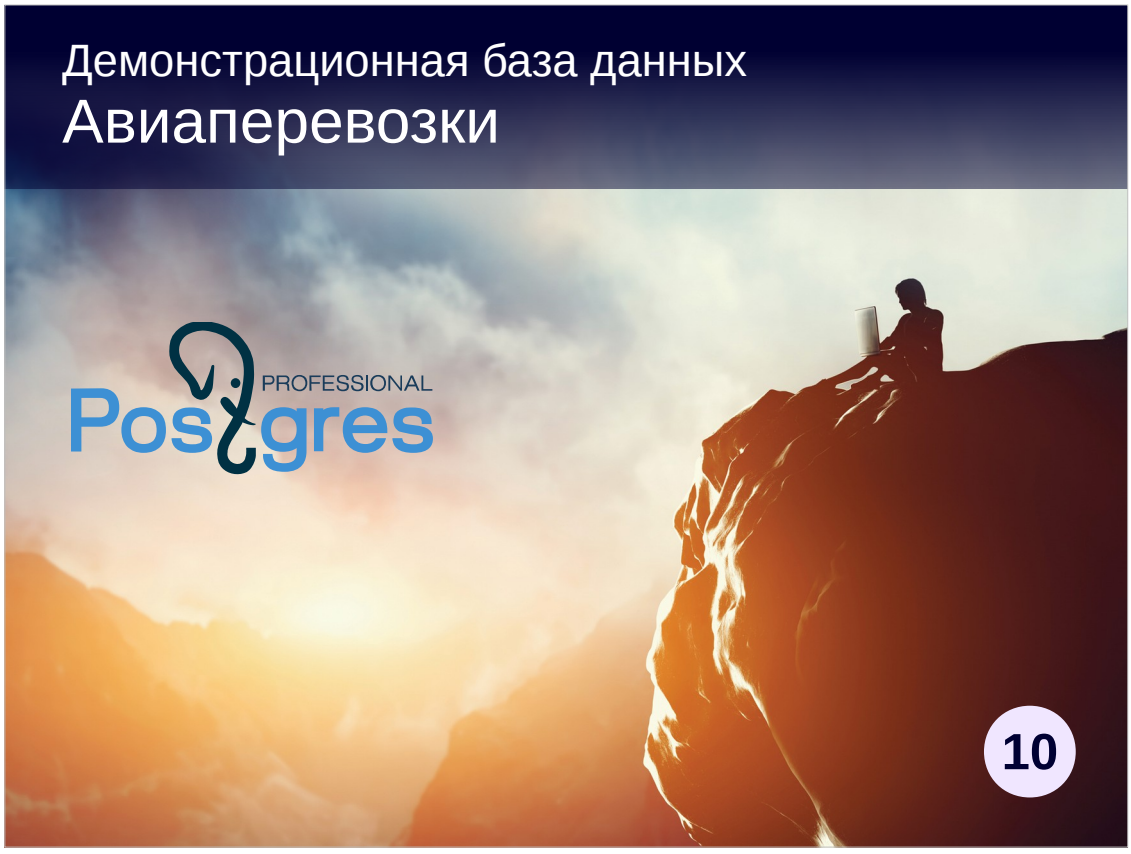


Авторские права
© Postgres Professional, 2019 год.
Авторы: Егор Рогов, Павел Лузанов

Использование материалов курса
Некоммерческое использование материалов курса (презентации, демонстрации) разрешается без ограничений. Коммерческое использование возможно только с письменного разрешения компании Postgres Professional. Запрещается внесение изменений в материалы курса.

Обратная связь
Отзывы, замечания и предложения направляйте по адресу:
edu@postgrespro.ru

Отказ от ответственности
Компания Postgres Professional не несет никакой ответственности за любые повреждения и убытки, включая потерю дохода, нанесенные прямым или непрямым, специальным или случайным использованием материалов курса. Компания Postgres Professional не предоставляет каких-либо гарантий на материалы курса. Материалы курса предоставляются на основе принципа «как есть» и компания Postgres Professional не обязана предоставлять сопровождение, поддержку, обновления, расширения и изменения.

