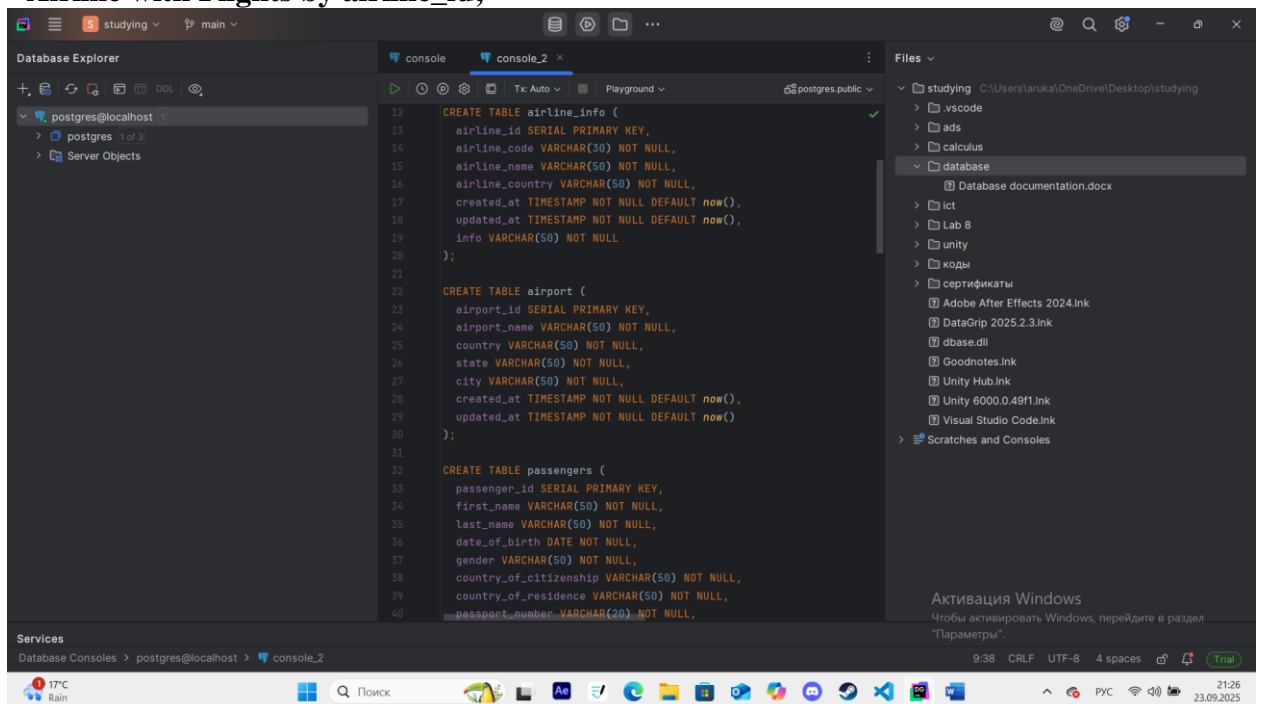


# DATABASE

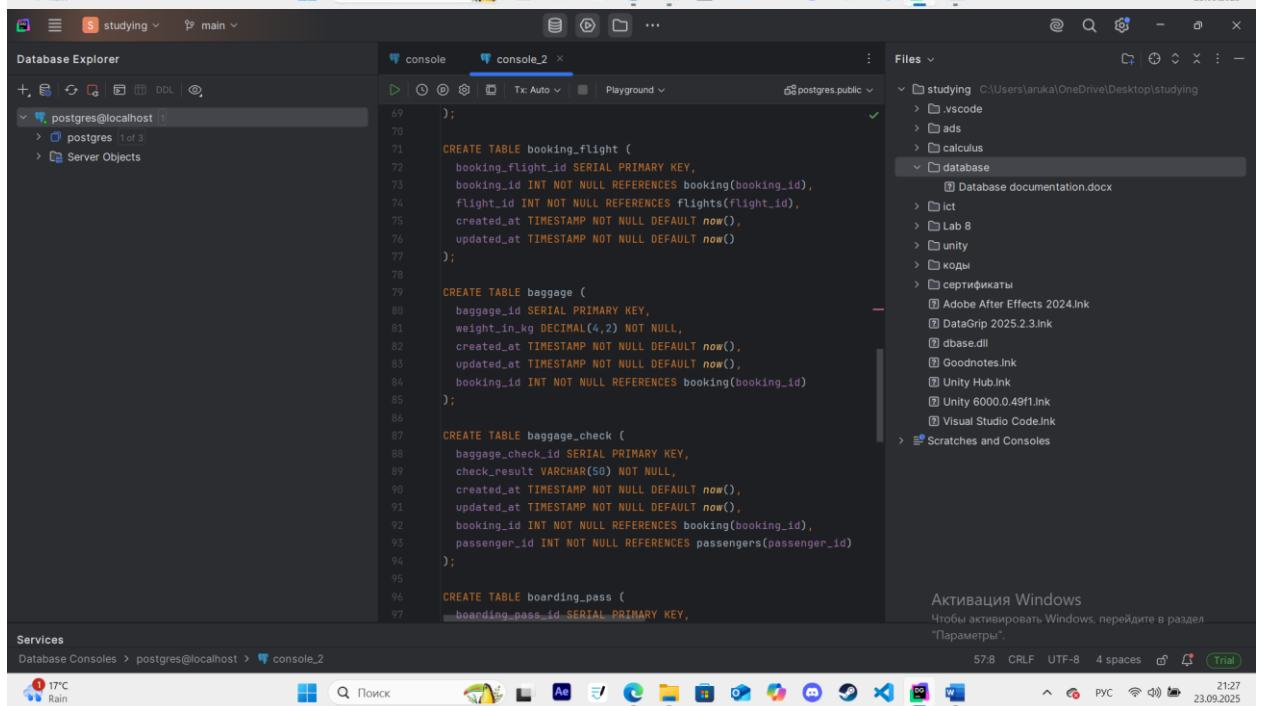
