

[Описание базы данных demo](#)

[Описание схемы](#)

[Схема таблиц и представлений в схеме bookings базы demo](#)

[Заметка к модели данных:](#)

[Таблица boarding_passes](#)

[Таблица ticket_flights](#)

[Таблица flights](#)

[Таблица seats](#)

[Таблица tickets](#)

[Таблица airports_data](#)

[Таблица aircrafts_data](#)

[Таблица bookings](#)

[Представление aircrafts](#)

[Представление airports](#)

[Представление routes](#)

[Представление flights_v](#)

[Функция bookings.now](#)

[Функция bookings.lang](#)

Описание базы данных demo

База данных demo представляет собой данные о авиаперевозках по России за один месяц. Готовая база, уже наполненная данными и связями, была взята с сайта компании Postgres Professional (<https://edu.postgrespro.ru/demo-small.zip>). Данная база распространяется под [лицензией PostgreSQL](#).

Описание схемы

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.

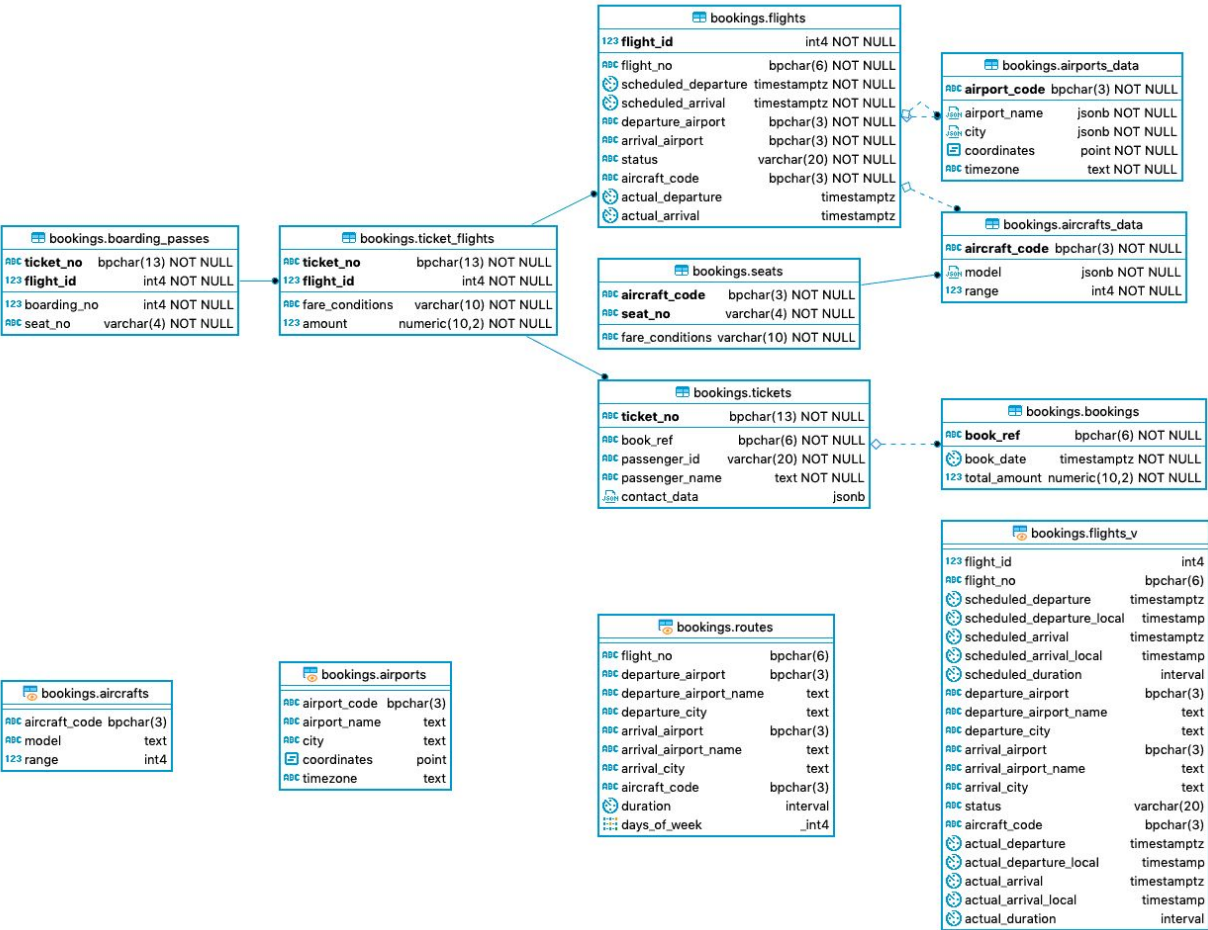
Билет включает один или несколько перелетов (ticket_flights). Несколько перелетов могут включаться в билет в случаях, когда нет прямого рейса, соединяющего пункты отправления и назначения (полет с пересадками), либо когда билет взят «туда и обратно». В схеме данных нет жёсткого ограничения, но предполагается, 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 выдаётся 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).

