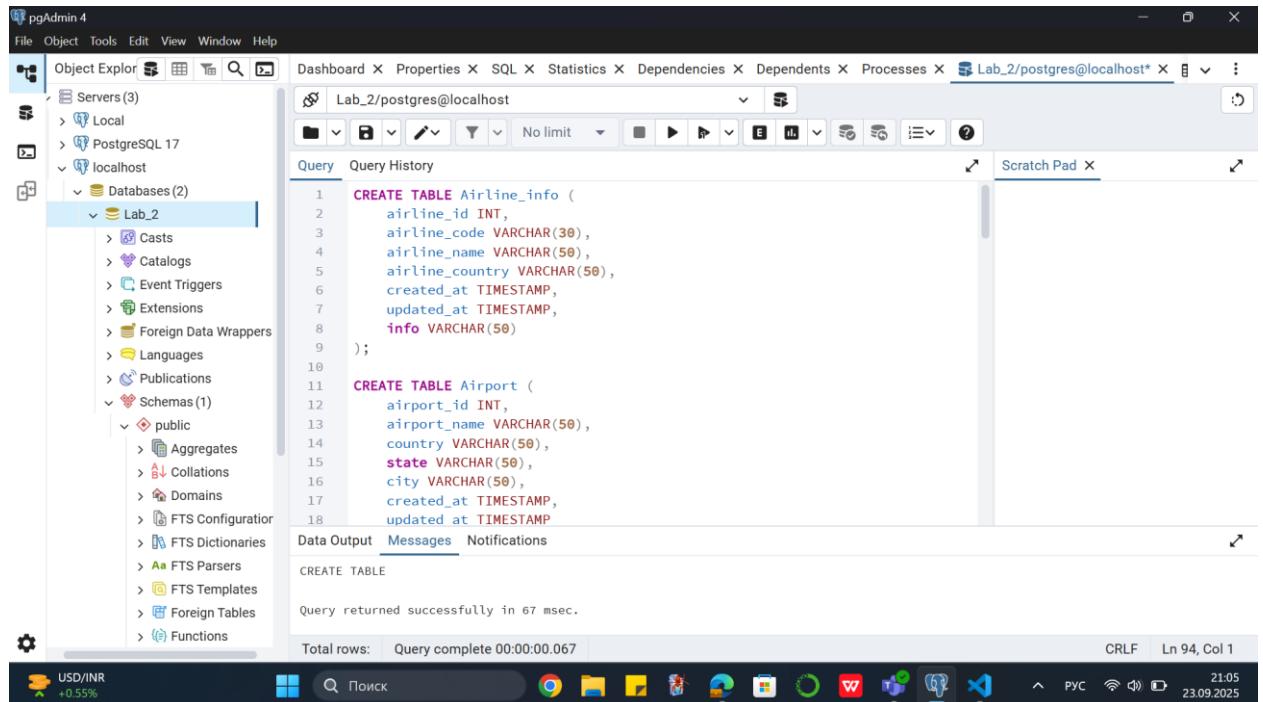


# Laboratory work #2

## DDL

1. Create following tables with corresponding attributes:

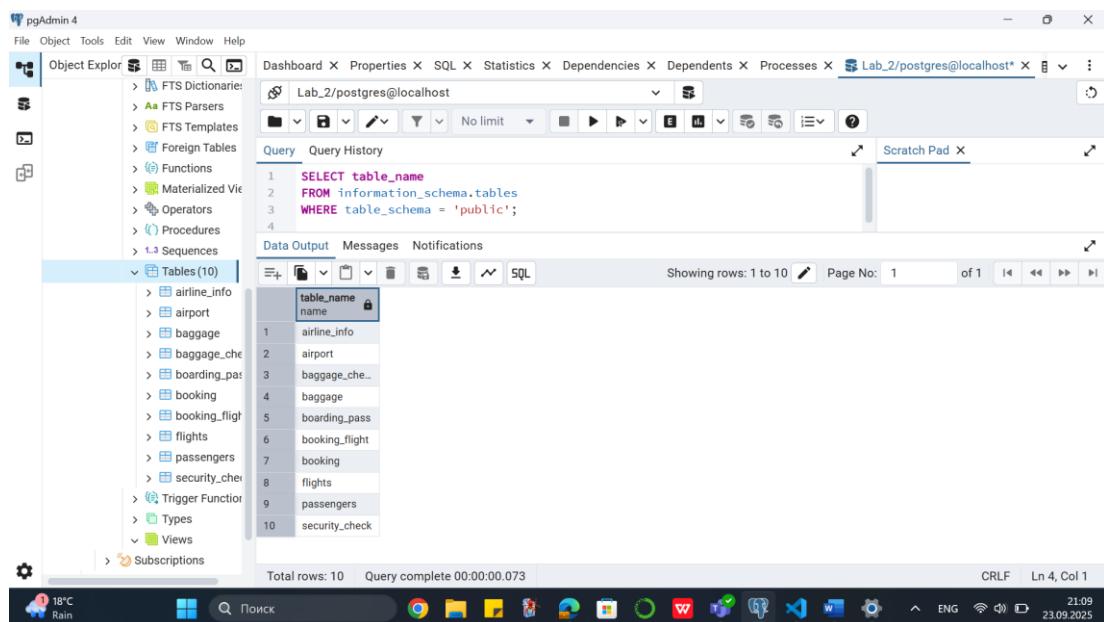
Here is the creation of tables



```
CREATE TABLE Airline_info (
    airline_id INT,
    airline_code VARCHAR(30),
    airline_name VARCHAR(50),
    airline_country VARCHAR(50),
    created_at TIMESTAMP,
    updated_at TIMESTAMP,
    info VARCHAR(50)
);

CREATE TABLE Airport (
