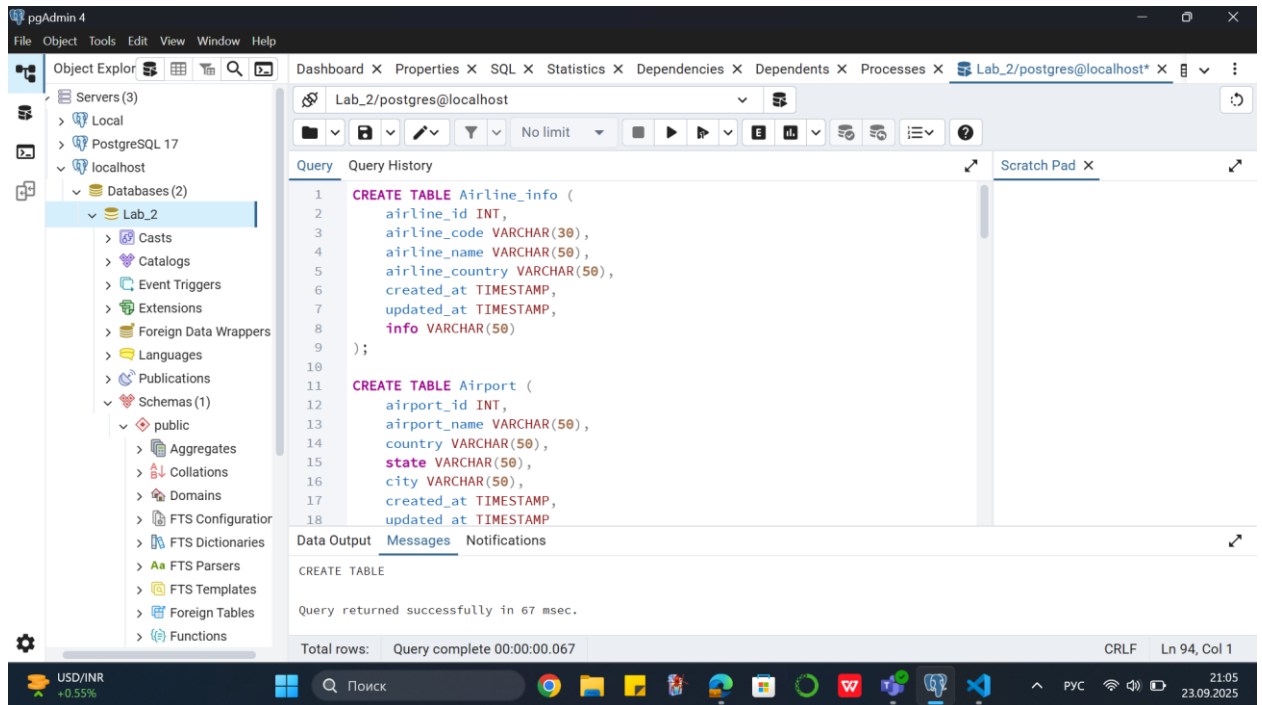


## Laboratory work #2

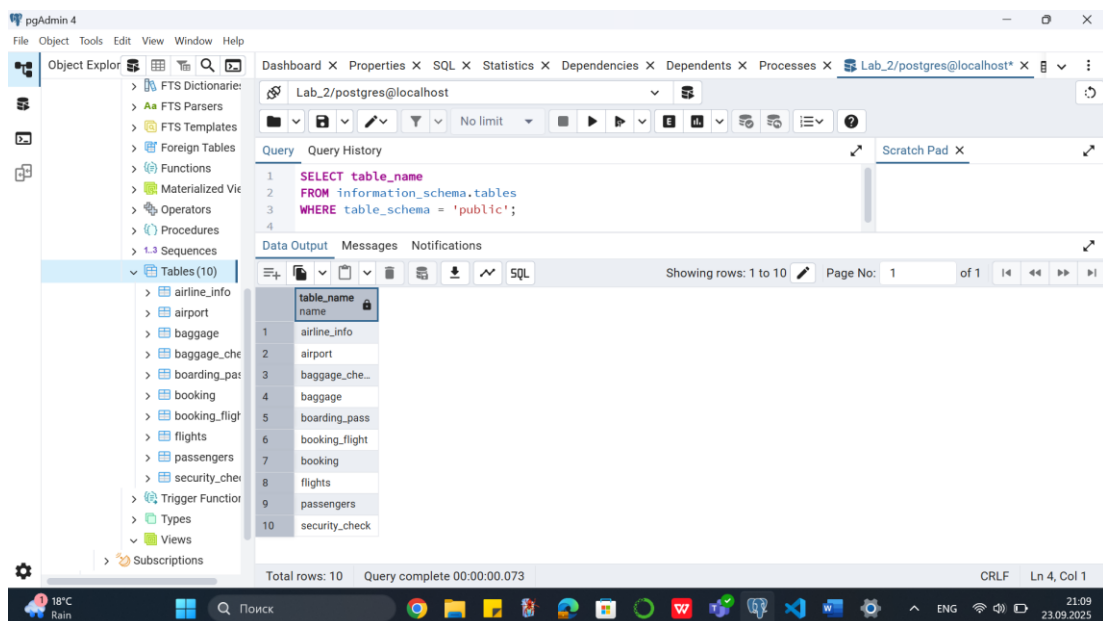
