

Итоговая работа по курсу SQL

1. Тип подключения и ER-диаграмма

База была развернута локально из *.backup файла, см. рис 1. и рис. 2.

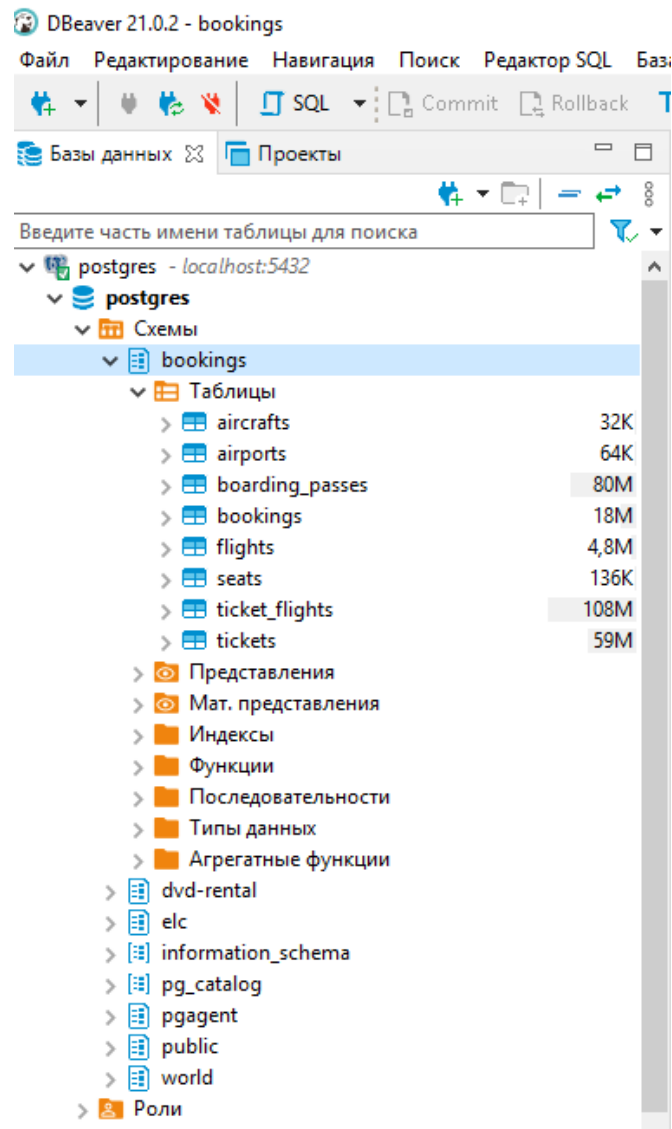


Рис. 1. Исследуемая БД.

