

Демонстрационная база данных Авиаперевозки



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

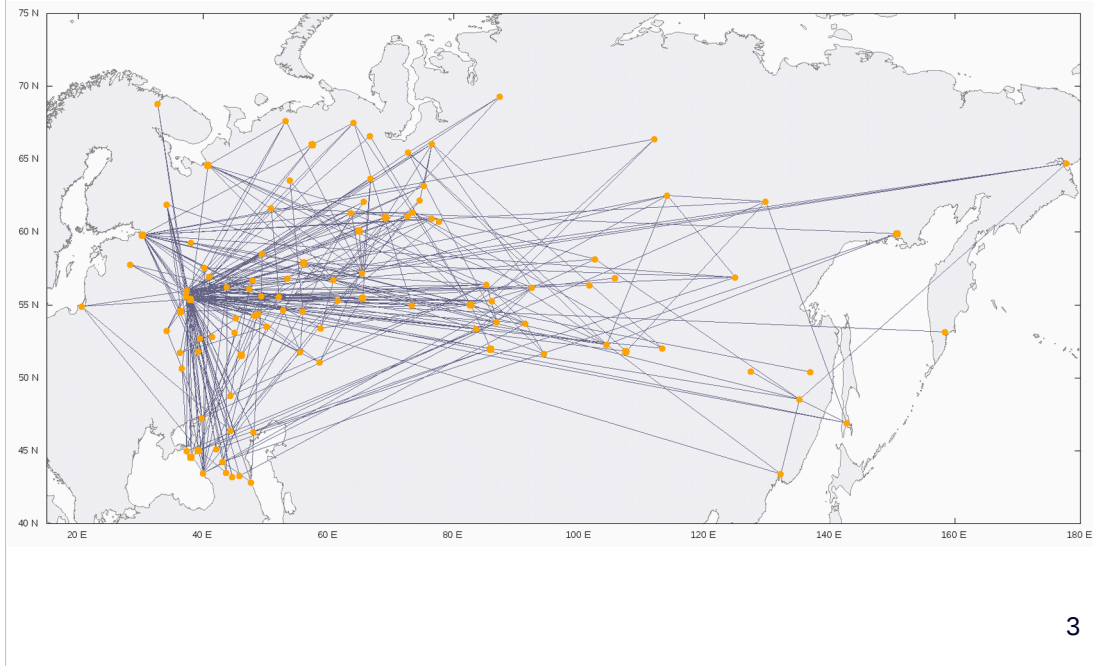
Отказ от ответственности

Компания Postgres Professional не несет никакой ответственности за любые повреждения и убытки, включая потерю дохода, нанесенные прямым или косвенным, 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.

Цели и задачи

Предметная область и общая схема демобазы

Подробное описание объектов



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.

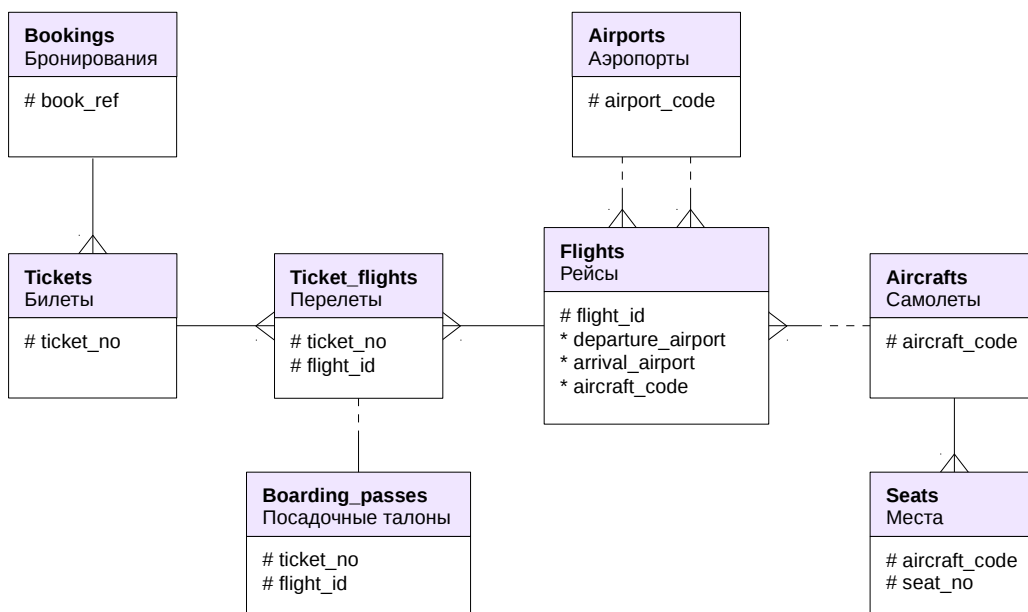
Демонстрационная база данных распространяется под лицензией PostgreSQL.