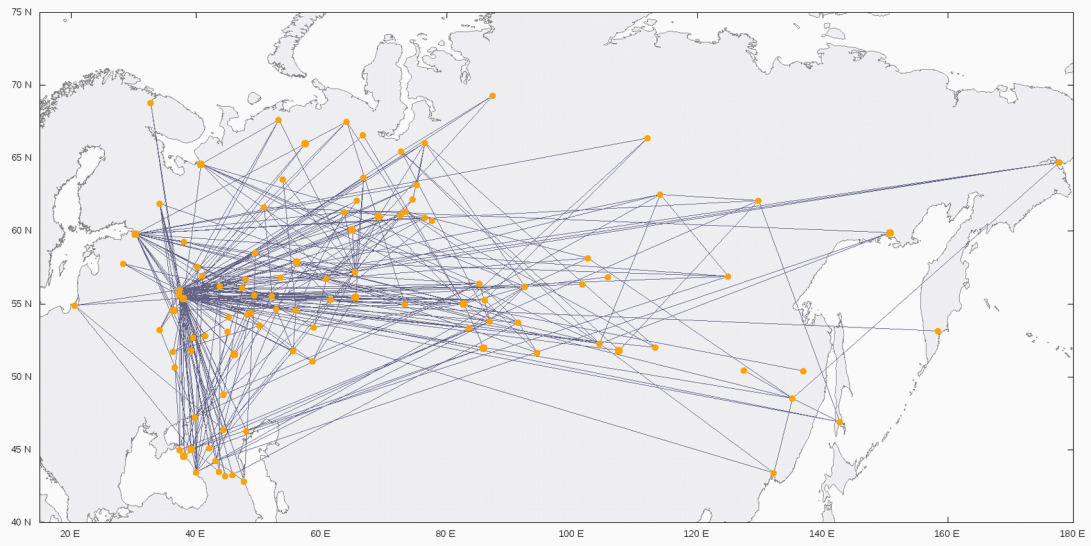
Темы

Цели и задачи

Предметная область и общая схема демобазы

Подробное описание объектов

Авиаперевозки



3

Демонстрационная база данных создавалась

- для самостоятельного изучения языка запросов SQL,
- для подготовки книг, пособий и учебных курсов по языку SQL,
- для демонстрации возможностей PostgreSQL в статьях и заметках.

При разработке демонстрационной базы данных мы преследовали несколько целей:

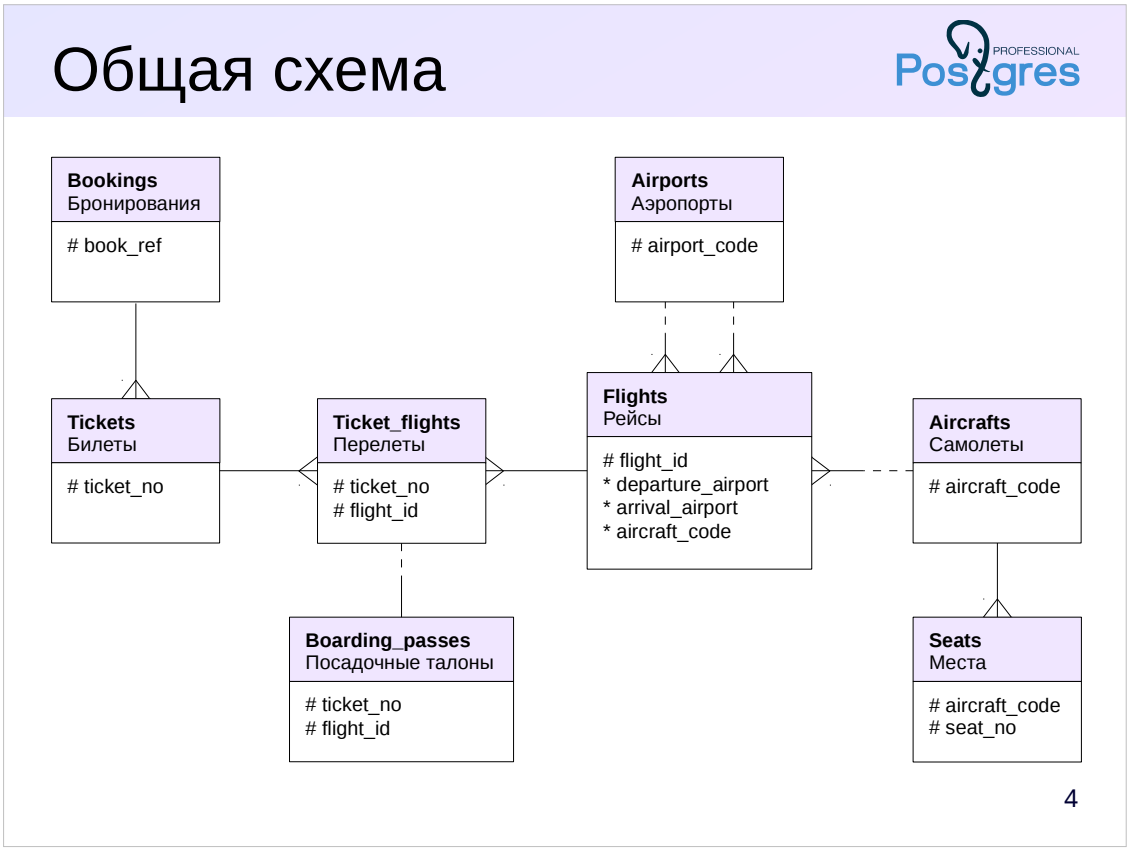
- схема данных должна быть достаточно простой, чтобы быть понятной без особых пояснений;
- в то же время схема данных должна быть достаточно сложной, чтобы позволять строить осмысленные запросы;
- база данных должна быть наполнена данными, напоминающими реальные, с которыми будет интересно работать.