Схема таблиц и представлений в схеме bookings базы demo



Заметка к модели данных:

Таблица boarding_passes

При регистрации на рейс, которая возможна за сутки до плановой даты отправления, пассажиру выдаётся посадочный талон. Он идентифицируется также, как и перелёт — 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).

Поле ticket_no - номер билета (Ticket number). PRIMARY KEY and FOREIGN KEY

Поле flight_id - ID рейса (Flight ID). PRIMARY KEY and FOREIGN KEY

Поле boarding_no - номер посадочного (Boarding pass number)

Поле seat_no - номер места (Seat number)

Indexes:

"boarding_passes_pkey" PRIMARY KEY, btree (ticket_no, flight_id)

"boarding_passes_flight_id_boarding_no_key" UNIQUE CONSTRAINT, btree (flight_id, boarding_no)

"boarding_passes_flight_id_seat_no_key" UNIQUE CONSTRAINT, btree (flight_id, seat_no)

Foreign-key constraints:

"boarding_passes_ticket_no_fkey" FOREIGN KEY (ticket_no, flight_id) REFERENCES ticket_flights(ticket_no, flight_id)

Таблица ticket_flights

Перелёт соединяет билет с рейсом и идентифицируется их номерами.

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

Поле ticket_no - номер билета (Ticket number). PRIMARY KEY and FOREIGN KEY

Поле flight_id - ID рейса (Flight ID). PRIMARY KEY and FOREIGN KEY

Поле fare_conditions - класс (Travel class)

Поле amount - стоимость (Travel cost)

Indexes:

```
"ticket_flights_pkey" PRIMARY KEY, btree (ticket_no, flight_id)
```

Check constraints:

```
"ticket_flights_amount_check" CHECK (amount >= 0::numeric)
```

```
"ticket_flights_fare_conditions_check" CHECK
```

```
(fare_conditions::text = ANY (ARRAY['Economy'::character  
varying::text, 'Comfort'::character varying::text,  
'Business'::character varying::text]))
```

Foreign-key constraints:

```
"ticket_flights_flight_id_fkey" FOREIGN KEY (flight_id)
```

```
REFERENCES flights(flight_id)
```

```
"ticket_flights_ticket_no_fkey" FOREIGN KEY (ticket_no)
```

```
REFERENCES tickets(ticket_no)
```

Referenced by:

```
TABLE "boarding_passes" CONSTRAINT
```

```
"boarding_passes_ticket_no_fkey" FOREIGN KEY (ticket_no, flight_id)
```

```
REFERENCES ticket_flights(ticket_no, flight_id)
```

Таблица flights

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.

Статус рейса (`status`) может принимать одно из следующих значений:

Scheduled

Рейс доступен для бронирования. Это происходит за месяц до плановой даты вылета; до этого запись о рейсе не существует в базе данных.

On Time

Рейс доступен для регистрации (за сутки до плановой даты вылета) и не задержан.

Delayed

Рейс доступен для регистрации (за сутки до плановой даты вылета), но задержан.

Departed

Самолет уже вылетел и находится в воздухе.

Arrived

Самолет прибыл в пункт назначения.

Cancelled

Рейс отменён.

Поле `flight_id` - ID рейса (Flight ID). PRIMARY KEY

Поле `flight_no` - номер рейса (Flight number)

Поле `scheduled_departure` - запланированное время отправления (Scheduled departure time). PRIMARY KEY

Поле `scheduled_arrival` - запланированное время прибытия (Scheduled arrival time)

Поле `departure_airport` - код аэропорта отправления (Airport of departure). FOREIGN KEY

Поле `arrival_airport` - код аэропорта прибытия (Airport of arrival). FOREIGN KEY

Поле `status` - статус рейса (Flight status)

Поле `aircraft_code` - код IATA воздушного судна (Aircraft code, IATA). FOREIGN KEY

Поле `actual_departure` - фактическое время отправления (Actual departure time)

Поле `actual_arrival` - фактическое время прибытия (Actual arrival time)

Indexes:

```
"flights_pkey" PRIMARY KEY, btree (flight_id)
```

```
"flights_flight_no_scheduled_departure_key" UNIQUE CONSTRAINT,  
btree (flight_no, scheduled_departure)
```

