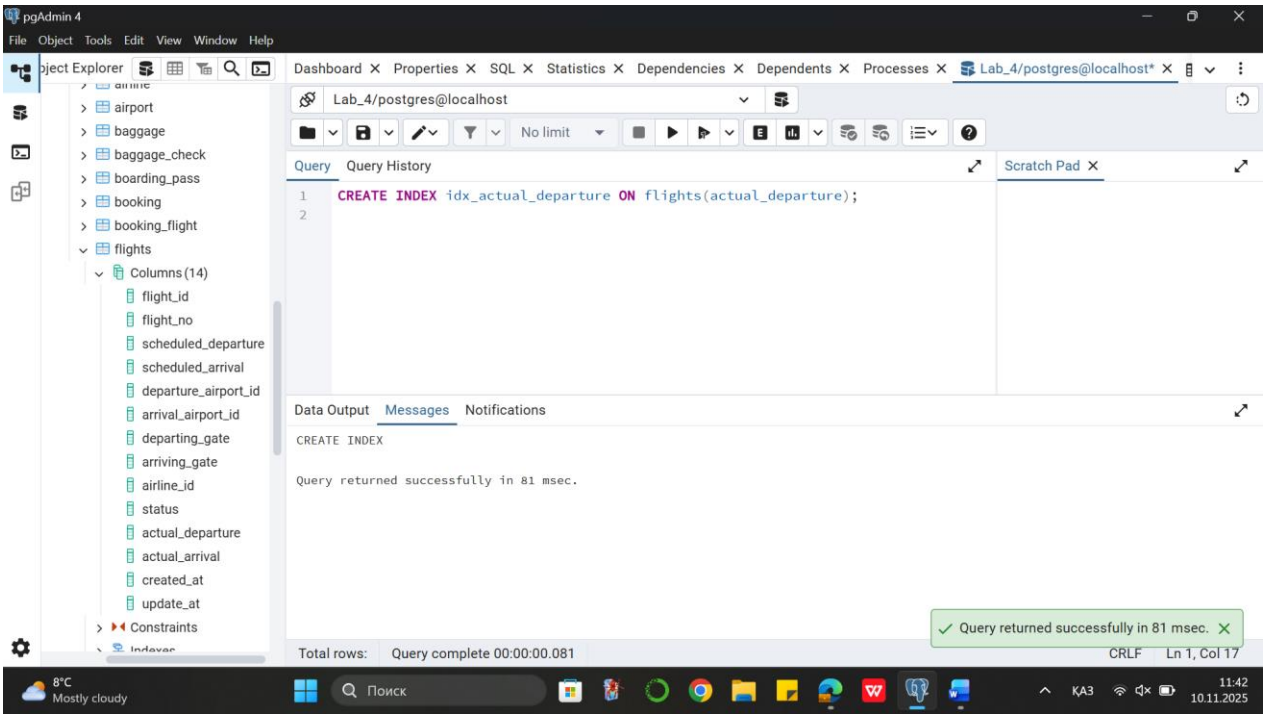
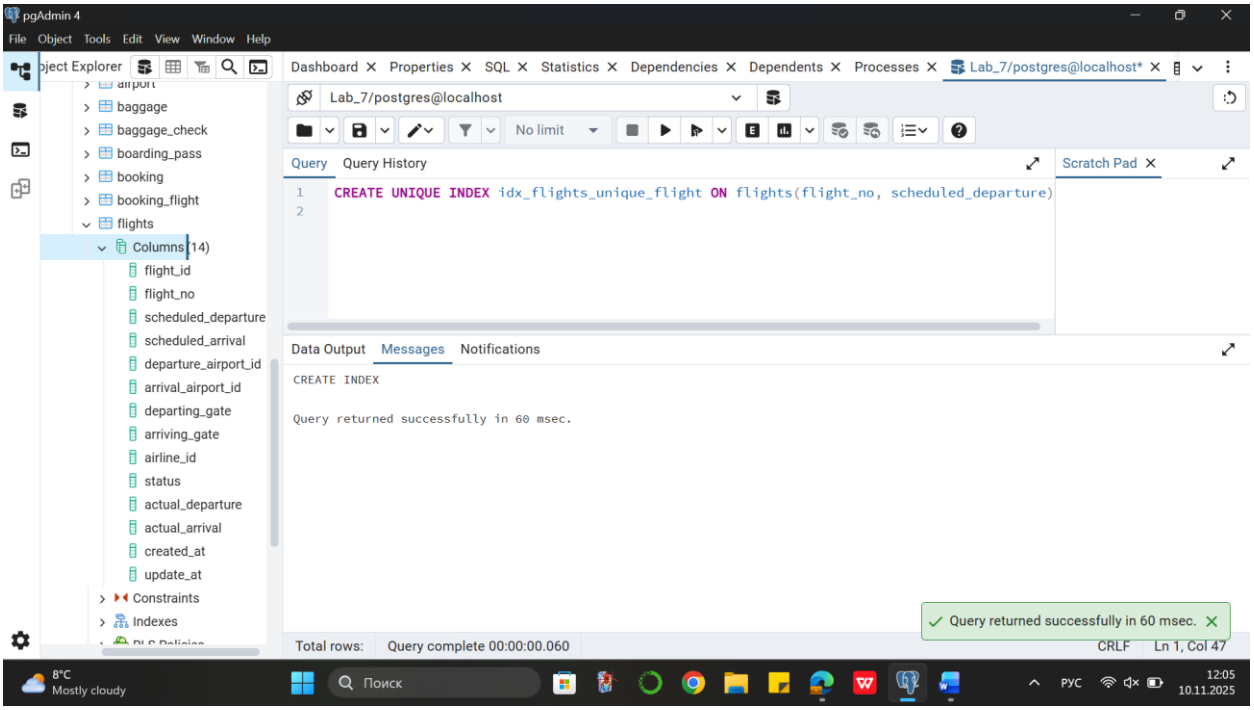
Laboratory work 7

