

J.4. Schema Objects

- [J.4.1. List of Relations](#)
- [J.4.2. View bookings.aircrafts](#)
- [J.4.3. Table bookings.aircrafts_data](#)
- [J.4.4. View bookings.airports](#)
- [J.4.5. Table bookings.airports_data](#)
- [J.4.6. Table bookings.boarding_passes](#)
- [J.4.7. Table bookings.bookings](#)
- [J.4.8. Table bookings.flights](#)
- [J.4.9. Table bookings.seats](#)
- [J.4.10. Table bookings.ticket_flights](#)
- [J.4.11. Table bookings.tickets](#)
- [J.4.12. View bookings.flights_v](#)
- [J.4.13. View bookings.routes](#)
- [J.4.14. Function bookings.now](#)
- [J.4.15. Function bookings.lang](#)

J.4.1. List of Relations

Name	Type	Small	Medium	Big	Description
aircrafts	view				Aircraft
aircrafts_data	table	16 kB	16 kB	16 kB	Aircraft (translations)
airports	view				Airports
airports_data	table	56 kB	56 kB	56 kB	Airports (translations)
boarding_passes	table	31 MB	102 MB	427 MB	Boarding passes
bookings	table	13 MB	30 MB	105 MB	Bookings
flights	table	3 MB	6 MB	19 MB	Flights
flights_v	view				Flights
routes	view				Routes
seats	table	88 kB	88 kB	88 kB	Seats
ticket_flights	table	64 MB	145 MB	516 MB	Flight segments
tickets	table	47 MB	107 MB	381 MB	Tickets

J.4.2. View bookings.aircrafts

Each aircraft model is identified by its three-digit code (`aircraft_code`). The view also includes the name of the aircraft model (`model`) and the maximal flying distance, in kilometers (`range`).

The value of the `model` field is selected according to the chosen language. See [Section J.4.15](#) for details.

Column	Type	Modifiers	Description
aircraft_code	char(3)	not null	Aircraft code, IATA
model	text	not null	Aircraft model
range	integer	not null	Maximal flying distance, km

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

J.4.3. Table bookings.aircrafts_data

This is the base table for the aircrafts view. The model field of this table contains translations of aircraft models to different languages, in the JSONB format. In most cases, this table is not supposed to be used directly.

Column	Type	Modifiers	Description
aircraft_code	char(3)	not null	Aircraft code, IATA
model	jsonb	not null	Aircraft model
range	integer	not null	Maximal flying distance, km

```
Indexes:
    PRIMARY KEY, btree (aircraft_code)
Check constraints:
    CHECK (range > 0)
Referenced by:
    TABLE "flights" FOREIGN KEY (aircraft_code)
        REFERENCES aircrafts_data(aircraft_code)
    TABLE "seats" FOREIGN KEY (aircraft_code)
        REFERENCES aircrafts_data(aircraft_code) ON DELETE CASCADE
```

J.4.4. View bookings.airports

An airport is identified by a three-letter code (airport_code) and has a name (airport_name).

There is no separate entity for the city, but there is a city name (city) to identify the airports of the same city. The view also includes coordinates (coordinates) and the time zone (timezone).

The values of the airport_name and city fields are selected according to the chosen language. See [Section J.4.15](#) for details.

Column	Type	Modifiers	Description
airport_code	char(3)	not null	Airport code
airport_name	text	not null	Airport name
city	text	not null	City
coordinates	point	not null	Airport coordinates (longitude and latitude)
timezone	text	not null	Airport time zone

```
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;
```

J.4.5. Table bookings.airports_data

This is the base table for the airports view. This table contains translations of airport_name and city values to different languages, in the JSONB format. In most cases, this table is not supposed to be used directly.

Column	Type	Modifiers	Description
airport_code	char(3)	not null	Airport code
airport_name	jsonb	not null	Airport name
city	jsonb	not null	City
coordinates	point	not null	Airport coordinates (longitude and latitude)
timezone	text	not null	Airport time zone

```
Indexes:
    PRIMARY KEY, btree (airport_code)
Referenced by:
    TABLE "flights" FOREIGN KEY (arrival_airport)
        REFERENCES airports_data(airport_code)
    TABLE "flights" FOREIGN KEY (departure_airport)
        REFERENCES airports_data(airport_code)
```

J.4.6. Table bookings.boarding_passes

At the time of check-in, which opens twenty-four hours before the scheduled departure, the passenger is issued a boarding pass. Like the flight segment, the boarding pass is identified by the ticket number and the flight number.

Boarding passes are assigned sequential numbers (boarding_no), in the order of check-ins for the flight (this number is unique only within the context of a particular flight). The boarding pass specifies the seat number (seat_no).