### DDL

1. Create following tables with corresponding attributes:

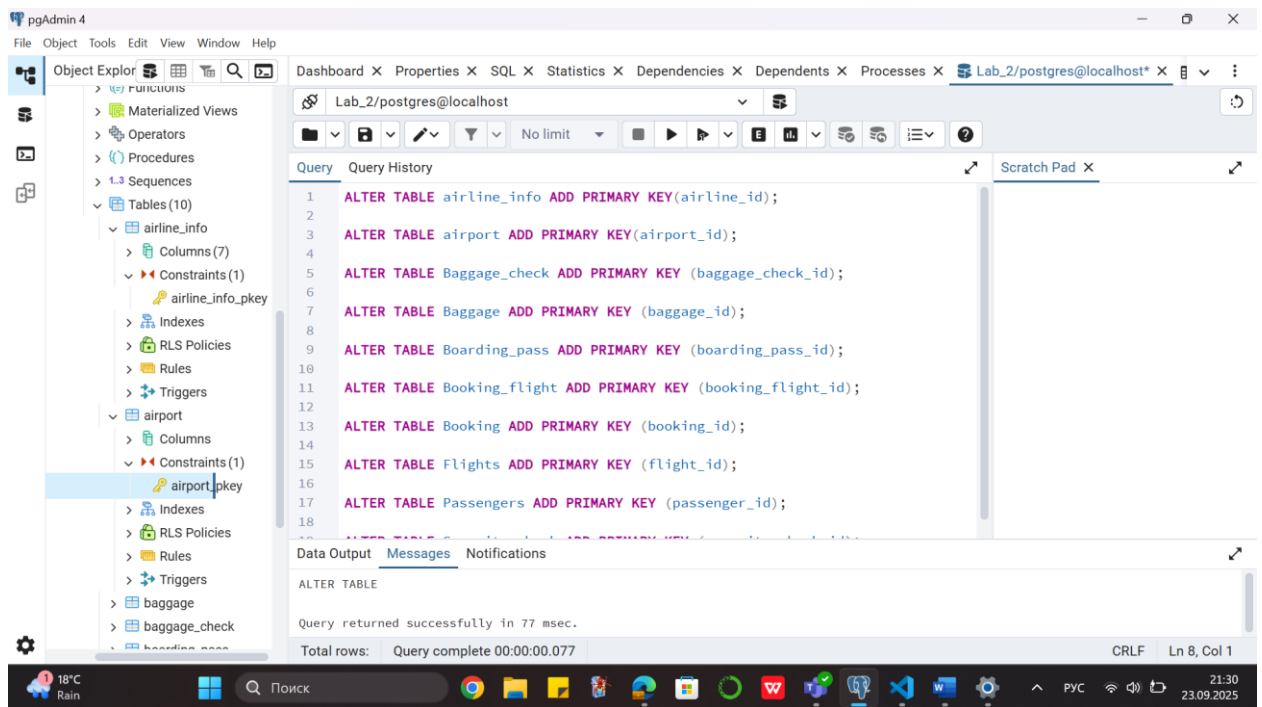
Here is the creation of tables