    airport_id INT,
    airport_name VARCHAR(50),
    country VARCHAR(50),
    state VARCHAR(50),
    city VARCHAR(50),
    created_at TIMESTAMP,
    updated_at TIMESTAMP
);
```

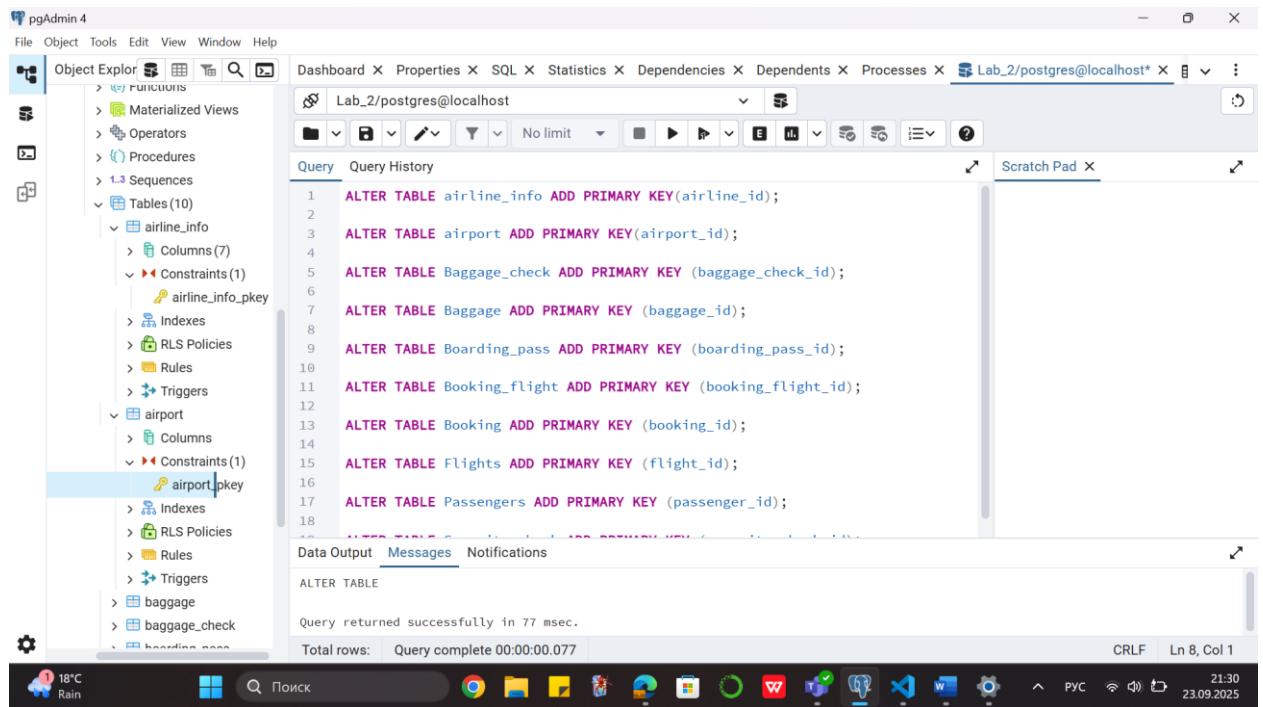
Here's what they look like



```
SELECT table_name
FROM information_schema.tables
WHERE table_schema = 'public';
```

table_name
airline_info
airport
baggage
baggage_check
boarding_pass
booking
booking_flight
flights
passengers
security_check