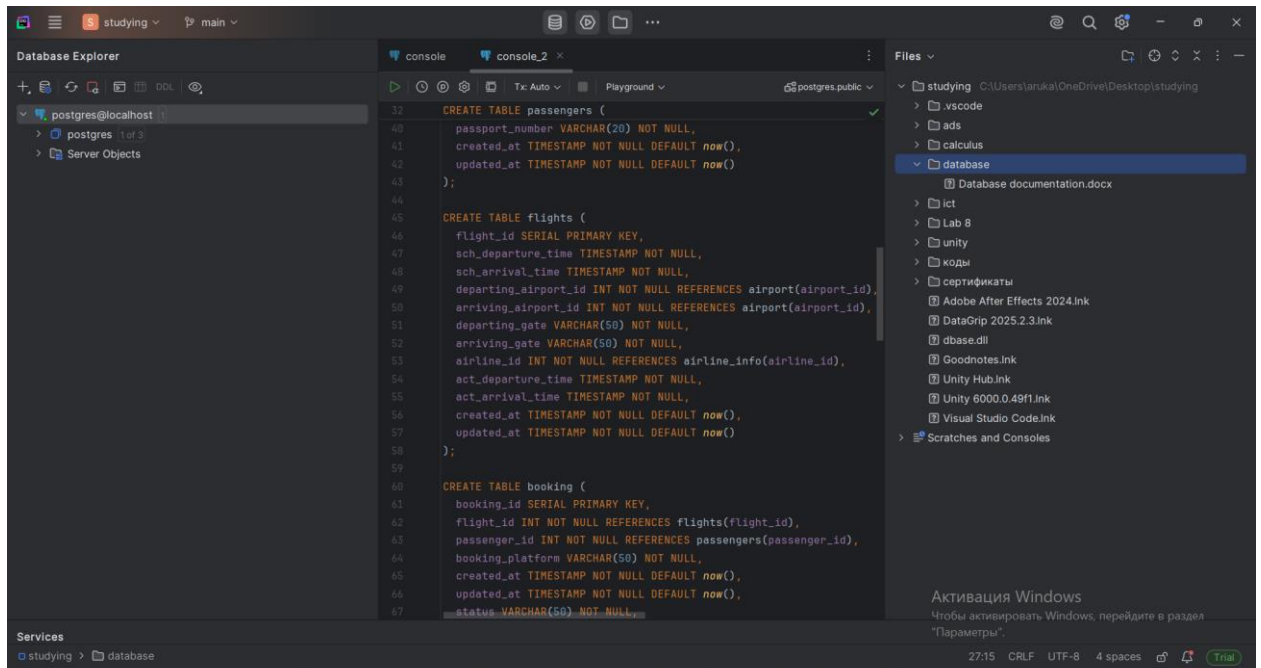
## Laboratory work 2