Here's what they look like



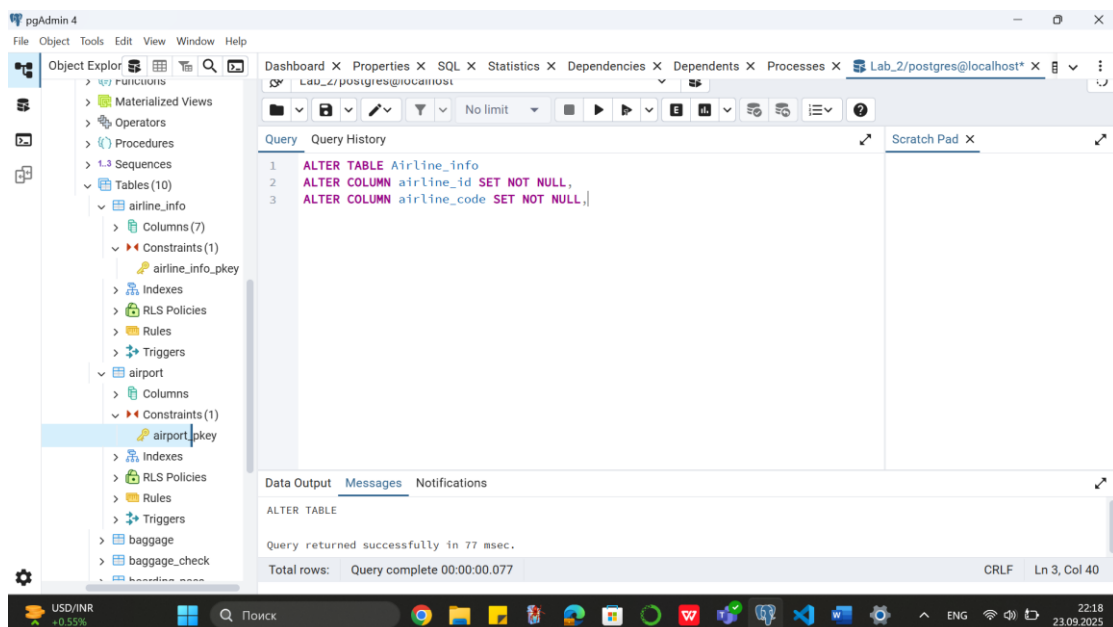
## 2. Define Primary Keys for each tables;