## 2. Define Primary Keys for each tables;

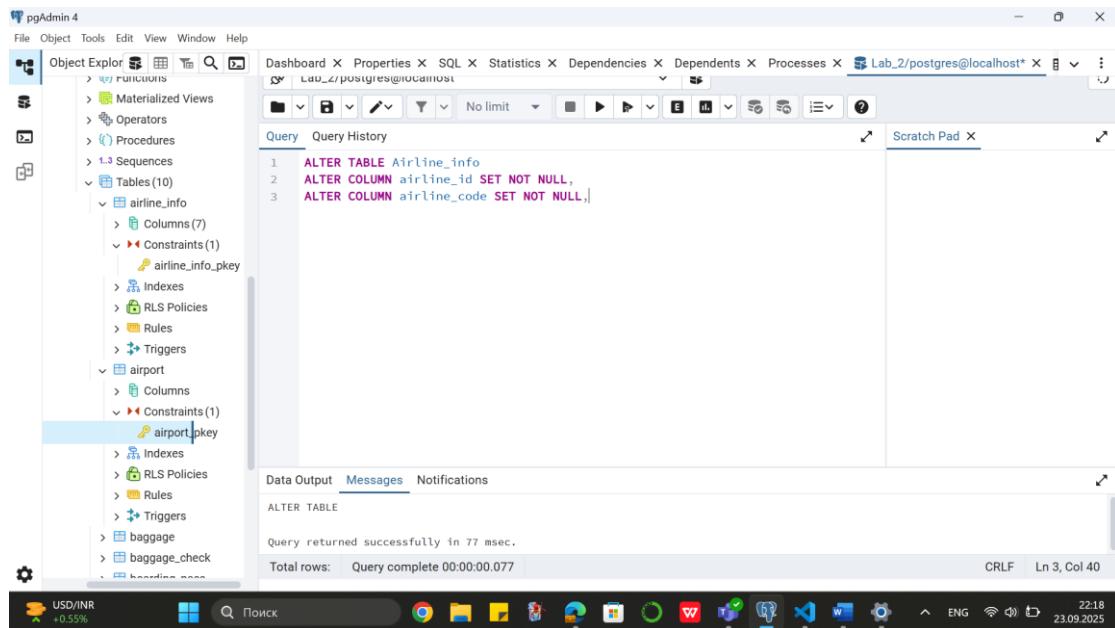


```
ALTER TABLE airline_info ADD PRIMARY KEY(airline_id);
ALTER TABLE airport ADD PRIMARY KEY(airport_id);
ALTER TABLE Baggage_check ADD PRIMARY KEY (baggage_check_id);
ALTER TABLE Baggage ADD PRIMARY KEY (baggage_id);
ALTER TABLE Boarding_pass ADD PRIMARY KEY (boarding_pass_id);
ALTER TABLE Booking_flight ADD PRIMARY KEY (booking_flight_id);
ALTER TABLE Booking ADD PRIMARY KEY (booking_id);
ALTER TABLE Flights ADD PRIMARY KEY (flight_id);
ALTER TABLE Passengers ADD PRIMARY KEY (passenger_id);
```

## 3. Define for all attributes not null constraint;

In this step I realized that it is too slow and hard, and I can do it more simply. So I decided to drop all tables, then go to the query history, copy how I created the tables, and remake all the code.

This is difficult variant:



```
ALTER TABLE Airline_info
ALTER COLUMN airline_id SET NOT NULL,
ALTER COLUMN airline_code SET NOT NULL;
```

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes Lab\_2/postgres@localhost

Query Query History

Show queries generated internally by pgAdmin?