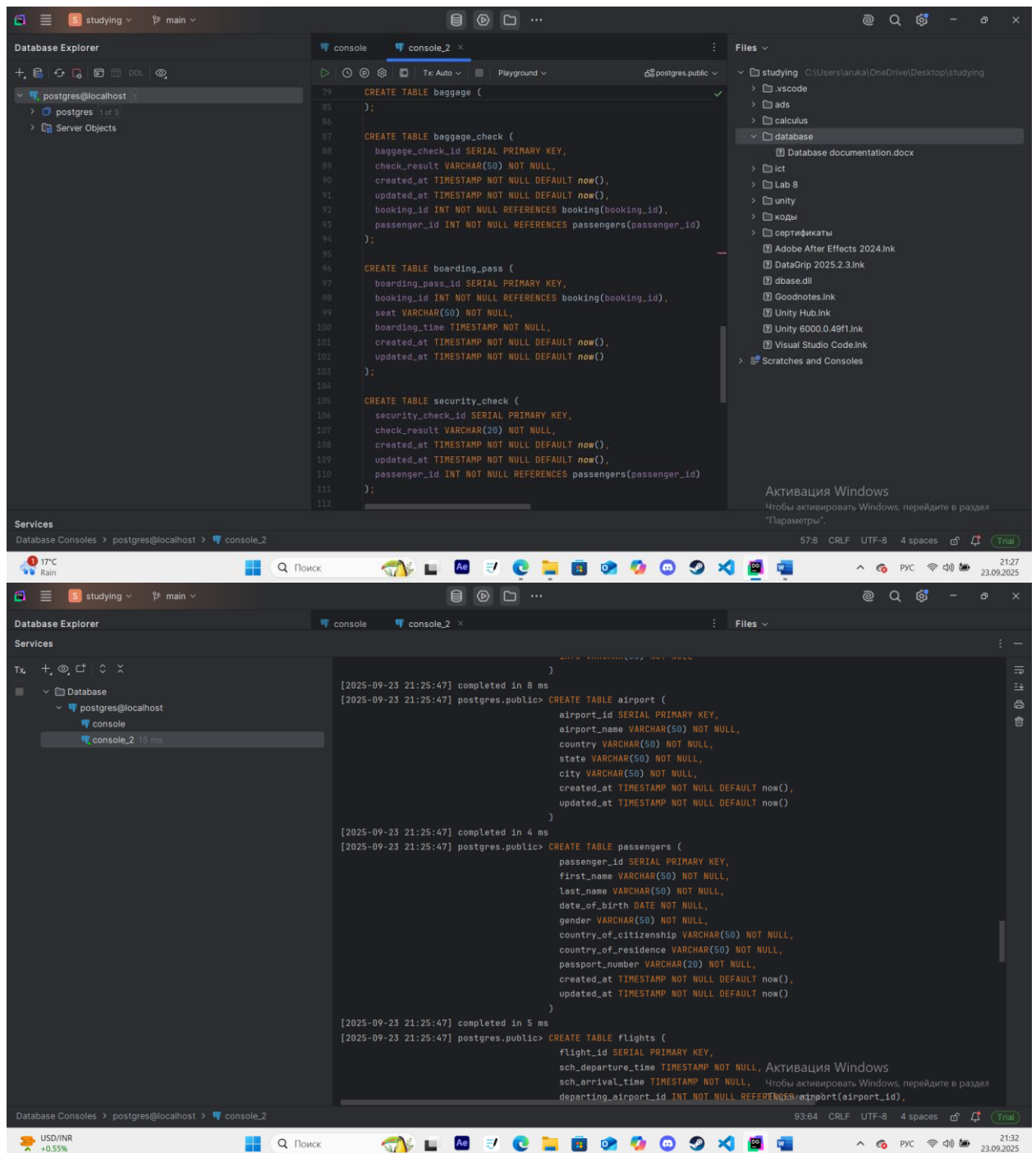
1. Create following tables with corresponding attributes;
  2. Define Primary Keys for each tables;
  3. Define for all attributes not null constraint;
- 
9. Make a relationship between following tables:
- Passengers with Secuitiry\_check, Booking, Baggage\_check by passenger\_id;
  - Booking with Baggage\_check, 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;



