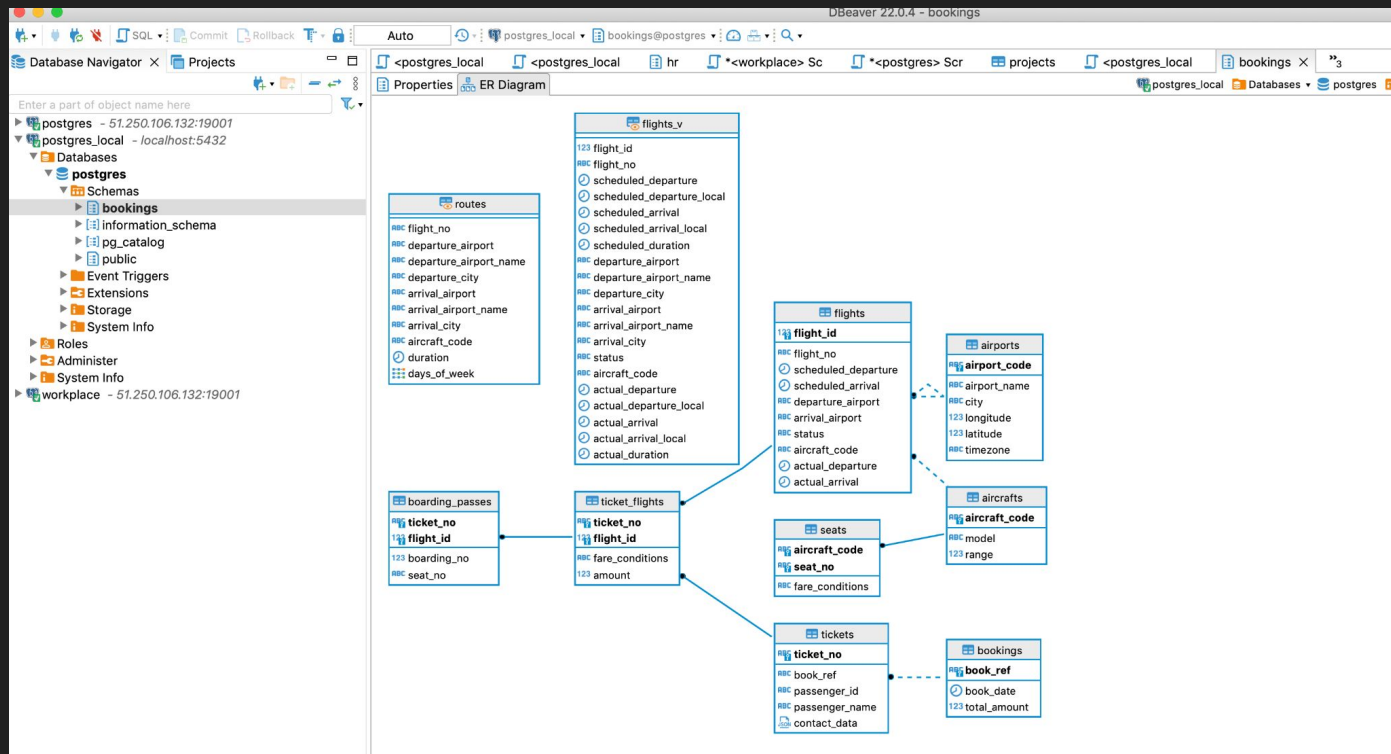


# Проектная работа по модулю “SQL и получение данных”

Михальченко Вера DSU-30

# ER-диаграмма



# Краткое описание БД

## Таблицы:















- 8 (aircrafts, airports, boarding\_passes, bookings, flights, seats, ticket\_flights, tickets)

## Представления:

- 1 ( flights\_v)

## Материализованные представления:

- 1 (routes)

▼  <b>bookings</b>	
▼  Tables	
▶  aircrafts	32K
▶  airports	64K
▶  boarding_passes	80M
▶  bookings	18M
▶  flights	4.8M
▶  seats	136K
▶  ticket_flights	108M
▶  tickets	59M
▼  Views	
▶  flights_v	
▼  Materialized Views	
▶  routes	144K

# Развёрнутый анализ 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 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.

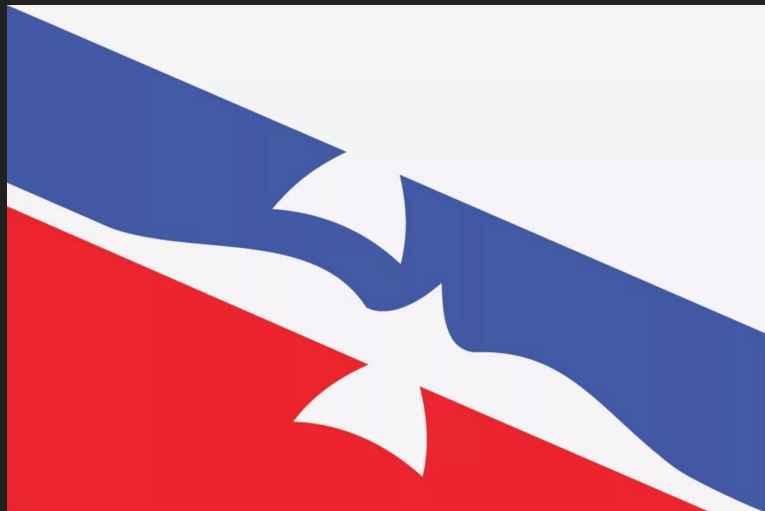
При регистрации на рейс пассажиру выдается *посадочный талон* (boarding\_passes), в котором указано место в самолете. Пассажир может зарегистрироваться только на тот рейс, который есть у него в билете. Комбинация рейса и места в самолете должна быть уникальной, чтобы не допустить выдачу двух посадочных 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).

# Бизнес задачи, которые можно решить

## 1. Аналитика для РосТуризма.

Прогноз динамики спроса на покупку билетов в определённых направлениях внутри России, в связи с ограничениями по возможностям международного туризма.



# Бизнес задачи, которые можно решить

2. Аналитика для агентства по продажам билетов.

Тестирование рекламной кампании, в рамках которой пользователям предлагается в приложении по покупке билетов вариант популярного направления по внутреннему перелёту в зависимости от времени года и цены на билет.



# Бизнес задачи, которые можно решить

## 3. Кризисная аналитика.

Оптимизация распределения самолетов по направлениям в связи с новыми ограничениями по импорту запчастей.



# Бизнес задачи, которые можно решить

4. Аналитика для стартапа по созданию новых туристических маршрутов.

Анализ загруженности самолётов, летающих в направлениях, где планируется создание новых туристических маршрутов.





# Бизнес задачи, которые можно решить

5. Аналитика для компании по созданию бизнес зон в аэропортах.

Анализ спроса на покупку билетов бизнес-класса по направлениям и его динамика.



# Список SQL запросов с описанием логики их выполнения

Ссылка на файл с запросами [ТУТ](#).

```
1  -- Итоговая работа Михальченко Вера DSU – 30
2
3  SET search_path TO bookings;
4
5  -- 1. В каких городах больше одного аэропорта?
6
7  select
8      city,
9      count (city) -- считаю количество городов
10 from airports
11 group by 1 -- группирую по городам, потому что есть агрегатная функция
12 having count (city) > 1;
13
```

Спасибо!  
Было интересно!