Remove Remove All

23.09.2025 67 msec

Date Rows affected Duration

Copy Copy to Query Editor

```

CREATE TABLE Airline_info (
    airline_id INT,
    airline_code VARCHAR(30),
    airline_name VARCHAR(50),
    airline_country VARCHAR(50),
    created_at TIMESTAMP,
    updated_at TIMESTAMP,
    info VARCHAR(50)
);

CREATE TABLE Airport (
    airport_id INT,
    airport_name VARCHAR(50)
);

```

Data Output Messages Notifications

ALTER TABLE

Query returned successfully in 77 msec.

Total rows: Query complete 00:00:00.077 CRLF Ln 1, Col 1

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes Lab\_2/postgres@localhost\*

Query Query History

```

1 DROP TABLE IF EXISTS Security_check;
2 DROP TABLE IF EXISTS Baggage_check;
3 DROP TABLE IF EXISTS Baggage;
4 DROP TABLE IF EXISTS Boarding_pass;
5 DROP TABLE IF EXISTS Booking_flight;
6 DROP TABLE IF EXISTS Booking;
7 DROP TABLE IF EXISTS Flights;
8 DROP TABLE IF EXISTS Passengers;
9 DROP TABLE IF EXISTS Airport;
10 DROP TABLE IF EXISTS Airline_info;
11
12

```

Data Output Messages Notifications

ЗАМЕЧАНИЕ: таблица "security\_check" не существует, пропускается  
ЗАМЕЧАНИЕ: таблица "baggage\_check" не существует, пропускается  
ЗАМЕЧАНИЕ: таблица "baggage" не существует, пропускается

Total rows: Query complete 00:00:00.105 CRLF Ln 12, Col 1

This is how it should have been

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer Dashboard Properties SQL Statistics Dependencies Dependents Processes Lab\_2/postgres@localhost\*

Query Query History

```

CREATE TABLE Airline_info (
    airline_id INT PRIMARY KEY,
    airline_code VARCHAR(30) NOT NULL,
    airline_name VARCHAR(50) NOT NULL,
    airline_country VARCHAR(50) NOT NULL,
    created_at TIMESTAMP NOT NULL,
    updated_at TIMESTAMP NOT NULL,
    info VARCHAR(50) NOT NULL
);

CREATE TABLE Airport (
    airport_id INT PRIMARY KEY,
    airport_name VARCHAR(50) NOT NULL,
    country VARCHAR(50) NOT NULL,
    state VARCHAR(50) NOT NULL,
    city VARCHAR(50) NOT NULL,
    created_at TIMESTAMP NOT NULL,
    updated_at TIMESTAMP NOT NULL
);

```

Data Output Messages Notifications

CREATE TABLE

Query returned successfully in 60 msec.

Total rows: Query complete 00:00:00.060 CRLF Ln 4, Col 1

4. Rename airline\_info table to airline;
5. Rename column price to ticket\_price in booking table;
6. Change data type of departing\_gate from varchar(50) to text;
7. Drop the column info(varchar(50)) from the airline table.

```

ALTER TABLE Airline_info RENAME TO airline;
ALTER TABLE booking RENAME COLUMN price TO ticket_price;
ALTER TABLE Flights ALTER COLUMN departing_gate TYPE TEXT;
ALTER TABLE Airline DROP COLUMN info;

```

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, showing the database structure with 'Tables(10)' selected. The main area is the SQL tab, which contains the four ALTER TABLE commands. Below the SQL tab, the Data Output tab shows the successful execution of the queries. The bottom status bar indicates 'Query complete 00:00:00.058'.

9. Make a relationship between following tables:
  - Passengers with Secuitiry\_check, Booking, Baggage\_check by passenger\_id;
  - Booking with Baggage\_check, Baggage, Boarding\_pass, Booking\_flight by booking\_id;
  - Flights with Booking\_flight by flight\_id;
  - Airport with Flights by departing\_airport\_id;
  - Airport with Flights by arriving\_airport\_id;
  - Airline with Flights by airline\_id;

```

--Passengers
ALTER TABLE security_check
ADD CONSTRAINT fk_security_check_passenger
FOREIGN KEY (passenger_id) REFERENCES passengers(passenger_id);

ALTER TABLE booking
ADD CONSTRAINT fk_booking_passenger
FOREIGN KEY (passenger_id) REFERENCES passengers(passenger_id);

ALTER TABLE baggage_check
ADD CONSTRAINT fk_baggage_check_passenger
FOREIGN KEY (passenger_id) REFERENCES passengers(passenger_id);

--Booking
ALTER TABLE Baggage_check
ADD CONSTRAINT fk_baggage_check_booking
FOREIGN KEY (booking_id) REFERENCES booking(booking_id);

