

Viktor Lokhanko

October 15, 2021

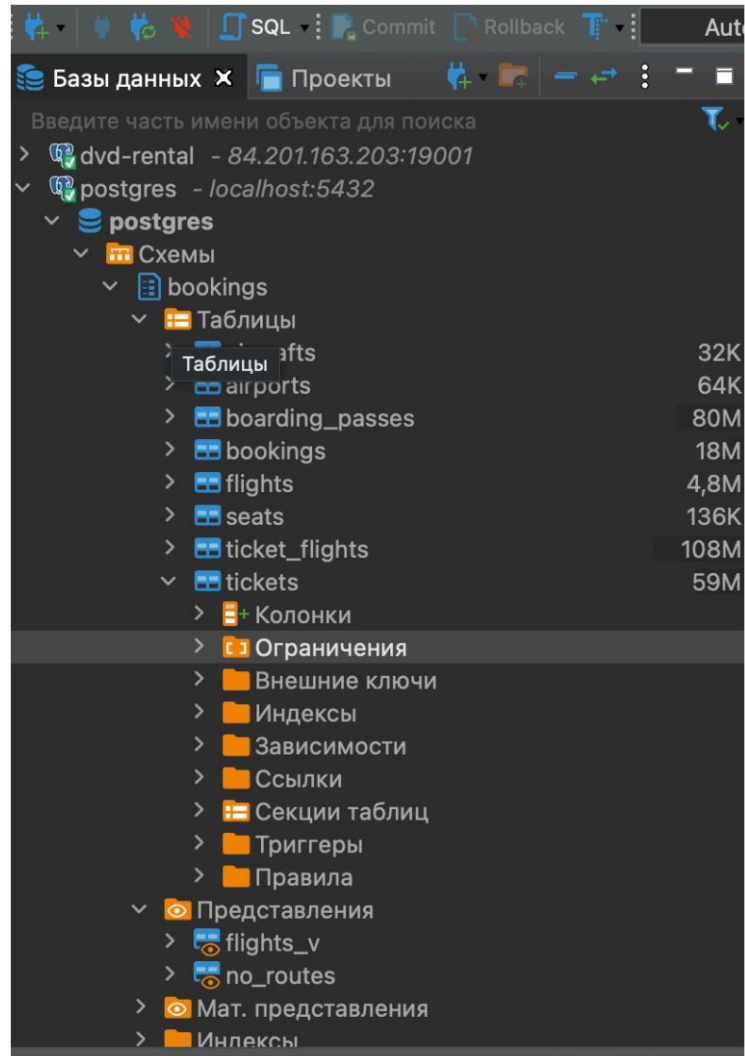
netology

# Project work on the module “SQL and Data Retrieval” **air transportation**

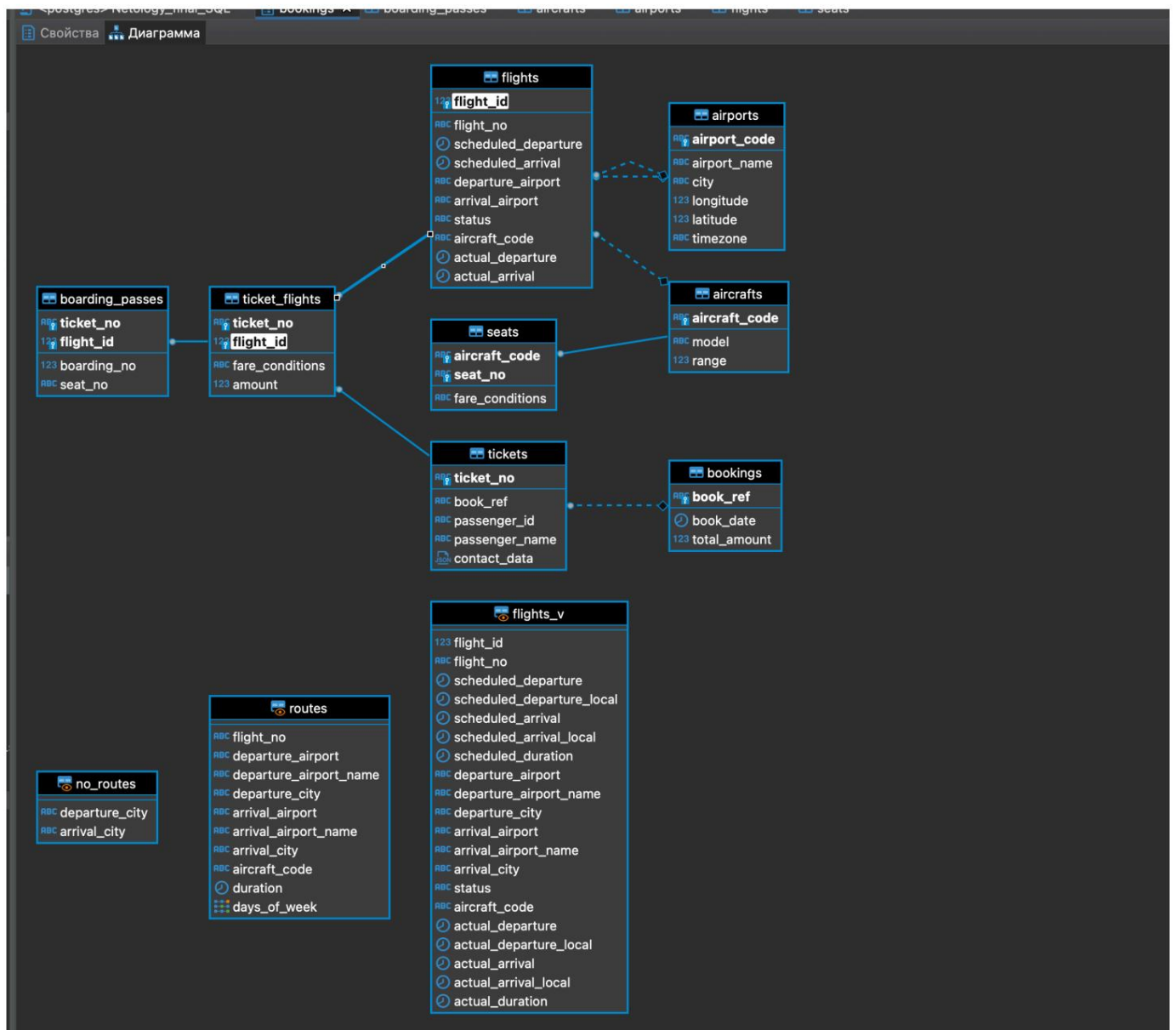


# Final work

1. The local connection type was used in the work.



## 2. Screenshot of ER diagram from DBeaver according to local connection



### 3. Brief description of the database. Tables and Views

#### tables

**aircrafts:** IATA aircraft code, aircraft model, maximum flight range, km

**airports:** airport code, airport name, city, airport coordinates: longitude,  
airport coordinates: latitude, airport coordinates: latitude

**boarding\_passes:** ticket number, flight id, boarding pass number,  
place number

**bookings:** booking number, booking date, total booking amount

**flights:** flight id, flight number, scheduled departure time, scheduled arrival time, departure airport, arrival airport,  
flight status, IATA aircraft code, actual departure time, actual arrival time

**seats:** IATA aircraft code, seat number, class of service

**ticket\_flights:** ticket number, flight id, class of service, cost  
flight

**tickets:** ticket number, booking number, passenger ID, name  
passenger, passenger contact details

#### Representation

**flights\_v view:** flight id, flight number, departure time

scheduled departure time, scheduled departure time, local time at departure point, scheduled arrival time,  
scheduled arrival time, local time at destination point, planned flight duration, airport code

departure, departure airport name, departure city, arrival airport code, arrival airport name, arrival city, flight status,  
IATA aircraft code, actual departure time, actual departure time, local time at origin, actual arrival time, actual arrival  
time, local time at point of arrival, actual flight duration

**Books.routes materialized view:** flight number, airport code

departure, departure airport name, departure city, arrival airport code, arrival airport name, arrival city, IATA aircraft  
code, flight duration, days of the week when flights are operated

## 4. Detailed analysis of the database - description of tables, logic, relationships and business areas

## tables

## aircrafts

Each aircraft model is identified by its three-digit code (*aircraft\_code*). The name of the model (*model*) and the maximum flight range in kilometers (*range*).