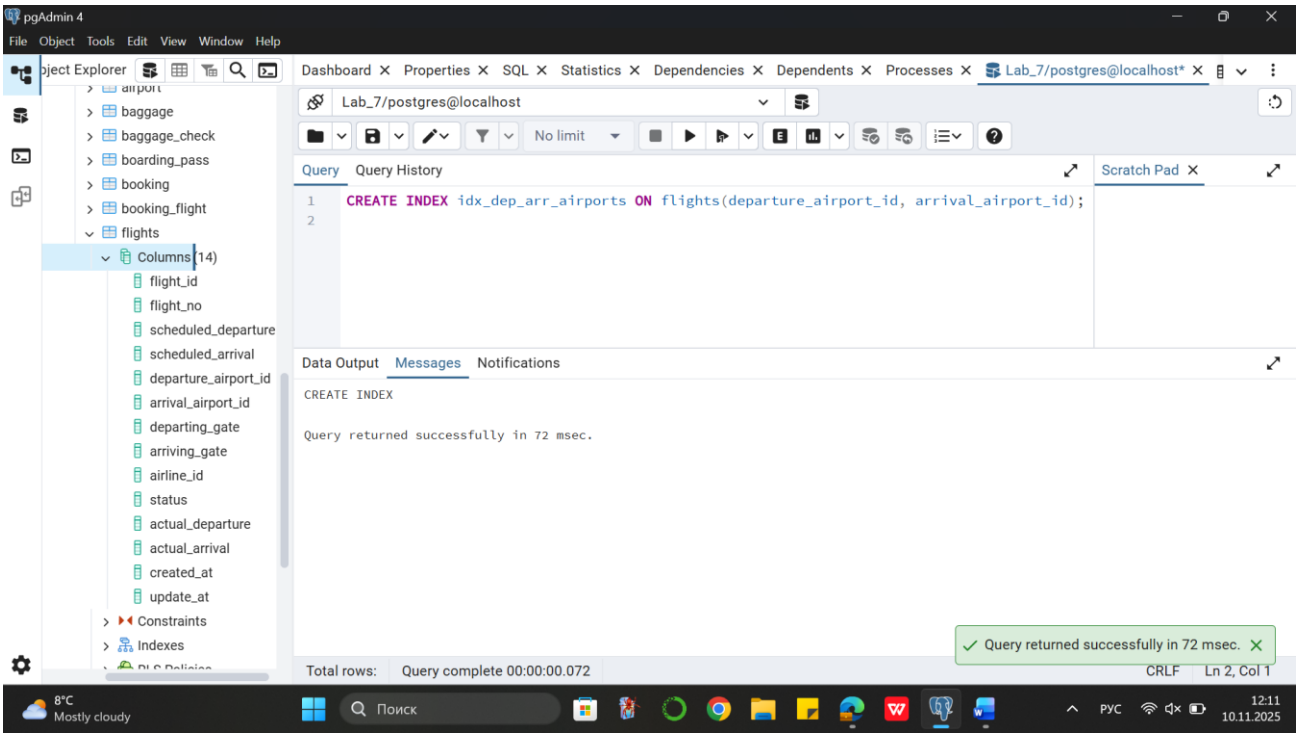
1. Create an index on the actual_departure column in the flights table.



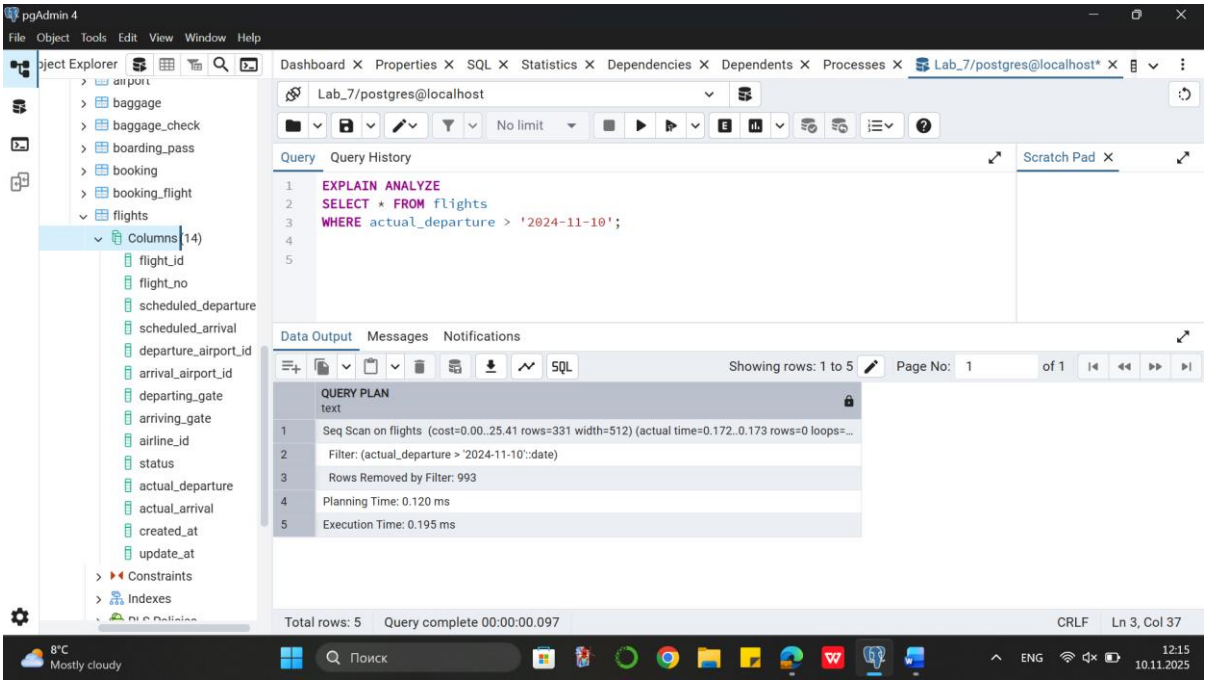
2. Create a unique index to ensure