Column	Type	Modifiers	Description
ticket_no	char(13)	not null	Ticket number
flight_id	integer	not null	Flight ID
boarding_no	integer	not null	Boarding pass number
seat_no	varchar(4)	not null	Seat number

Indexes:

```
PRIMARY KEY, btree (ticket_no, flight_id)
UNIQUE CONSTRAINT, btree (flight_id, boarding_no)
UNIQUE CONSTRAINT, btree (flight_id, seat_no)
```

Foreign-key constraints:

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

J.4.7. Table bookings.bookings

Passengers book tickets for themselves, and, possibly, for several other passengers, in advance (book_date, not earlier than one month before the flight). The booking is identified by its number (book_ref, a six-position combination of letters and digits).

The total_amount field stores the total cost of all tickets included into the booking, for all passengers.

Column	Type	Modifiers	Description
book_ref	char(6)	not null	Booking number
book_date	timestamptz	not null	Booking date
total_amount	numeric(10,2)	not null	Total booking cost

Indexes:

```
PRIMARY KEY, btree (book_ref)
```

Referenced by:

```
TABLE "tickets" FOREIGN KEY (book_ref) REFERENCES bookings(book_ref)
```

J.4.8. Table bookings.flights

The natural key of the bookings.flights table consists of two fields – flight_no and scheduled_departure. To make foreign keys for this table more compact, a surrogate key is used as the primary key (flight_id).

A flight always connects two points – the airport of departure (departure_airport) and arrival (arrival_airport). There is no such entity as a “connecting flight”: if there are no non-stop flights from one airport to another, the ticket simply includes several required flight segments.

Each flight has a scheduled date and time of departure (scheduled_departure) and arrival (scheduled_arrival). The actual departure time (actual_departure) and arrival time (actual_arrival) can differ: the difference is usually not very big, but sometimes can be up to several hours if the flight is delayed.

Flight status (status) can take one of the following values:

Scheduled

The flight is available for booking. It happens one month before the planned departure date; before that time, there is no entry for this flight in the database.

On Time

The flight is open for check-in (in twenty-four hours before the scheduled departure) and is not delayed.

Delayed

The flight is open for check-in (in twenty-four hours before the scheduled departure) but is delayed.

Departed

The aircraft has already departed and is airborne.

Arrived

The aircraft has reached the point of destination.

Cancelled

The flight is canceled.

Column	Type	Modifiers	Description
flight_id	serial	not null	Flight ID
flight_no	char(6)	not null	Flight number
scheduled_departure	timestamptz	not null	Scheduled departure time
scheduled_arrival	timestamptz	not null	Scheduled arrival time
departure_airport	char(3)	not null	Airport of departure
arrival_airport	char(3)	not null	Airport of arrival
status	varchar(20)	not null	Flight status
aircraft_code	char(3)	not null	Aircraft code, IATA
actual_departure	timestamptz		Actual departure time
actual_arrival	timestamptz		Actual arrival time

Indexes:

PRIMARY KEY, btree (flight_id)

UNIQUE CONSTRAINT, btree (flight_no, scheduled_departure)

Check constraints:

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'))

Foreign-key constraints:

FOREIGN KEY (aircraft_code)

REFERENCES aircrafts(aircraft_code)

FOREIGN KEY (arrival_airport)

REFERENCES airports(airport_code)

FOREIGN KEY (departure_airport)

REFERENCES airports(airport_code)

Referenced by:

TABLE "ticket_flights" FOREIGN KEY (flight_id)

REFERENCES flights(flight_id)

J.4.9. Table bookings.seats

Seats define the cabin configuration of each aircraft model. Each seat is defined by its number (seat_no) and has an assigned travel class (fare_conditions): Economy, Comfort or Business.

Column	Type	Modifiers	Description
aircraft_code	char(3)	not null	Aircraft code, IATA
seat_no	varchar(4)	not null	Seat number
fare_conditions	varchar(10)	not null	Travel class

```
Indexes:
    PRIMARY KEY, btree (aircraft_code, seat_no)
Check constraints:
    CHECK (fare_conditions IN ('Economy', 'Comfort', 'Business'))
Foreign-key constraints:
    FOREIGN KEY (aircraft_code)
        REFERENCES aircrafts(aircraft_code) ON DELETE CASCADE
```

J.4.10. Table bookings.ticket_flights

A flight segment connects a ticket with a flight and is identified by their numbers.

Each flight has its cost (amount) and travel class (fare_conditions).

Column	Type	Modifiers	Description
ticket_no	char(13)	not null	Ticket number
flight_id	integer	not null	Flight ID
fare_conditions	varchar(10)	not null	Travel class
amount	numeric(10,2)	not null	Travel cost

```
Indexes:
    PRIMARY KEY, btree (ticket_no, flight_id)
Check constraints:
    CHECK (amount >= 0)
    CHECK (fare_conditions IN ('Economy', 'Comfort', 'Business'))
Foreign-key constraints:
    FOREIGN KEY (flight_id) REFERENCES flights(flight_id)
    FOREIGN KEY (ticket_no) REFERENCES tickets(ticket_no)
Referenced by:
    TABLE "boarding_passes" FOREIGN KEY (ticket_no, flight_id)
        REFERENCES ticket_flights(ticket_no, flight_id)
```

J.4.11. Table bookings.tickets

A ticket has a unique number (ticket_no) that consists of 13 digits.