Indices:

PRIMARY KEY, btree  
(*aircraft\_code*)

check-limits:

CHECK (*range* > 0)

Links from outside:

TABLE "flights" FOREIGN KEY (*aircraft\_code*) REFERENCES aircrafts(*aircraft\_code*)  
TABLE "seats" FOREIGN KEY (*aircraft\_code*) REFERENCES aircrafts(*aircraft\_code*) ON  
DELETE CASCADE

bookings.aircrafts Самолеты	
ABC <b>aircraft_code - Код самолета, IATA</b>	bpchar(3) NOT NULL
ABC <b>model - Модель самолета</b>	text NOT NULL
123 <b>range - Максимальная дальность полета, км</b>	int4 NOT NULL

## airports

The airport is identified by a three-letter code (*airport\_code*) and has its own name (*airport\_name*). There is no separate entity for the city, but the name (*city*) is specified and can be used to identify the airports of one city.

The latitude (*longitude*), longitude (*latitude*) and time zone (*timezone*) are also indicated .

Indices:

PRIMARY KEY, btree  
(*airport\_code*)

Links from outside:

bookings.airports Аэропорты	
ABC <b>airport_code - Код аэропорта</b>	bpchar(3) NOT NULL
ABC <b>airport_name - Название аэропорта</b>	text NOT NULL
ABC <b>city - Город</b>	text NOT NULL
123 <b>longitude - Координаты аэропорта: долгота</b>	float8 NOT NULL
123 <b>latitude - Координаты аэропорта: широта</b>	float8 NOT NULL
ABC <b>timezone - Временная зона аэропорта</b>	text NOT NULL

TABLE "flights" FOREIGN KEY  
(*arrival\_airport*) REFERENCES airports(*airport\_code*)      TABLE "flights" FOREIGN KEY  
(*departure\_airport*) REFERENCES airports(*airport\_code*)





## boarding\_passes

When checking in for a flight, which is possible one day before the planned departure date, the passenger is issued a boarding pass. It is identified in the same way as the flight  
- ticket number and flight number. Boarding passes are assigned consecutive numbers (*boarding\_no*) in the order in which passengers check in for the flight (this number will be unique only within this flight). at the boarding the ticket indicates the seat number (*seat\_no*).

Indices:

PRIMARY KEY, btree  
(ticket\_no, flight\_id)  
UNIQUE CONSTRAINT, btree  
(flight\_id, boarding\_no)  
UNIQUE CONSTRAINT, btree  
(flight\_id, seat\_no)

bookings.ticket_flights	
Перелеты	
ABC  ticket_no - Номер билета	bpchar(13) NOT NULL
123  flight_id - Идентификатор рейса	int4 NOT NULL
ABC fare_conditions - Класс обслуживания	varchar(10) NOT NULL
123 amount - Стоимость перелета	numeric(10, 2) NOT NULL

Foreign key constraints:



FOREIGN KEY (ticket\_no, flight\_id) REFERENCES ticket\_flights(ticket\_no, flight\_id)

## bookings

The passenger in advance (*book\_date*, maximum one month before the flight) books a ticket for himself and, possibly, for several other passengers. Booking is identified number (*book\_ref*, a six-digit combination of letters and numbers). The *total\_amount* field stores the total cost of flights included in the booking for all passengers.

Indices:

PRIMARY KEY, btree  
(book\_ref)

bookings.bookings	
Бронирования	
ABC  book_ref - Номер бронирования	bpchar(6) NOT NULL
 book_date - Дата бронирования	timestampz NOT NULL
123 total_amount - Полная сумма бронирования	numeric(10, 2) NOT NULL

Links from outside:

TABLE "tickets" FOREIGN KEY (book\_ref) REFERENCES bookings(book\_ref)

## flights

The natural key of the flight table consists of two fields - flight number (*flight\_no*) and departure date (*scheduled\_departure*). To make foreign keys to this the table is more compact, a surrogate key is used as the primary (*flight\_id*). A flight always connects two points - departure airports (*departure\_airport*) and arrival (*arrival\_airport*). There is no such thing as a "flight with transfers": if there is no direct flight from one airport to another, the ticket simply includes several required flights. Each flight has a scheduled date and departure time (*scheduled\_departure*) and arrival time (*scheduled\_arrival*). real time departure (*actual\_departure*) and arrival (*actual\_arrival*) may differ: usually not strongly, but sometimes for several hours if the flight is delayed. Flight *status* (*status*) can take one of the following values:











- *Scheduled* Flight is available for booking. This happens a month before planned date of departure; prior to this, the flight record does not exist in the database.
- *On Time* The flight is available for check-in (one day before the scheduled departure date) and not detained.
- *Delayed* The flight is available for check-in (one day before the scheduled departure date), but detained.
- *Departed* The aircraft has already departed and is in the air.
- *Arrived* The plane has arrived at its destination.
- *Cancelled* Flight canceled