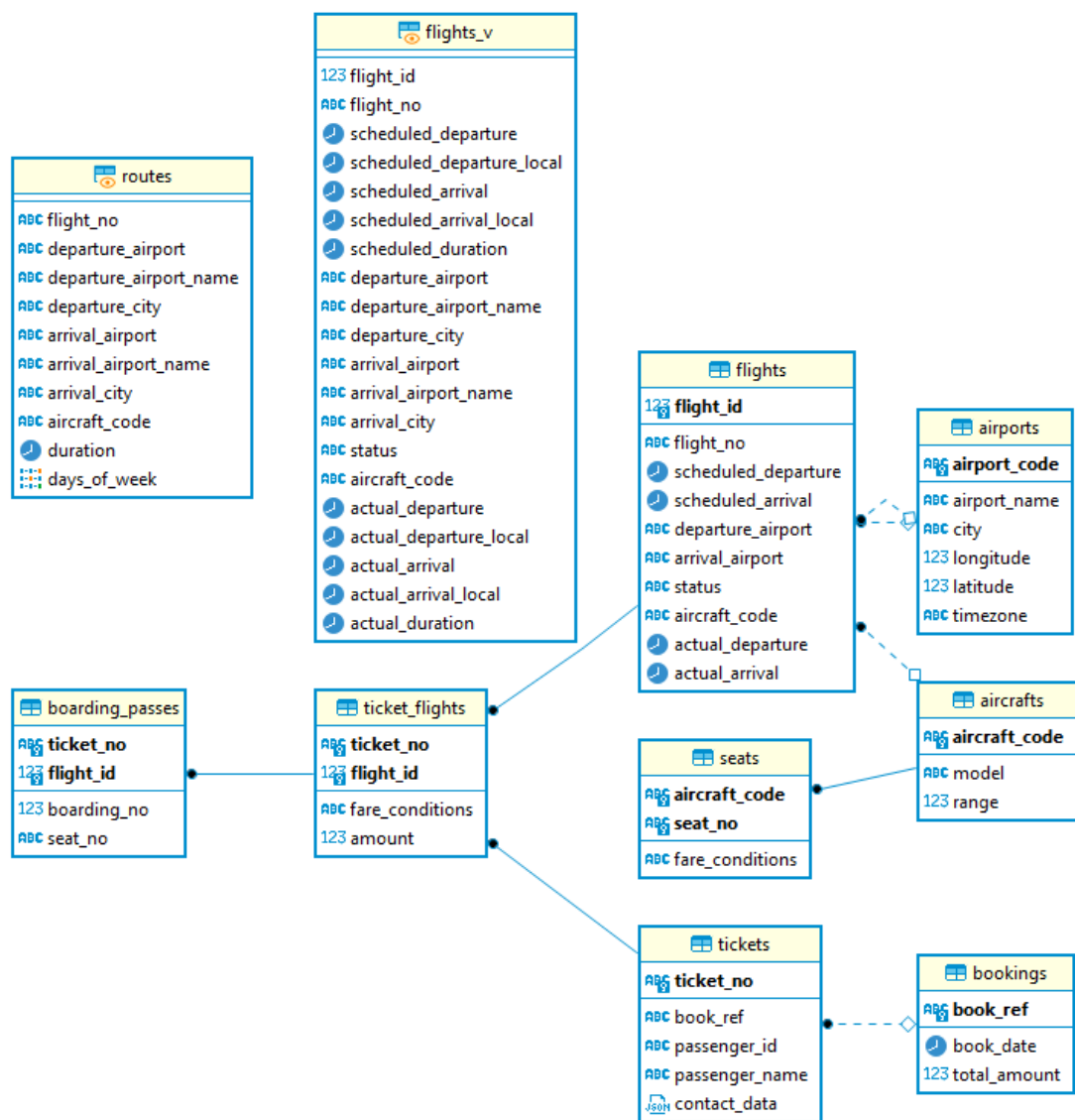


Рис. 2. ER-диаграмма БД.

2. Краткое описание БД

БД содержит следующие отношения (я использовал малую версию базы, опасаясь тормозов).

Имя	Тип	Small	Описание
aircrafts	таблица	16 kB	Самолеты
airports	таблица	48 kB	Аэропорты
boarding_passes	таблица	31 MB	Посадочные талоны
bookings	таблица	13 MB	Бронирования
flights	таблица	3 MB	Рейсы
flights_v	представление	0 kb	Рейсы
routes	мат. предст.	136 kB	Маршруты
seats	таблица	88 kB	Места
ticket_flights	таблица	64 MB	Перелеты
tickets	таблица	47 MB	Билеты

Как изложено в описании БД, основной сущностью является бронирование (bookings). В одно бронирование можно включить нескольких пассажиров, каждому из которых выписывается отдельный билет (tickets). Пассажир не является однозначно идентифицируемым, так как его данные могут со временем изменяться.

Билет включает один или несколько перелетов (ticket_flights). Билет имеет уникальный ID и содержит информацию о пассажире. 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». 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.

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.

3. Развернутый анализ БД

В принципе что с чем связано – очевидно из анализа ER-диаграммы.

Таблица aircrafts описывает доступные модели самолетов, каждая модель идентифицируется кодом (aircraft_code). Указывается также название модели (model) и максимальная дальность полета в километрах (range)

Таблица airports описывает аэропорты, аналогично каждый аэропорт идентифицируется 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).

Таблица 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. 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 `flights`. Естественный ключ таблицы рейсов состоит из двух полей — номера рейса (`flight_no`) и даты отправления (`scheduled_departure`). В качестве первичного используется суррогатный ключ (`flight_id`). Рейс всегда соединяет две точки — аэропорты вылета (`departure_airport`) и прибытия (`arrival_airport`). 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. Статус рейса (`status`) может принимать одно из следующих значений:

1. `Scheduled` Рейс доступен для бронирования. Это происходит за месяц до плановой даты вылета; до этого запись о рейсе не существует в базе данных.
2. `On Time` Рейс доступен для регистрации (за сутки до плановой даты вылета) и не задержан.
3. `Delayed` Рейс доступен для регистрации (за сутки до плановой даты вылета), но задержан.
4. `Departed` Самолет уже вылетел и находится в воздухе.
5. `Arrived` Самолет прибыл в пункт назначения
6. `Cancelled` Рейс отменен

Таблица `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 `ticket_flights`. Перелет соединяет билет с рейсом и идентифицируется их номерами. Для каждого перелета указываются его стоимость (`amount`) и класс обслуживания (`fare_conditions`).

Таблица `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_data`).

Также БД содержит одно представление, одно материализованное представление и одну функцию.

Представление `flights_v`. Оно 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 `routes`. Таблица рейсов содержит избыточность: из нее можно было бы выделить информацию о маршруте,

которая не зависит от конкретных дат рейсов. Именно такая информация и составляет материализованное представление routes.

Функция `bookings.now()`. БД содержит временной срез данных — так, как будто в некоторый момент была сделана резервная копия реальной системы. Позиция среза сохранена в функции `bookings.now()`. Ей можно пользоваться в запросах там, где обычно использовалась бы функция `now()`. Кроме того, значение этой функции определяет версию БД.

Собственно данная база может решить практически любые бизнес-задачи, связанные с обработкой билетов для авиаперевозок.

Во-первых, естественно, можно создать записи о бронировании билета для любых возможных комбинаций доступных аэропортов, самолетов, классов обслуживания и количества пассажиров, в прямом и обратном направлении. Далее можно получить всю информацию о текущих рейсах, находящихся в воздухе, отмененных и так далее. Можно получить информацию о всех свободных местах на конкретную дату для любого доступного рейса, класса обслуживания и т.п. Можно получить информацию о задержанных и отмененных рейсах, о наличии прямого сообщения между указанными точками, о числе пересадок, которые необходимо сделать и т.п. Можно найти ближайшие рейсы (или рейсы на любую дату) из любой точки в любую точку.

В общем и целом, данная БД представляет собой типичный инструмент для работы в бизнесе авиаперевозок и онлайн-бронировании билетов.