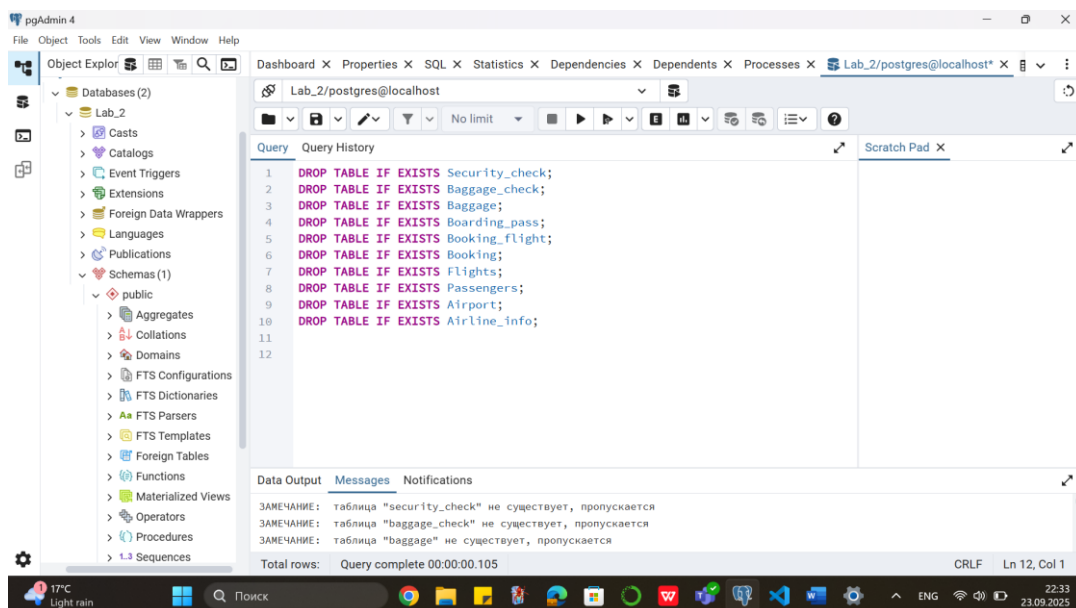
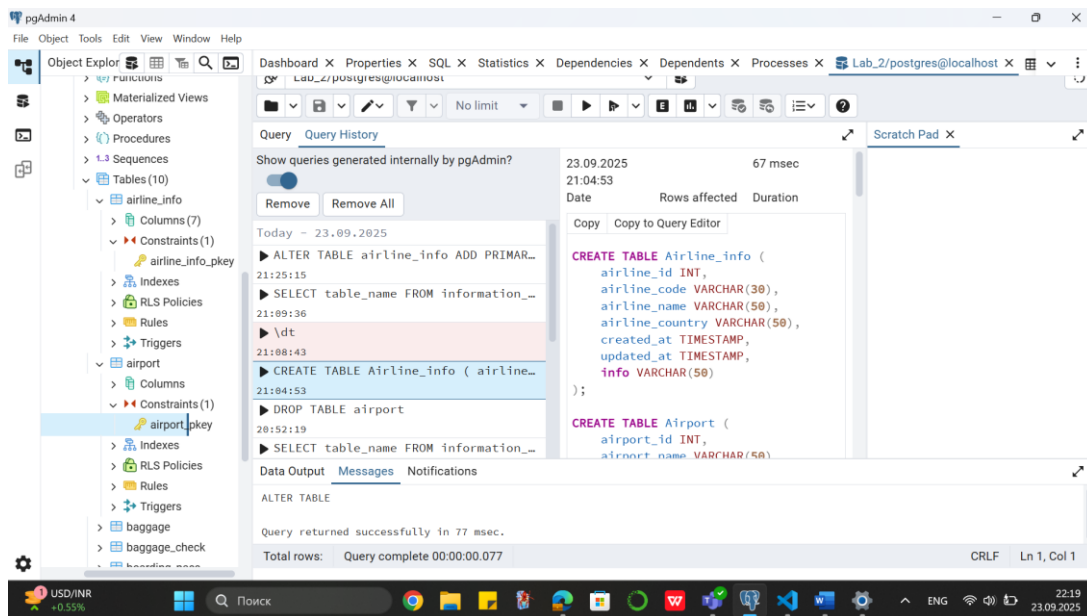