```

Total rows: Query complete 00:00:00.072

## DML

1. Generate and insert 200 random rows in your airport database.

```

INSERT INTO airport (airport_id, airport_name, country, state, city, created_at, updated_at)
SELECT
    id,
    'Airport_' || id,
    'Country_' || (id % 40 + 1),
    'State_' || (id % 20 + 1),
    'City_' || (id % 80 + 1),
    NOW() - (random() * INTERVAL '365 days'),
    NOW()
FROM generate_series(1, 200) AS id;

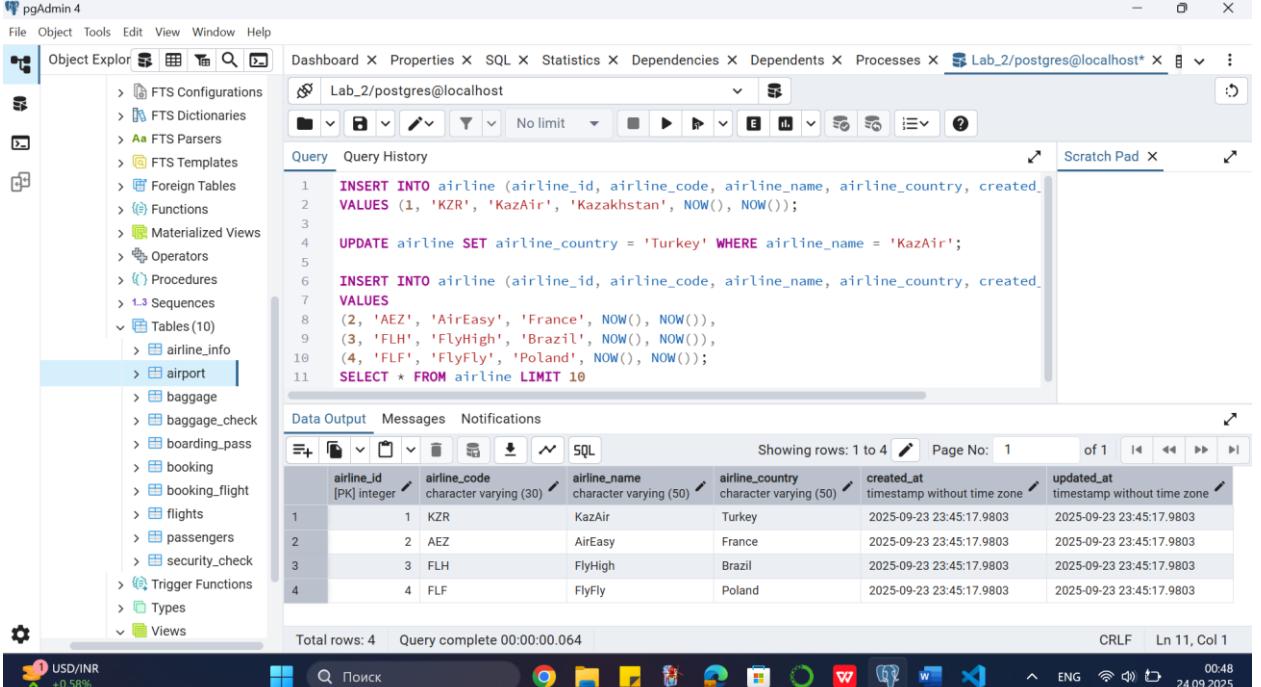
```

INSERT 0 200

Total rows: Query complete 00:00:00.064

2. Add a new airline named "KazAir" based in "Kazakhstan" to the airline table.
3. Update the airline country "KazAir" to "Turkey".