flight_no and scheduled_departure combinations are unique.



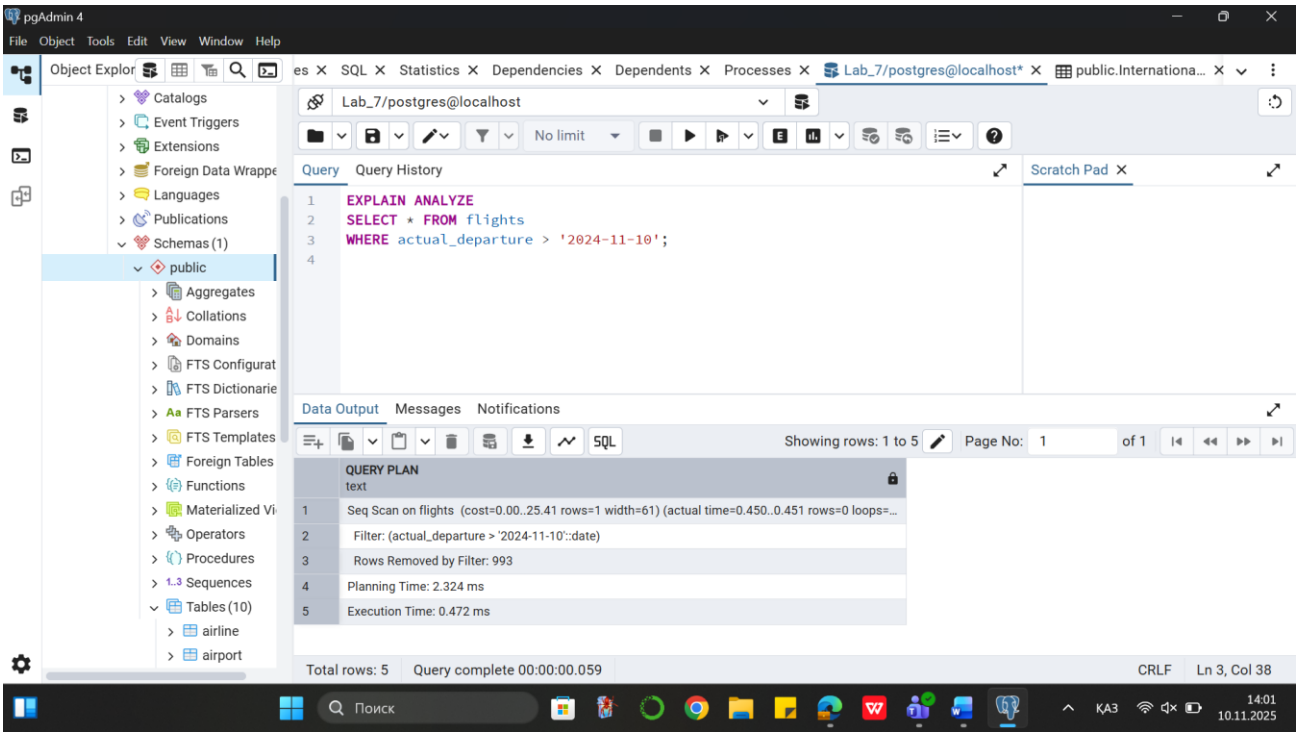
3. Create a composite index on the departure_airport_id and arrival_airport_id columns.