Демонстрационная база данных распространяется под лицензией PostgreSQL.

База данных доступна в трех вариантах, отличающихся размером. Например, в курсе по оптимизации запросов используется база большого объема, содержащая данные по полетам за один год.

В данной теме рассматривается версия демобазы от 15.08.2017.

https://postgrespro.ru/education/demodb



Основной сущностью является **бронирование** (bookings).

В одно бронирование можно включить несколько пассажиров, каждому из которых выписывается отдельный **билет** (tickets).

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».

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 уникальна.

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.

На приведенной схеме отмечены только столбцы, соответствующие первичным и внешним ключам. Далее мы рассмотрим основные объекты демонстрационной базы данных подробнее.

Бронирования

Bookings

пассажир заранее (за месяц) бронирует билет себе и, возможно, нескольким другим пассажирам

book_ref	*номер бронирования (комбинация букв и цифр)*
book_date	*дата бронирования*
total_amount	*общая стоимость включенных в бронирование билетов*

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.

```
   Столбец    |      Тип      | Модификаторы |         Описание
--------------+---------------+--------------+--------------------------
 book_ref     | char(6)       | not null     | Номер бронирования
 book_date    | timestamptz   | not null     | Дата бронирования
 total_amount | numeric(10,2) | not null     | Полная сумма бронирования
Индексы:
    PRIMARY KEY, btree (book_ref)
Ссылки извне:
    TABLE "tickets" FOREIGN KEY (book_ref) REFERENCES bookings(book_ref)
```

Билеты

Tickets

билет выдается на одного пассажира и может включать несколько перелетов

ни идентификатор пассажира, ни имя не являются постоянными; нельзя однозначно найти все билеты одного и того же пассажира

ticket_no	*номер билета*
book_ref	*номер бронирования*
passenger_id	*идентификатор пассажира (номер документа)*
passenger_name	*имя пассажира*
contact_data	*контактные данные пассажира*

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).

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.

```
     Столбец    |     Тип     | Модификаторы |          Описание
----------------+-------------+--------------+----------------------------
 ticket_no      | char(13)    | not null     | Номер билета
 book_ref       | char(6)     | not null     | Номер бронирования
 passenger_id   | varchar(20) | not null     | Идентификатор пассажира
 passenger_name | text        | not null     | Имя пассажира
 contact_data   | jsonb       |              | Контактные данные пассажира
Индексы:
    PRIMARY KEY, btree (ticket_no)
Ограничения внешнего ключа:
    FOREIGN KEY (book_ref) REFERENCES bookings(book_ref)
Ссылки извне:
    TABLE "ticket_flights" FOREIGN KEY (ticket_no) REFERENCES
tickets(ticket_no)
```

Рейсы

Flights

рейс выполняется по расписанию из одного аэропорта в другой

естественный ключ — номер рейса и дата отправления,
но используется суррогатный ключ

flight_id	*идентификатор рейса*
flight_no	*номер рейса*
scheduled_departure/arrival	*вылет и прилет по расписанию*
actual_departure/arrival	*фактический вылет и прилет*
departure/arrival_airport	*аэропорты отправления и прибытия*
status	*статус рейса*
aircraft_code	*код самолета*