4. Add three airlines at once: "AirEasy" in "France", "FlyHigh" in "Brazil" and "FlyFly" in "Poland".



```

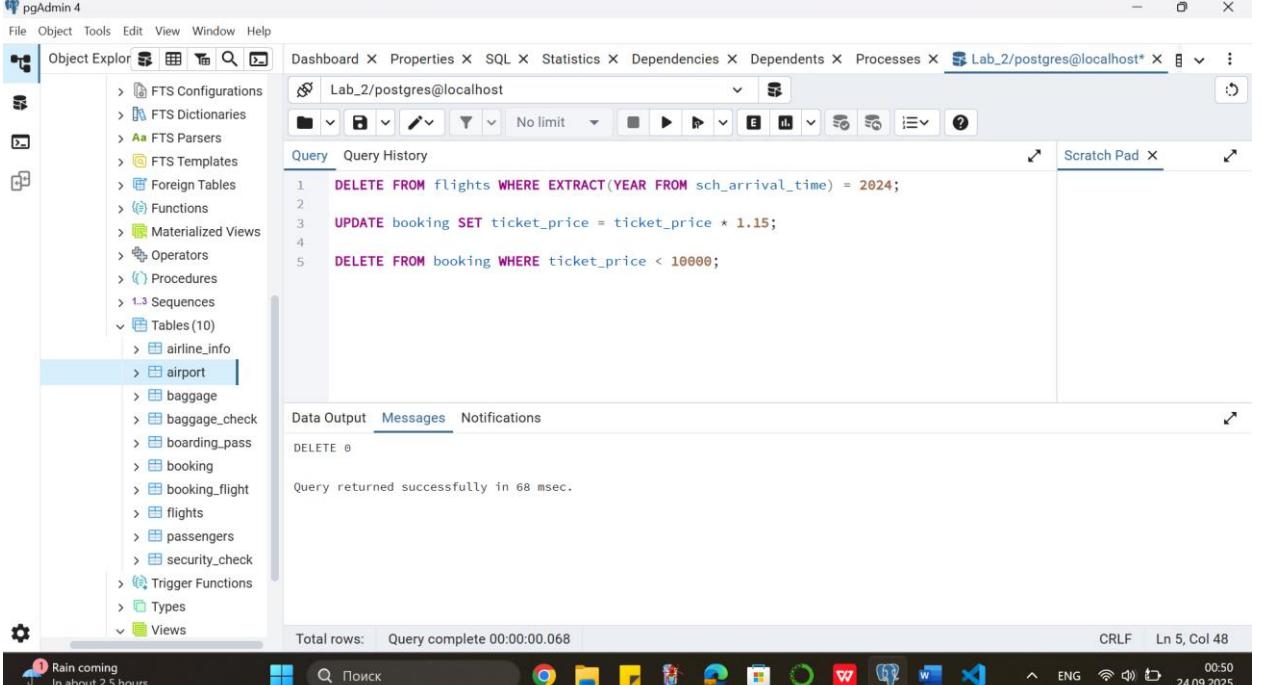
INSERT INTO airline (airline_id, airline_code, airline_name, airline_country, created_at, updated_at)
VALUES (1, 'KZR', 'KazAir', 'Kazakhstan', NOW(), NOW());
UPDATE airline SET airline_country = 'Turkey' WHERE airline_name = 'KazAir';
INSERT INTO airline (airline_id, airline_code, airline_name, airline_country, created_at, updated_at)
VALUES (2, 'AEZ', 'AirEasy', 'France', NOW(), NOW()),
(3, 'FLH', 'FlyHigh', 'Brazil', NOW(), NOW()),
(4, 'FLF', 'FlyFly', 'Poland', NOW(), NOW());
SELECT * FROM airline LIMIT 10
  
```

airline_id	airline_code	airline_name	airline_country	created_at	updated_at
1	KZR	KazAir	Turkey	2025-09-23 23:45:17.9803	2025-09-23 23:45:17.9803
2	AEZ	AirEasy	France	2025-09-23 23:45:17.9803	2025-09-23 23:45:17.9803
3	FLH	FlyHigh	Brazil	2025-09-23 23:45:17.9803	2025-09-23 23:45:17.9803
4	FLF	FlyFly	Poland	2025-09-23 23:45:17.9803	2025-09-23 23:45:17.9803

5. Delete all flights whose arrival in 2024 year.

6. Increase the price of all tickets in booking table for flights by 15%.

7. Delete all tickets where price is less than 10000.



```

DELETE FROM flights WHERE EXTRACT(YEAR FROM sch_arrival_time) = 2024;
UPDATE booking SET ticket_price = ticket_price * 1.15;
DELETE FROM booking WHERE ticket_price < 10000;
  
```

DELETE 0

Query returned successfully in 68 msec.