Check constraints:

```
"flights_check" CHECK (scheduled_arrival > scheduled_departure)
```

```
"flights_check1" CHECK (actual_arrival IS NULL OR  
actual_departure IS NOT NULL AND actual_arrival IS NOT NULL AND  
actual_arrival > actual_departure)
```

```
"flights_status_check" CHECK (status::text = ANY (ARRAY['On  
Time'::character varying::text, 'Delayed'::character varying::text,  
'Departed'::character varying::text, 'Arrived'::character  
varying::text, 'Scheduled'::character varying::text,  
'Cancelled'::character varying::text]))
```

Foreign-key constraints:

```

    "flights_aircraft_code_fkey" FOREIGN KEY (aircraft_code)
REFERENCES aircrafts_data(aircraft_code)
    "flights_arrival_airport_fkey" FOREIGN KEY (arrival_airport)
REFERENCES airports_data(airport_code)
    "flights_departure_airport_fkey" FOREIGN KEY (departure_airport)
REFERENCES airports_data(airport_code)
Referenced by:
    TABLE "ticket_flights" CONSTRAINT "ticket_flights_flight_id_fkey"
FOREIGN KEY (flight_id) REFERENCES flights(flight_id)

```

Таблица seats

Места определяют схему салона каждой модели. Каждое место определяется своим номером (seat_no) и имеет закреплённый за ним класс обслуживания (fare_conditions) — Economy, Comfort или Business.

Поле aircraft_code - код IATA воздушного судна (Aircraft code, IATA). PRIMARY KEY and FOREIGN KEY

Поле seat_no - номер места (Seat number). PRIMARY KEY

Поле fare_conditions - класс (Travel class)

Indexes:

```

"seats_pkey" PRIMARY KEY, btree (aircraft_code, seat_no)

```

Check constraints:

```

"seats_fare_conditions_check" CHECK (fare_conditions::text = ANY
(ARRAY['Economy'::character varying::text, 'Comfort'::character
varying::text, 'Business'::character varying::text]))

```

Foreign-key constraints:

```

"seats_aircraft_code_fkey" FOREIGN KEY (aircraft_code) REFERENCES
aircrafts_data(aircraft_code) ON DELETE CASCADE

```

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).

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.

Поле `ticket_no` - номер билета (Ticket number). PRIMARY KEY

Поле `book_ref` - код бронирования (Booking number). FOREIGN KEY

Поле `passenger_id` - ID пассажира (Passenger ID)

Поле `passenger_name` - имя пассажира (Passenger name)

Поле `contact_data` - контактная информация пассажира (Passenger contact information)

Indexes:

`"tickets_pkey" PRIMARY KEY, btree (ticket_no)`

Foreign-key constraints:

`"tickets_book_ref_fkey" FOREIGN KEY (book_ref) REFERENCES bookings(book_ref)`

Referenced by:

`TABLE "ticket_flights" CONSTRAINT "ticket_flights_ticket_no_fkey" FOREIGN KEY (ticket_no) REFERENCES tickets(ticket_no)`

Таблица `airports_data`

Это базовая таблица для представления `airports`. Она содержит переводы значений `airport_name` и `city` на разные языки, в формате JSONB. В большинстве случаев к этой таблице не следует обращаться напрямую.

Поле `airport_code` - код аэропорта (Airport code). PRIMARY KEY

Поле `airport_name` - название аэропорта (Airport name)

Поле `city` - город (City)

Поле `coordinates` - координаты аэропорта (широта и долгота). Airport coordinates (longitude and latitude)

Поле `timezone` - тайм зона аэропорта (Airport time zone)

Indexes:

`"airports_data_pkey" PRIMARY KEY, btree (airport_code)`

Referenced by:

`TABLE "flights" CONSTRAINT "flights_arrival_airport_fkey" FOREIGN KEY (arrival_airport) REFERENCES airports_data(airport_code)`

`TABLE "flights" CONSTRAINT "flights_departure_airport_fkey" FOREIGN KEY (departure_airport) REFERENCES airports_data(airport_code)`


```
airports_data(airport_code)
```

Таблица aircrafts_data

Это базовая таблица для представления aircrafts. Поле model этой таблицы содержит переводы моделей самолётов на разные языки, в формате JSONB. В большинстве случаев к этой таблице не следует обращаться напрямую.