Рейс соединяет аэропорты вылета и прибытия. Такое понятие, как «рейс с пересадками» отсутствует: если нет прямого рейса, в билет просто включаются несколько рейсов.

```
       Столбец       |     Тип     | Модификаторы |          Описание
---------------------+-------------+--------------+----------------------------
 flight_id           | serial      | not null     | Идентификатор рейса
 flight_no           | char(6)     | not null     | Номер рейса
 scheduled_departure | timestamptz | not null     | Время вылета по расписанию
 scheduled_arrival   | timestamptz | not null     | Время прилёта по расписанию
 departure_airport   | char(3)     | not null     | Аэропорт отправления
 arrival_airport     | char(3)     | not null     | Аэропорт прибытия
 status              | varchar(20) | not null     | Статус рейса
 aircraft_code       | char(3)     | not null     | Код самолета, IATA
 actual_departure    | timestamptz |              | Фактическое время вылета
 actual_arrival      | timestamptz |              | Фактическое время прилёта
Индексы:
    PRIMARY KEY, btree (flight_id)
    UNIQUE CONSTRAINT, btree (flight_no, scheduled_departure)
Ограничения-проверки:
    CHECK (scheduled_arrival > scheduled_departure)
    CHECK ((actual_arrival IS NULL)
       OR  ((actual_departure IS NOT NULL AND actual_arrival IS NOT NULL)
            AND (actual_arrival > actual_departure)))
    CHECK (status IN ('On Time', 'Delayed', 'Departed',
                      'Arrived', 'Scheduled', 'Cancelled'))
Ограничения внешнего ключа:
    FOREIGN KEY (aircraft_code) REFERENCES aircrafts(aircraft_code)
    FOREIGN KEY (arrival_airport) REFERENCES airports(airport_code)
    FOREIGN KEY (departure_airport) REFERENCES airports(airport_code)
```

Перелеты

Ticket_flights

перелет соединяет билеты с рейсами

ticket_no	*номер билета*
flight_id	*идентификатор рейса*
fare_conditions	*класс обслуживания*
amount	*стоимость перелета*

Перелет соединяет билет с рейсом и идентифицируется их номерами.

Для каждого перелета указываются его стоимость (amount) и класс обслуживания (fare_conditions).

```
     Столбец     |     Тип       | Модификаторы |    Описание
-----------------+---------------+--------------+---------------------
 ticket_no       | char(13)      | not null     | Номер билета
 flight_id       | integer       | not null     | Идентификатор рейса
 fare_conditions | varchar(10)   | not null     | Класс обслуживания
 amount          | numeric(10,2) | not null     | Стоимость перелета
Индексы:
    PRIMARY KEY, btree (ticket_no, flight_id)
Ограничения-проверки:
    CHECK (amount >= 0)
    CHECK (fare_conditions IN ('Economy', 'Comfort', 'Business'))
Ограничения внешнего ключа:
    FOREIGN KEY (flight_id) REFERENCES flights(flight_id)
    FOREIGN KEY (ticket_no) REFERENCES tickets(ticket_no)
Ссылки извне:
    TABLE "boarding_passes" FOREIGN KEY (ticket_no, flight_id)
        REFERENCES ticket_flights(ticket_no, flight_id)
```

Аэропорты

Airports

город не выделен в отдельную таблицу

реализация: многоязычное представление над airports_data

airport_code	*код аэропорта*
airport_name	*название аэропорта*
city	*город*
coordinates	*координаты аэропорта (долгота и широта)*
timezone	*часовой пояс*

Аэропорт идентифицируется трехбуквенным кодом (airport_code) и имеет свое имя (airport_name).

Для города не предусмотрено отдельной сущности, но введено поле с названием города (city), позволяющее найти аэропорты одного города. Это представление также включает координаты аэропорта (coordinates) и часовой пояс (timezone).

Значения полей airport_name и city определяются в зависимости от выбранного в конфигурационном параметре *bookings.lang* языка.

```
   Столбец    |   Тип   | Модификаторы |                  Описание
--------------+---------+--------------+-------------------------------------
 airport_code | char(3) | not null     | Код аэропорта
 airport_name | text    | not null     | Название аэропорта
 city         | text    | not null     | Город
 coordinates  | point   | not null     | Координаты аэропорта
 timezone     | text    | not null     | Часовой пояс аэропорта
Определение представления:
 SELECT ml.airport_code,
    ml.airport_name ->> lang() AS airport_name,
    ml.city ->> lang() AS city,
    ml.coordinates,
    ml.timezone
   FROM airports_data ml;
```