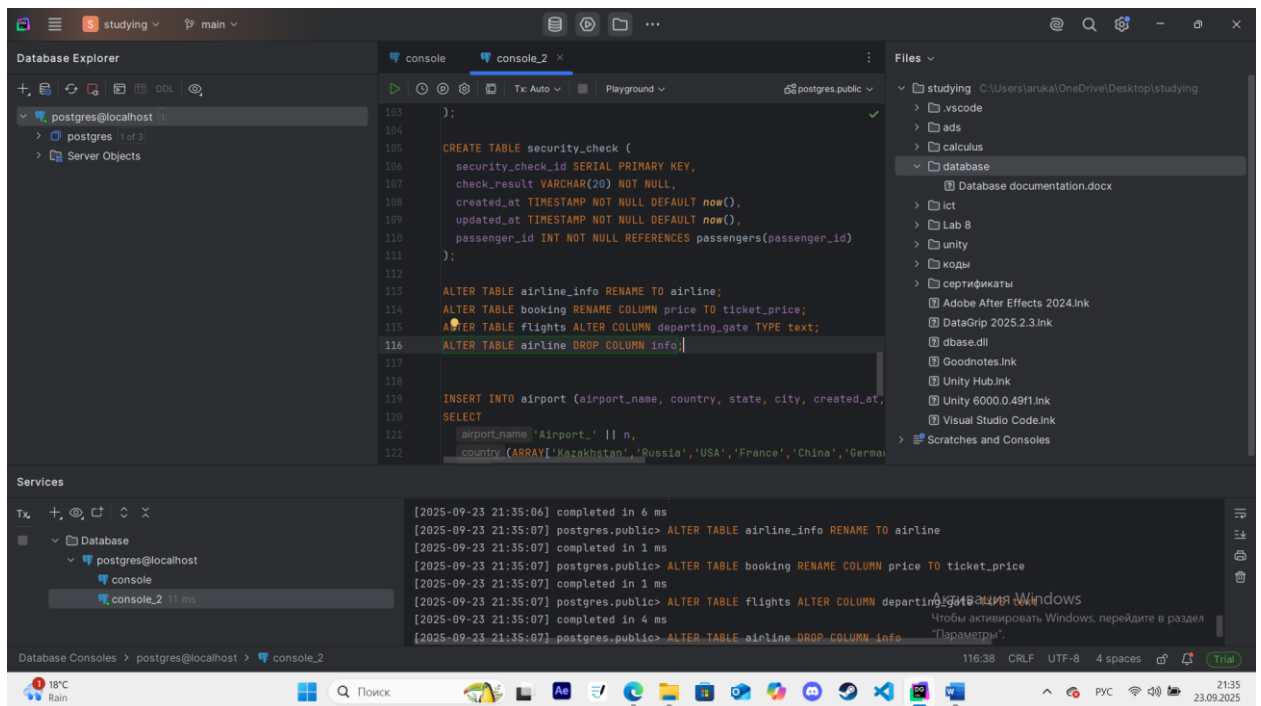
The screenshot shows a Visual Studio Code editor window with a dark theme. The 'Database Explorer' on the left shows a connection to 'postgres@localhost'. The main editor area displays SQL code for creating three tables: 'airline\_info', 'airport', and 'passengers'. The 'console' tab shows the execution of this code, with green checkmarks indicating success. The 'Files' explorer on the right shows a project structure with folders like 'studying', 'ads', 'calculus', and 'database'. The status bar at the bottom shows '9:38 CRLF UTF-8 4 spaces' and a 'Trial' watermark.

```
12 CREATE TABLE airline_info (  
13     airline_id SERIAL PRIMARY KEY,  
14     airline_code VARCHAR(30) NOT NULL,  
15     airline_name VARCHAR(50) NOT NULL,  
16     airline_country VARCHAR(50) NOT NULL,  
17     created_at TIMESTAMP NOT NULL DEFAULT now(),  
18     updated_at TIMESTAMP NOT NULL DEFAULT now(),  
19     info VARCHAR(50) NOT NULL  
20 );  
21  
22  
23 CREATE TABLE airport (  
24     airport_id SERIAL PRIMARY KEY,  
25     airport_name VARCHAR(50) NOT NULL,  
26     country VARCHAR(50) NOT NULL,  
27     state VARCHAR(50) NOT NULL,  
28     city VARCHAR(50) NOT NULL,  
29     created_at TIMESTAMP NOT NULL DEFAULT now(),  
30     updated_at TIMESTAMP NOT NULL DEFAULT now()  
31 );  
32  
33 CREATE TABLE passengers (  
34     passenger_id SERIAL PRIMARY KEY,  
35     first_name VARCHAR(50) NOT NULL,  
36     last_name VARCHAR(50) NOT NULL,  
37     date_of_birth DATE NOT NULL,  
38     gender VARCHAR(50) NOT NULL,  
39     country_of_citizenship VARCHAR(50) NOT NULL,  
40     country_of_residence VARCHAR(50) NOT NULL,  
41     passport_number VARCHAR(20) NOT NULL,
```