## 3. Define for all attributes not null constraint;

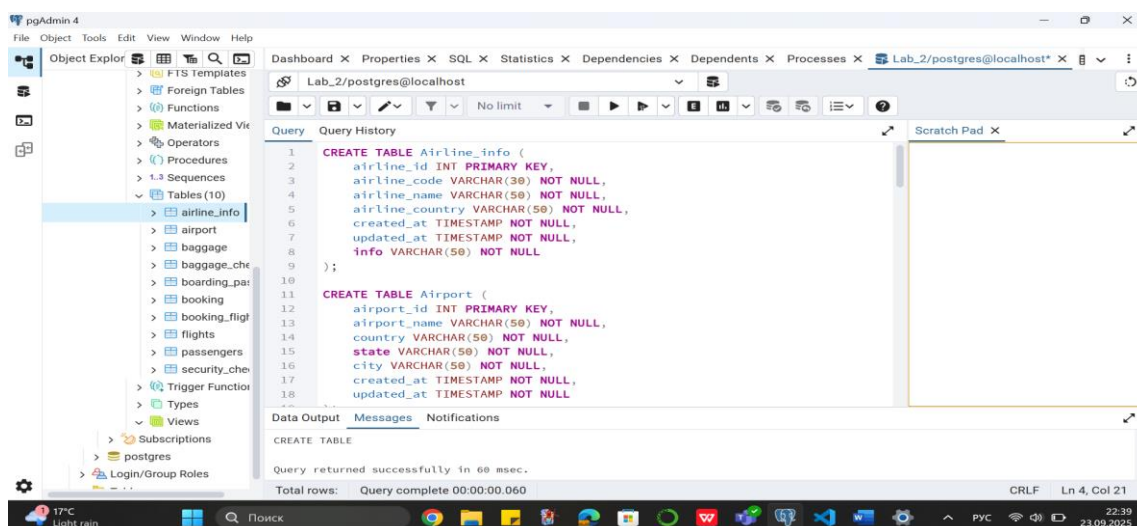
In this step I realized that it is too slow and hard, and I can do it more simply. So I decided to drop all tables, then go to the query history, copy how I created the tables, and remake all the code.

This is difficult variant:

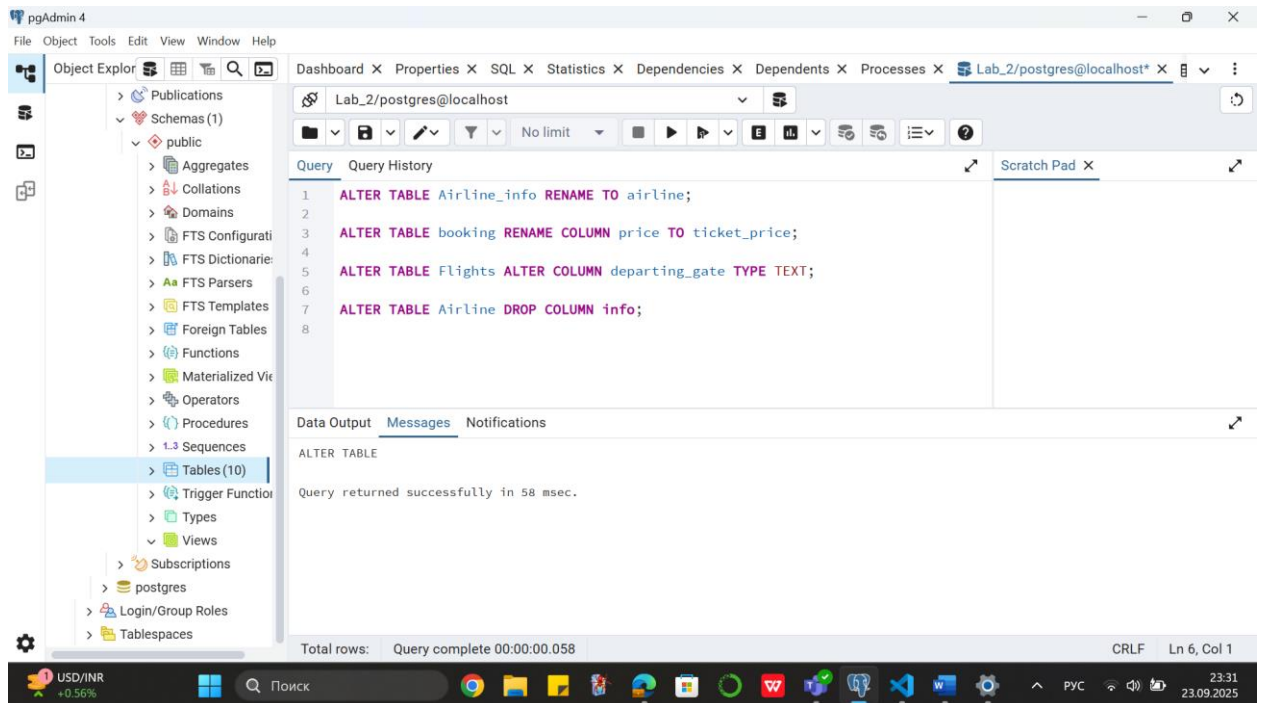




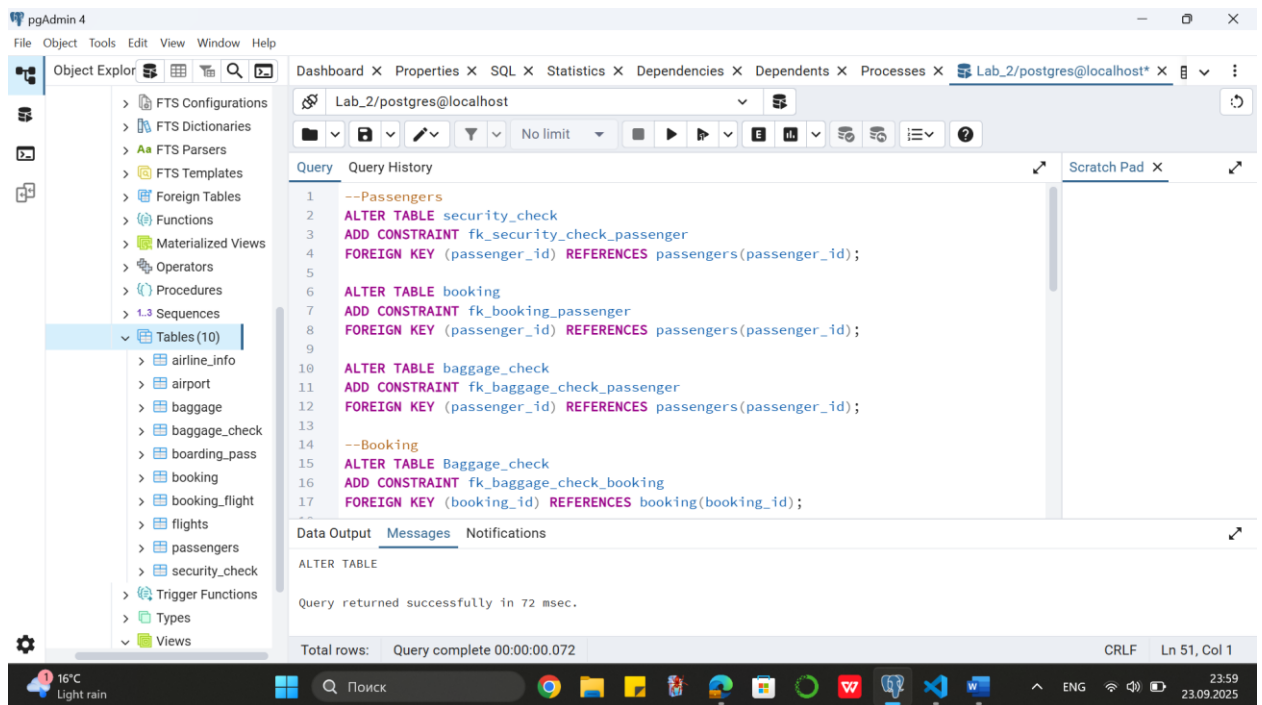
This is how it should have been



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.