The ticket includes a passenger ID (passenger_id) – the identity document number, – their first and last names (passenger_name), and contact information (contact_data).

Neither the passenger ID, nor the name is permanent (for example, one can change the last name or passport), so it is impossible to uniquely identify all tickets of a particular passenger.

Column	Type	Modifiers	Description
ticket_no	char(13)	not null	Ticket number
book_ref	char(6)	not null	Booking number
passenger_id	varchar(20)	not null	Passenger ID
passenger_name	text	not null	Passenger name
contact_data	jsonb		Passenger contact information

Indexes:

PRIMARY KEY, btree (ticket_no)

Foreign-key constraints:

FOREIGN KEY (book_ref) REFERENCES bookings(book_ref)

Referenced by:

TABLE "ticket_flights" FOREIGN KEY (ticket_no) REFERENCES tickets(ticket_no)

J.4.12. View bookings.flights_v

There is a flights_v view over the flights table that provides additional information:

- Details about the airport of departure – departure_airport, departure_airport_name, departure_city
- Details about the airport of arrival – arrival_airport, arrival_airport_name, arrival_city
- Local departure time – scheduled_departure_local, actual_departure_local
- Local arrival time – scheduled_arrival_local, actual_arrival_local
- Flight duration – scheduled_duration, actual_duration.

Column	Type	Description
flight_id	integer	Flight ID
flight_no	char(6)	Flight number
scheduled_departure	timestamptz	Scheduled departure time
scheduled_departure_local	timestamp	Scheduled departure time, local time at the point of departure
scheduled_arrival	timestamptz	Scheduled arrival time
scheduled_arrival_local	timestamp	Scheduled arrival time, local time at the point of destination
scheduled_duration	interval	Scheduled flight duration
departure_airport	char(3)	Departure airport code
departure_airport_name	text	Departure airport name
departure_city	text	City of departure
arrival_airport	char(3)	Arrival airport code
arrival_airport_name	text	Arrival airport name
arrival_city	text	City of arrival
status	varchar(20)	Flight status
aircraft_code	char(3)	Aircraft code, IATA
actual_departure	timestamptz	Actual departure time
actual_departure_local	timestamp	Actual departure time, local time at the point of departure
actual_arrival	timestamptz	Actual arrival time
actual_arrival_local	timestamp	Actual arrival time, local time at the point of destination
actual_duration	interval	Actual flight duration

J.4.13. View bookings . routes

The bookings . flights table contains some redundancies, which you can use to single out route information (flight number, airports of departure and destination) that does not depend on the exact flight dates.

Such information is shown in the routes view.

Column	Type	Description
flight_no	char(6)	Flight number
departure_airport	char(3)	Departure airport code
departure_airport_name	text	Departure airport name
departure_city	text	City of departure
arrival_airport	char(3)	Arrival airport code
arrival_airport_name	text	Arrival airport name
arrival_city	text	City of arrival
aircraft_code	char(3)	Aircraft code, IATA
duration	interval	Flight duration
days_of_week	integer[]	Days of the week on which flights are performed

J.4.14. Function bookings . now

The demo database contains “snapshots” of data – similar to a backup copy of a real system captured at some point in time. For example, if a flight has the Departed status, it means that the aircraft had already departed and was airborne at the time of the backup copy.

The “snapshot” time is saved in the bookings . now() function. You can use this function in demo queries for cases where you would use the now() function in a real database.

In addition, the return value of this function determines the version of the demo database. The latest version available is of August 15, 2017.

J.4.15. Function bookings . lang

Some fields in the demo database are available in English and Russian. Translations to other languages are not provided, but are easy to add. The bookings . lang returns the value of the bookings . lang parameter, that is, the language in which these fields will be displayed.

This function is used in the aircrafts and airports views and is not intended to be used directly in queries.

Products

- Postgres Pro Enterprise
- Postgres Pro Standard
- Cloud Solutions
- Postgres Extensions

Resources

- Blog
- Documentation
- Webinars
- Videos
- Presentations

Services

- 24×7×365 Technical Support
- Migration to Postgres
- High Availability Deployment
- Database Audit
- Remote DBA for PostgreSQL

Community

- Events
- Training Courses
- Intro Book
- Demo Database
- Mailing List Archives

About

- Leadership team
- Partners
- Customers
- In the News
- Press Releases
- Press Info

Contacts

Neptune House, Marina Bay, office 207,
Gibraltar, GX11 1AA
info@postgrespro.com



Get in touch!

Your First and Last Name

Company

E-mail

Message

- ☐ I confirm that I have read and accepted PostgresPro’s [Privacy Policy](#).
- ☐ I agree to get Postgres Pro discount offers and other marketing communications.

Send a message