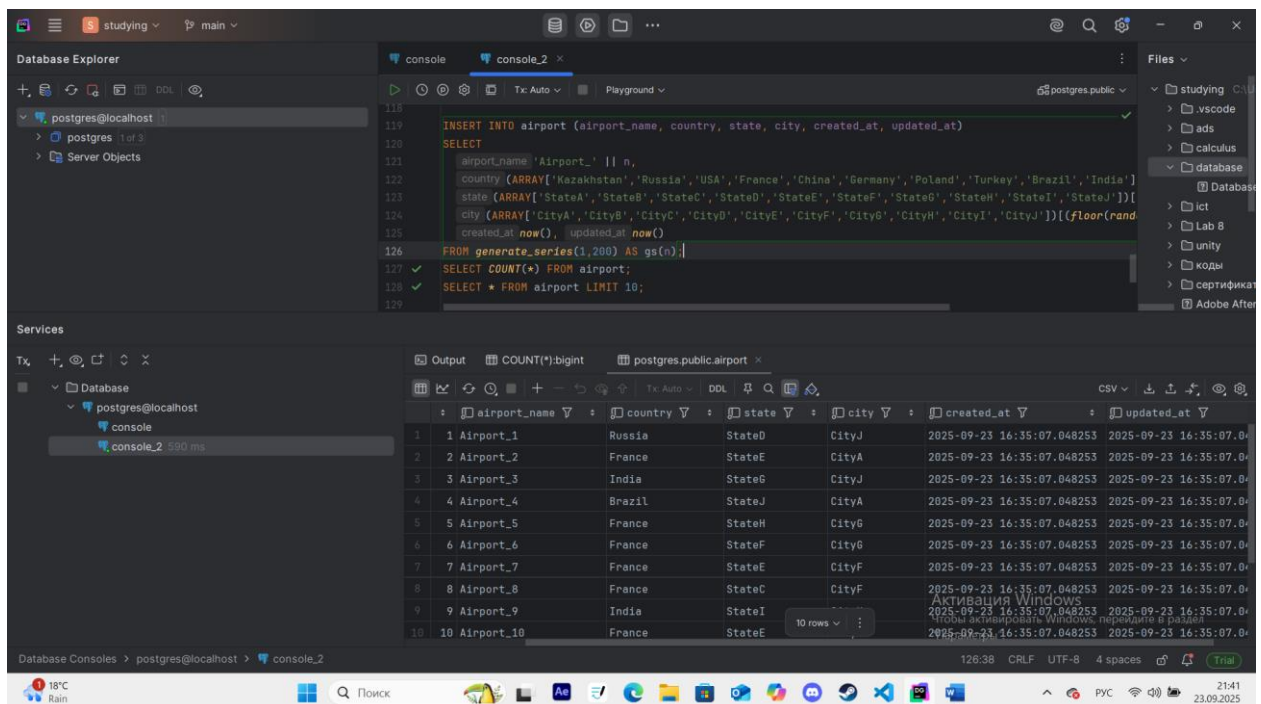


5. Rename `airline_info` table to `airline`;
6. Rename column `price` to `ticket_price` in `booking` table;
7. Change data type of `departing_gate` from `varchar(50)` to `text`;
8. Drop the column `info(varchar(50))` from the `airline` table.



## DML

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



### 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".

```
124 city (ARRAY['CityA','CityB','CityC','CityD','CityE','CityF','CityG','CityH','CityI','CityJ'])[(floor(random()*10)::int)+1],
125 created_at now(), updated_at now())
126 FROM generate_series(1,200) AS gs(n);
127 SELECT COUNT(*) FROM airport;
128 SELECT * FROM airport LIMIT 10;
129
130 INSERT INTO airline (airline_code, airline_name, airline_country, created_at, updated_at)
131 VALUES ('KZ', 'KazAir', 'Kazakhstan', created_at now(), updated_at now());
132 SELECT * FROM airline;
133
134 UPDATE airline SET airline_country = 'Turkey' WHERE airline_name = 'KazAir';
135
```