Поле aircraft_code - код IATA воздушного судна (Aircraft code, IATA). PRIMARY KEY

Поле model - модель воздушного судна (Aircraft model)

Поле range - максимальная дальность полета воздушного судна в километрах (Maximal flying distance, km)

Indexes:

```
"aircrafts_pkey" PRIMARY KEY, btree (aircraft_code)
```

Check constraints:

```
"aircrafts_range_check" CHECK (range > 0)
```

Referenced by:

```
TABLE "flights" CONSTRAINT "flights_aircraft_code_fkey" FOREIGN KEY (aircraft_code) REFERENCES aircrafts_data(aircraft_code)
```

```
TABLE "seats" CONSTRAINT "seats_aircraft_code_fkey" FOREIGN KEY (aircraft_code) REFERENCES aircrafts_data(aircraft_code) ON DELETE CASCADE
```

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 включённых в бронирование перелетов всех пассажиров.

Поле book_ref - код бронирования (Booking number). PRIMARY KEY

Поле book_date - дата бронирования (Booking date)

Поле total_amount - общая стоимость бронирования (Total booking cost)

Indexes:

```
"bookings_pkey" PRIMARY KEY, btree (book_ref)
Referenced by:
  TABLE "tickets" CONSTRAINT "tickets_book_ref_fkey" FOREIGN KEY
  (book_ref) REFERENCES bookings(book_ref)
```

Представление aircrafts

Поле aircraft_code - код IATA воздушного судна (Aircraft code, IATA)

Поле model - модель воздушного судна (Aircraft model)

Поле range - максимальная дальность полета воздушного судна в километрах (Maximal flying distance, km)

Значение поля model определяется в зависимости от выбранного языка.

View definition:

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

Представление airports

Поле airport_code - код аэропорта (Airport code)

Поле airport_name - название аэропорта (Airport name)

Поле city - город (City)

Поле coordinates - координаты аэропорта (широта и долгота). Airport coordinates (longitude and latitude)

Поле timezone - тайм зона аэропорта (Airport time zone)

Значения полей airport_name и city определяются в зависимости от выбранного языка.

View definition:

```
SELECT ml.airport_code,
       ml.airport_name ->> lang() AS airport_name,
       ml.city ->> lang() AS city,
       ml.coordinates,
```

```
ml.timezone  
FROM airports_data ml;
```

Представление routes

Таблица рейсов (bookings.flights) содержит избыточность: из неё можно было бы выделить информацию о маршруте (номер рейса, аэропорты отправления и назначения), которая не зависит от конкретных дат рейсов.

Именно такая информация составляет представление routes.

Поле flight_no - номер рейса (Flight number)

Поле departure_airport - код аэропорта отправления (Code of airport of departure)

Поле departure_airport_name - имя аэропорта отправления (Name of airport of departure)

Поле departure_city - город отправления (City of departure)

Поле arrival_airport - код аэропорта прибытия (Code of airport of arrival)

Поле arrival_airport_name - имя аэропорта прибытия (Name of airport of arrival)

Поле arrival_city - город прибытия (City of arrival)

Поле aircraft_code - код IATA воздушного судна (Aircraft code, IATA)

Поле duration - запланированная продолжительность полета (Scheduled duration of flight)

Поле days_of_week - дни недели, на которые запланированы рейсы (Days of week on which flights are scheduled)

View definition:

```
WITH f3 AS (  
    SELECT f2.flight_no,  
           f2.departure_airport,  
           f2.arrival_airport,  
           f2.aircraft_code,  
           f2.duration,  
           array_agg(f2.days_of_week) AS days_of_week  
    FROM ( SELECT f1.flight_no,  
                  f1.departure_airport,  
                  f1.arrival_airport,  
                  f1.aircraft_code,  
                  f1.duration,  
                  f1.days_of_week  
          FROM ( SELECT flights.flight_no,
```

```

        flights.departure_airport,
        flights.arrival_airport,
        flights.aircraft_code,
        flights.scheduled_arrival -
flights.scheduled_departure AS duration,
        to_char(flights.scheduled_departure,
'ID'::text)::integer AS days_of_week
        FROM flights) f1
        GROUP BY f1.flight_no, f1.departure_airport,
f1.arrival_airport, f1.aircraft_code, f1.duration, f1.days_of_week
        ORDER BY f1.flight_no, f1.departure_airport,
f1.arrival_airport, f1.aircraft_code, f1.duration, f1.days_of_week)
f2
        GROUP BY f2.flight_no, f2.departure_airport,
f2.arrival_airport, f2.aircraft_code, f2.duration
    )
SELECT f3.flight_no,
    f3.departure_airport,
    dep.airport_name AS departure_airport_name,
    dep.city AS departure_city,
    f3.arrival_airport,
    arr.airport_name AS arrival_airport_name,
    arr.city AS arrival_city,
    f3.aircraft_code,
    f3.duration,
    f3.days_of_week
FROM f3,
    airports dep,
    airports arr
WHERE f3.departure_airport = dep.airport_code AND
f3.arrival_airport = arr.airport_code;

```

Представление flights_v

Над таблицей flights создано представление flights_v, содержащее 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`.