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>

Общая схема



4

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

В одно бронирование можно включить нескольких 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.

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

5

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	timestampz	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)
```

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	<i>номер билета</i>
book_ref	<i>номер бронирования</i>
passenger_id	<i>идентификатор пассажира (номер документа)</i>
passenger_name	<i>имя пассажира</i>
contact_data	<i>контактные данные пассажира</i>

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)
```

Flights

рейс выполняется по расписанию из одного аэропорта в другой
 естественный ключ — номер рейса и дата отправления,
 но используется суррогатный ключ

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

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
<code>flight_id</code>	<code>serial</code>	<code>not null</code>	Идентификатор рейса
<code>flight_no</code>	<code>char(6)</code>	<code>not null</code>	Номер рейса
<code>scheduled_departure</code>	<code>timestampz</code>	<code>not null</code>	Время вылета по расписанию
<code>scheduled_arrival</code>	<code>timestampz</code>	<code>not null</code>	Время прилёта по расписанию
<code>departure_airport</code>	<code>char(3)</code>	<code>not null</code>	Аэропорт отправления
<code>arrival_airport</code>	<code>char(3)</code>	<code>not null</code>	Аэропорт прибытия
<code>status</code>	<code>varchar(20)</code>	<code>not null</code>	Статус рейса
<code>aircraft_code</code>	<code>char(3)</code>	<code>not null</code>	Код самолета, IATA
<code>actual_departure</code>	<code>timestampz</code>		Фактическое время вылета
<code>actual_arrival</code>	<code>timestampz</code>		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)
```

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)
```

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 m1.airport_code,  
       m1.airport_name ->> lang() AS airport_name,  
       m1.city ->> lang() AS city,  
       m1.coordinates,  
       m1.timezone  
FROM airports_data m1;
```

Boarding_passes

посадочный талон выдается при регистрации на рейс

<code>ticket_no</code>	<i>номер билета</i>
<code>flight_id</code>	<i>идентификатор рейса</i>
<code>boarding_no</code>	<i>номер посадочного талона (в порядке регистрации)</i>
<code>seat_no</code>	<i>номер места</i>

10

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
<code>ticket_no</code>	<code>char(13)</code>	<code>not null</code>	Номер билета
<code>flight_id</code>	<code>integer</code>	<code>not null</code>	Идентификатор рейса
<code>boarding_no</code>	<code>integer</code>	<code>not null</code>	Номер посадочного талона
<code>seat_no</code>	<code>varchar(4)</code>	<code>not null</code>	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)
```

Aircrafts

модели самолетов, выполняющие рейсы

реализация: многоязычное представление над `aircrafts_data`

<code>aircraft_code</code>	<i>код самолета</i>
<code>model</code>	<i>модель самолета</i>
<code>range</code>	<i>максимальная дальность полета, км</i>

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
<code>aircraft_code</code>	<code>char(3)</code>	<code>not null</code>	Код самолета, IATA
<code>model</code>	<code>text</code>	<code>not null</code>	Модель самолета
<code>range</code>	<code>integer</code>	<code>not null</code>	Максимальная дальность полета, км

Определение представления:

```
SELECT m1.aircraft_code,  
       m1.model ->> lang() AS model,  
       m1.range  
FROM aircrafts_data m1;
```

Seats

места определяют схему салона

все самолеты одной модели имеют одну и ту же компоновку салона

aircraft_code	<i>код самолета</i>
seat_no	<i>номер места</i>
fare_conditions	<i>класс обслуживания</i>

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
```



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

Данные в демобазе похожи на настоящие

Демобазы могут использоваться для изучения языка SQL, демонстрации возможностей PostgreSQL и т. п.

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?