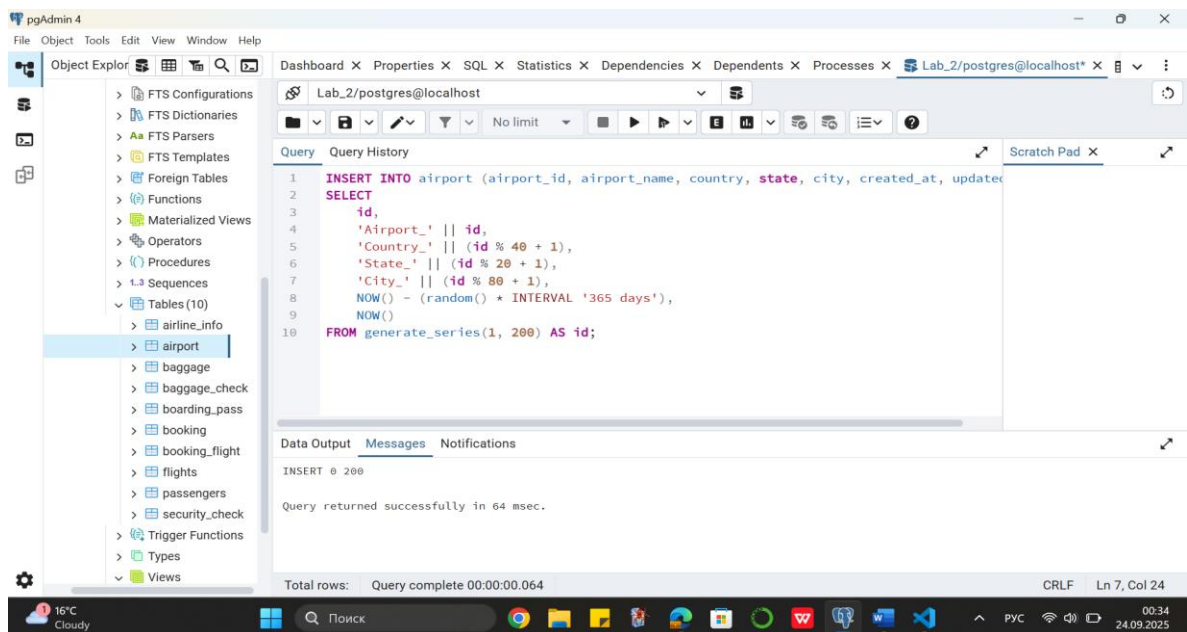


9. Make a relationship between following tables:
  - Passengers with Secutiry\_check, Booking, Baggage\_check by passenger\_id;
  - Booking with Baggage\_check, Baggage, Boarding\_pass, Booking\_flight by booking\_id;
  - Flights with Booking\_flight by flight\_id;
  - Airport with Flights by departing\_airport\_id;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



## 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".

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with 'Tables (10)' expanded and 'airport' selected. The main pane shows a SQL query in the 'Query' tab. The query is as follows:

```
1 INSERT INTO airline (airline_id, airline_code, airline_name, airline_country, created_at)
2 VALUES (1, 'KZR', 'KazAir', 'Kazakhstan', NOW(), NOW());
3
4 UPDATE airline SET airline_country = 'Turkey' WHERE airline_name = 'KazAir';
5
6 INSERT INTO airline (airline_id, airline_code, airline_name, airline_country, created_at)
7 VALUES
8 (2, 'AEZ', 'AirEasy', 'France', NOW(), NOW()),
9 (3, 'FLH', 'FlyHigh', 'Brazil', NOW(), NOW()),
10 (4, 'FLF', 'FlyFly', 'Poland', NOW(), NOW());
11 SELECT * FROM airline LIMIT 10
```

The 'Data Output' tab shows the results of the query, displaying 4 rows of data:

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

The status bar at the bottom indicates 'Total rows: 4' and 'Query complete 00:00:00.064'.

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.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure, with 'Tables (10)' expanded and 'airport' selected. The main pane shows a SQL query in the 'Query' tab. The query is as follows:

```
1 DELETE FROM flights WHERE EXTRACT(YEAR FROM sch_arrival_time) = 2024;
2
3 UPDATE booking SET ticket_price = ticket_price * 1.15;
4
5 DELETE FROM booking WHERE ticket_price < 10000;
```

The 'Data Output' tab shows the results of the query, displaying 0 rows of data. The status bar at the bottom indicates 'Total rows: 0' and 'Query complete 00:00:00.068'.