Поле `flight_id` - ID рейса (Flight ID)

Поле `flight_no` - номер рейса (Flight number)

Поле `scheduled_departure` - запланированное время отправления (Scheduled departure time)

Поле `scheduled_departure_local` - запланированное время отправления, локальное время в месте отправления (Scheduled departure time, local time at the point of departure)

Поле `scheduled_arrival` - запланированное время прибытия (Scheduled arrival time)

Поле `scheduled_arrival_local` - запланированное время прибытия, локальное время в месте прибытия (Scheduled arrival time, local time at the point of destination)

Поле `scheduled_duration` - запланированная продолжительность полета (Scheduled flight duration)

Поле `departure_airport` - код аэропорта отправления (Departure airport code)

Поле `departure_airport_name` - имя аэропорта отправления (Departure airport name)

Поле `departure_city` - город отправления (City of departure)

Поле `arrival_airport` - код аэропорта прибытия (Arrival airport code)

Поле `arrival_airport_name` - имя аэропорта прибытия (Arrival airport name)

Поле `arrival_city` - город прибытия (City of arrival)

Поле `status` - статус рейса (Flight status)

Поле `aircraft_code` - код IATA воздушного судна (Aircraft code, IATA)

Поле `actual_departure` - фактическое время отправления (Actual departure time)

Поле `actual_departure_local` - фактическое время отправления, локальное время в месте отправления (Actual departure time, local time at the point of departure)

Поле `actual_arrival` - фактическое время прибытия (Actual arrival time)

Поле `actual_arrival_local` - фактическое время прибытия, , локальное время в месте прибытия (Actual arrival time, local time at the point of destination)

Поле `actual_duration` - фактическая продолжительность полета (Actual flight duration)

View definition:

```
SELECT f.flight_id,
       f.flight_no,
       f.scheduled_departure,
       timezone(dep.timezone, f.scheduled_departure) AS
scheduled_departure_local,
       f.scheduled_arrival,
       timezone(arr.timezone, f.scheduled_arrival) AS
scheduled_arrival_local,
```

```

    f.scheduled_arrival - f.scheduled_departure AS
scheduled_duration,
    f.departure_airport,
    dep.airport_name AS departure_airport_name,
    dep.city AS departure_city,
    f.arrival_airport,
    arr.airport_name AS arrival_airport_name,
    arr.city AS arrival_city,
    f.status,
    f.aircraft_code,
    f.actual_departure,
    timezone(dep.timezone, f.actual_departure) AS
actual_departure_local,
    f.actual_arrival,
    timezone(arr.timezone, f.actual_arrival) AS actual_arrival_local,
    f.actual_arrival - f.actual_departure AS actual_duration
FROM flights f,
    airports dep,
    airports arr
WHERE f.departure_airport = dep.airport_code AND f.arrival_airport
= arr.airport_code;

```

Функция bookings.now

Демонстрационная база содержит временной «срез» » данных — так, как будто в некоторый момент была сделана резервная копия реальной системы. Например, если некоторый рейс имеет статус Departed, это означает, что в момент резервного копирования самолет вылетел и находился в воздухе.

Позиция «среза» сохранена в функции bookings.now() function. Ей можно пользоваться в запросах там, где в обычной жизни использовалась бы функция now().

Кроме того, значение этой функции определяет версию демонстрационной базы данных. Актуальная версия на текущий момент — от 15 августа 2017 г.

```

CREATE OR REPLACE FUNCTION bookings.now()
RETURNS timestamp with time zone

```

```
LANGUAGE sql
IMMUTABLE
1 AS $function$SELECT '2017-08-15 18:00:00'::TIMESTAMP AT TIME
ZONE 'Europe/Moscow';$function$
```

Функция bookings.lang

Некоторые поля в демонстрационной базе содержат текст на английском и русском языках. Переводы на другие языки отсутствуют, но их несложно добавить. Функция bookings.lang возвращает значение параметра bookings.lang, то есть язык, на котором будут выдаваться значения этих полей.

Эта функция используется в представлениях aircrafts и airports и не предназначена для непосредственного использования в запросах.

```
CREATE OR REPLACE FUNCTION bookings.lang()
  RETURNS text
  LANGUAGE plpgsql
  STABLE
1 AS $function$
2 BEGIN
3   RETURN current_setting('bookings.lang');
4 EXCEPTION
5   WHEN undefined_object THEN
6     RETURN NULL;
7 END;
8 $function$
```