Посадочные талоны

Boarding_passes

посадочный талон выдается при регистрации на рейс

ticket_no	*номер билета*
flight_id	*идентификатор рейса*
boarding_no	*номер посадочного талона (в порядке регистрации)*
seat_no	*номер места*

При регистрации на рейс, которая возможна за сутки до плановой даты отправления, пассажиру выдаётся 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).

```
   Столбец   |    Тип     | Модификаторы |         Описание
-------------+------------+--------------+--------------------------
 ticket_no   | char(13)   | not null     | Номер билета
 flight_id   | integer    | not null     | Идентификатор рейса
 boarding_no | integer    | not null     | Номер посадочного талона
 seat_no     | varchar(4) | not null     | Номер места
Индексы:
    PRIMARY KEY, btree (ticket_no, flight_id)
    UNIQUE CONSTRAINT, btree (flight_id, boarding_no)
    UNIQUE CONSTRAINT, btree (flight_id, seat_no)
Ограничения внешнего ключа:
    FOREIGN KEY (ticket_no, flight_id)
        REFERENCES ticket_flights(ticket_no, flight_id)
```

Самолеты

Aircrafts

модели самолетов, выполняющие рейсы

реализация: многоязычное представление над aircrafts_data

aircraft_code	*код самолета*
model	*модель самолета*
range	*максимальная дальность полета, км*

Каждая модель воздушного судна идентифицируется своим трехзначным кодом (aircraft_code). Указывается также название модели (model) и максимальная дальность полета в километрах (range).

Значение поля model определяется в зависимости от выбранного в конфигурационном параметре *bookings.lang* языка.

```
    Столбец    |   Тип   | Модификаторы |             Описание
---------------+---------+--------------+----------------------------------
 aircraft_code | char(3) | not null     | Код самолета, IATA
 model         | text    | not null     | Модель самолета
 range         | integer | not null     | Максимальная дальность полета, км
Определение представления:
 SELECT ml.aircraft_code,
    ml.model ->> lang() AS model,
    ml.range
   FROM aircrafts_data ml;
```

Места

Seats

места определяют схему салона

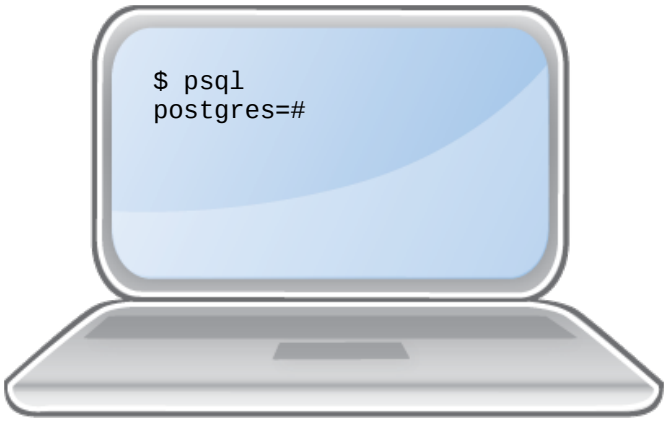
все самолеты одной модели имеют одну и ту же компоновку салона

aircraft_code	*код самолета*
seat_no	*номер места*
fare_conditions	*класс обслуживания*

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.

```
     Столбец     |     Тип     | Модификаторы |      Описание
-----------------+-------------+--------------+--------------------
 aircraft_code   | char(3)     | not null     | Код самолета, IATA
 seat_no         | varchar(4)  | not null     | Номер места
 fare_conditions | varchar(10) | not null     | Класс обслуживания
Индексы:
    PRIMARY KEY, btree (aircraft_code, seat_no)
Ограничения-проверки:
    CHECK (fare_conditions IN ('Economy', 'Comfort', 'Business'))
Ограничения внешнего ключа:
    FOREIGN KEY (aircraft_code)
        REFERENCES aircrafts(aircraft_code) ON DELETE CASCADE
```

Демонстрация



Итоги

Схема демобазы достаточно проста, но позволяет писать сложные и интересные запросы

Данные в демобазе похожи на настоящие

Демобаза может использоваться для изучения языка SQL, демонстрации возможностей PostgreSQL и т. п.

Практика

Напишите несколько запросов к демонстрационной базе данных.

1. Сколько человек бывает включено в одно бронирование?
2. До каких городов нельзя добраться без пересадок из Москвы?
3. Какая модель самолета выполняет больше всего рейсов, а какая — меньше всего?
4. А какая модель перевозит больше всего пассажиров?