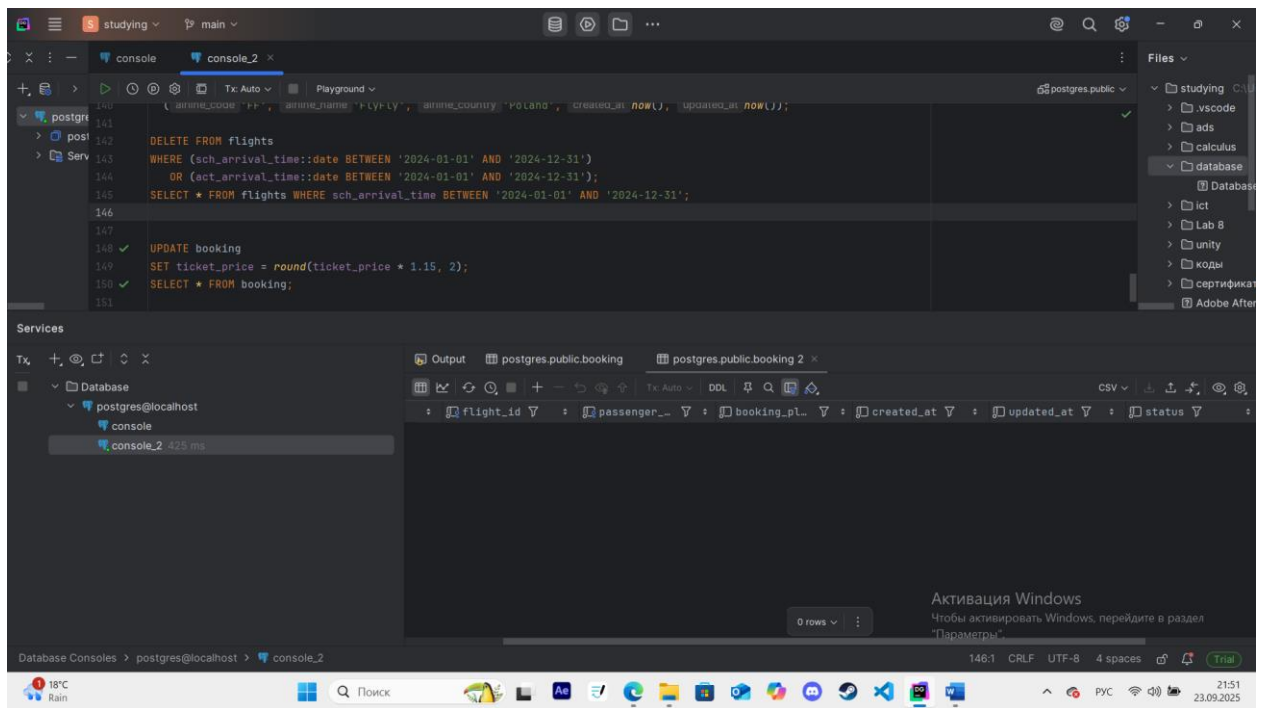
| airline_id | airline_code | airline_name | airline_count | created_at               | updated_at               |
|------------|--------------|--------------|---------------|--------------------------|--------------------------|
| 1          | KZ           | KazAir       | Turkey        | 2025-09-23 16:35:07.0... | 2025-09-23 16:35:07.0... |
| 2          | AE           | AirEasy      | France        | 2025-09-23 16:35:07.0... | 2025-09-23 16:35:07.0... |
| 3          | FH           | FlyHigh      | Brazil        | 2025-09-23 16:35:07.0... | 2025-09-23 16:35:07.0... |
| 4          | FF           | FlyFly       | Poland        | 2025-09-23 16:35:07.0... | 2025-09-23 16:35:07.0... |

5. Delete all flights whose arrival in 2024 year.

```
136 INSERT INTO airline (airline_code, airline_name, airline_country, created_at, updated_at)
137 VALUES
138 ('AE', 'AirEasy', 'France', created_at now(), updated_at now()),
139 ('FH', 'FlyHigh', 'Brazil', created_at now(), updated_at now()),
140 ('FF', 'FlyFly', 'Poland', created_at now(), updated_at now());
141
142 DELETE FROM flights
143 WHERE (sch_arrival_time::date BETWEEN '2024-01-01' AND '2024-12-31')
144 OR (act_arrival_time::date BETWEEN '2024-01-01' AND '2024-12-31');
145 SELECT * FROM flights WHERE sch_arrival_time BETWEEN '2024-01-01' AND '2024-12-31';
146
147
```

| sch_departure_time | sch_arrival_time | departing_airport_id | arriving_airport_id | depart |
|--------------------|------------------|----------------------|---------------------|--------|
|--------------------|------------------|----------------------|---------------------|--------|

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



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