Indices:

PRIMARY KEY, btree (*flight\_id*)  
 UNIQUE CONSTRAINT,  
 btree (*flight\_no*,  
*scheduled\_departure*)

check-limits:

```
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'))
```

bookings.flights Рейсы	
 <b>flight_id - Идентификатор рейса</b>	serial4 NOT NULL
 flight_no - Номер рейса	bpchar(6) NOT NULL
 scheduled_departure - Время вылета по расписанию	timestampz NOT NULL
 scheduled_arrival - Время прилёта по расписанию	timestampz NOT NULL
 departure_airport - Аэропорт отправления	bpchar(3) NOT NULL
 arrival_airport - Аэропорт прибытия	bpchar(3) NOT NULL
 status - Статус рейса	varchar(20) NOT NULL
 aircraft_code - Код самолета, IATA	bpchar(3) NOT NULL
 actual_departure - Фактическое время вылета	timestampz
 actual_arrival - Фактическое время прилёта	timestampz

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)
```

Links from outside:

TABLE "ticket\_flights" FOREIGN KEY

## seats

The seats define the interior layout of each model. Each place is defined by its number (*seat\_no*) and has a class of service assigned to it (*fare\_conditions*) - Economy, Comfort or Business.

Indices:




PRIMARY KEY, btree  
(aircraft\_code, seat\_no)

check-limits:

```
CHECK(fare_conditions IN('Economy', 'Comfort', 'Business'))
```

Foreign key constraints:

```
FOREIGN KEY (aircraft_code) REFERENCES aircrafts(aircraft_code) ON DELETE CASCADE
```

bookings.seats Места	
ABC  aircraft_code - Код самолета, IATA	bpchar(3) NOT NULL
ABC  seat_no - Номер места	varchar(4) NOT NULL
ABC  fare_conditions - Класс обслуживания	varchar(10) NOT NULL

## ticket\_flights





A flight connects a ticket to a flight and is identified by their flight numbers. For everybody flight, its cost (amount) and class of service (*fare\_conditions*) are indicated.

Indices:

PRIMARY KEY, btree  
(ticket\_no, flight\_id)

check-limits:

```
CHECK (amount >= 0) CHECK (fare_conditions IN ('Economy', 'Comfort', 'Business'))
```

bookings.ticket_flights Перелеты	
ABC  ticket_no - Номер билета	bpchar(13) NOT NULL
123  flight_id - Идентификатор рейса	int4 NOT NULL
ABC  fare_conditions - Класс обслуживания	varchar(10) NOT NULL
123  amount - Стоимость перелета	numeric(10, 2) NOT NULL



Foreign key constraints:

```
FOREIGN KEY (flight_id) REFERENCES flights(flight_id)
```

```
FOREIGN KEY (ticket_no) REFERENCES tickets(ticket_no)
```

Links from outside:

```
TABLE "boarding_passes" FOREIGN KEY (ticket_no, flight_id)
REFERENCES ticket_flights(ticket_no, flight_id)
```

## tickets

The ticket has a unique number (*ticket\_no*) consisting of 13 digits. Ticket contains passenger ID (*passenger\_id*) — number of the document certifying identity, - his last name and first name (*passenger\_name*) and contact information (*contact\_data*). Neither the passenger ID nor the name is permanent (you can change your passport, you can change your last name), so you can definitely find everything tickets of the same passenger is not possible.

Indices:

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

bookings.tickets Билеты	
ABC ticket_no - Номер билета	bpchar(13) NOT NULL
ABC book_ref - Номер бронирования	bpchar(6) NOT NULL
ABC passenger_id - Идентификатор пассажира	varchar(20) NOT NULL
ABC passenger_name - Имя пассажира	text NOT NULL
JSON contact_data - Контактные данные пассажира	jsonb

Foreign key constraints:

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

Links from outside:

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

## Representation

### flights\_v view












A *flights\_v* view is created above the *flights* table, containing an additional information:

- decryption of data about the airport of departure (*departure\_airport*, *departure\_airport\_name*, *departure\_city*),
- decoding data 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*).

bookings.flights_v Рейсы		
123 flight_id - Идентификатор рейса		int4
flight_no - Номер рейса		bpchar(6)
🕒 scheduled_departure - Время вылета по расписанию		timestampz
🕒 scheduled_departure_local - Время вылета по расписанию, местное время в пункте отправления		timestamp
🕒 scheduled_arrival - Время прилёта по расписанию		timestampz
🕒 scheduled_arrival_local - Время прилёта по расписанию, местное время в пункте прибытия		timestamp
🕒 scheduled_duration - Планируемая продолжительность полета		interval
airport departure_airport - Код аэропорта отправления		bpchar(3)
airport departure_airport_name - Название аэропорта отправления		text
airport departure_city - Город отправления		text
airport arrival_airport - Код аэропорта прибытия		bpchar(3)
airport arrival_airport_name - Название аэропорта прибытия		text
airport arrival_city - Город прибытия		text
airport status - Статус рейса		varchar(20)
airport aircraft_code - Код самолета, IATA		bpchar(3)
🕒 actual_departure - Фактическое время вылета		timestampz
🕒 actual_departure_local - Фактическое время вылета, местное время в пункте отправления		timestamp
🕒 actual_arrival - Фактическое время прилёта		timestampz
🕒 actual_arrival_local - Фактическое время прилёта, местное время в пункте прибытия		timestamp
🕒 actual_duration - Фактическая продолжительность полета		interval

Materialized view **bookings.routes**

The flight table contains redundancy: from it it would be possible to extract information about the route (flight number, airports of departure and destination), which does not depend on specific flight dates. It is this information that makes up the materialized view of routes.

 <b>bookings.routes</b> Маршруты		
 flight_no - Номер рейса		bpchar(6)
 departure_airport - Код аэропорта отправления		bpchar(3)
 departure_airport_name - Название аэропорта отправления		text
 departure_city - Город отправления		text
 arrival_airport - Код аэропорта прибытия		bpchar(3)
 arrival_airport_name - Название аэропорта прибытия		text
 arrival_city - Город прибытия		text
 aircraft_code - Код самолета, IATA		bpchar(3)
 duration - Продолжительность полета		interval
 days_of_week - Дни недели, когда выполняются рейсы		_int4

What business problems can be solved using this data

1 Key business metrics for airlines:

**Capacity** - the number of seats available (ASM). The number of seats is calculated by multiplying the number of miles a given aircraft will fly by the number of seats available for that flight.

**Traffic** - Revenue Passenger Miles (RPM). Income passenger miles are calculated by multiplying the number of paying passengers by the distance travelled.

**The load factor** is a percentage that reflects how efficiently an airline generates revenue. To calculate an airline's load factor, divide revenue passenger miles (RPM) by the number of seats available (ASM).

**Revenue** - Revenue per available seat (empty or full) per flight mile (RASM). The calculation of revenue per available seat mile (RASM) is a general operating revenue divided by the number of seats available.

**Cost** - Cost Per Available Seat Mile (CASM)

**Passenger revenue per mile** of available seat and counts all seats, including empty seats (PRASM)

Based on the methodology for calculating the above metrics and the data available to us, we can only calculate capacity, traffic, and load factor. Having made additional calculations, we obtain the following data (taking into account that we have flight data for 1 month):

**Capacity** - Seats Available (ASM) - 1,145,118,307 seat miles

**Traffic** - Revenue passenger miles (RPM) - 579,254,762 passenger miles

**Load Factor** (RPM/ASM) - 0.51

Let's see what was the statistics on the coefficient of statistics on the market during this period. Publicly available data from the US Bureau of Transportation Statistics. In the period under consideration (September-November 2016), in the sector of domestic passenger air transportation, the load factor was *0.85*. (*Bureau of Transportation Statistics*). This is 1.66 times more than in the

---

airlines.

This suggests that, most likely, this airline is unprofitable in domestic passenger air transportation.

What can we advise to increase the load factor:

1. Reducing the frequency of departures on low-load routes
2. Increasing the frequency of departures on highly loaded destinations
3. Analysis of the correctness of the time of departure / arrival (morning, afternoon, evening)
4. Replacement with aircraft of smaller capacity (on highly loaded routes - with aircraft of larger capacity).
5. Increase in prices, tariffs
6. On potentially promising areas that can "shoot"
  - carrying out marketing activities

Of course, all these changes need to be carried out by carrying out a huge analytical work with the analysis of a lot of additional data.

5. List of SQL queries from application No. 2 with a description of the logic of their execution.

The query execution logic is written in the form of comments in the SQL query.