4. Evaluate the difference in query performance with and without indexes. Measure performance differences. Without indexes:

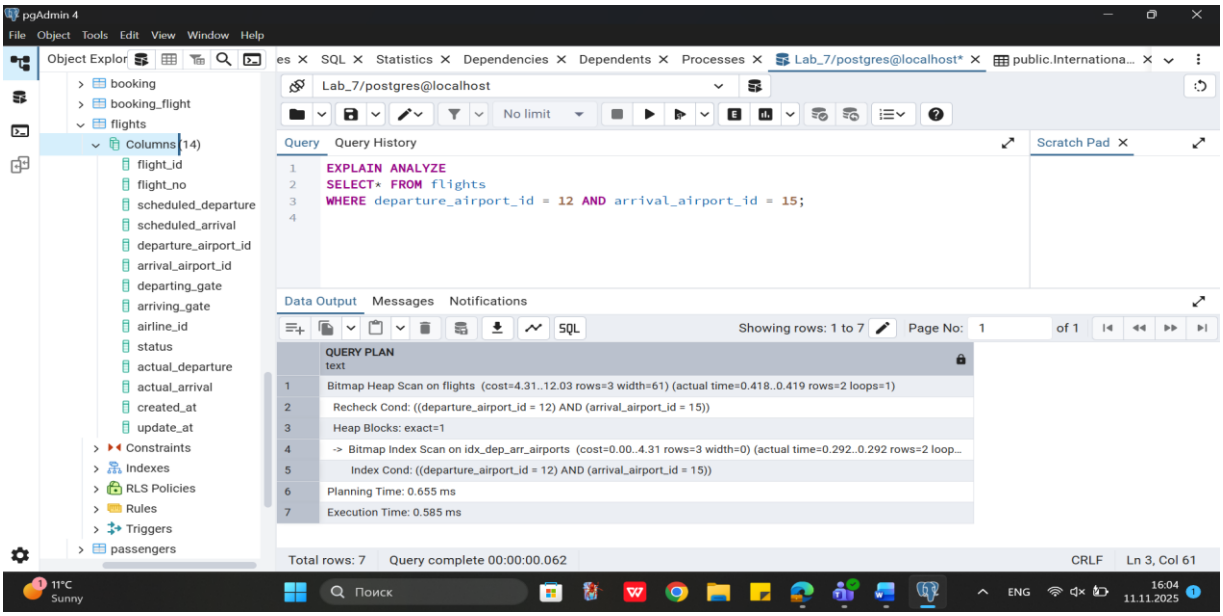


With indexes:

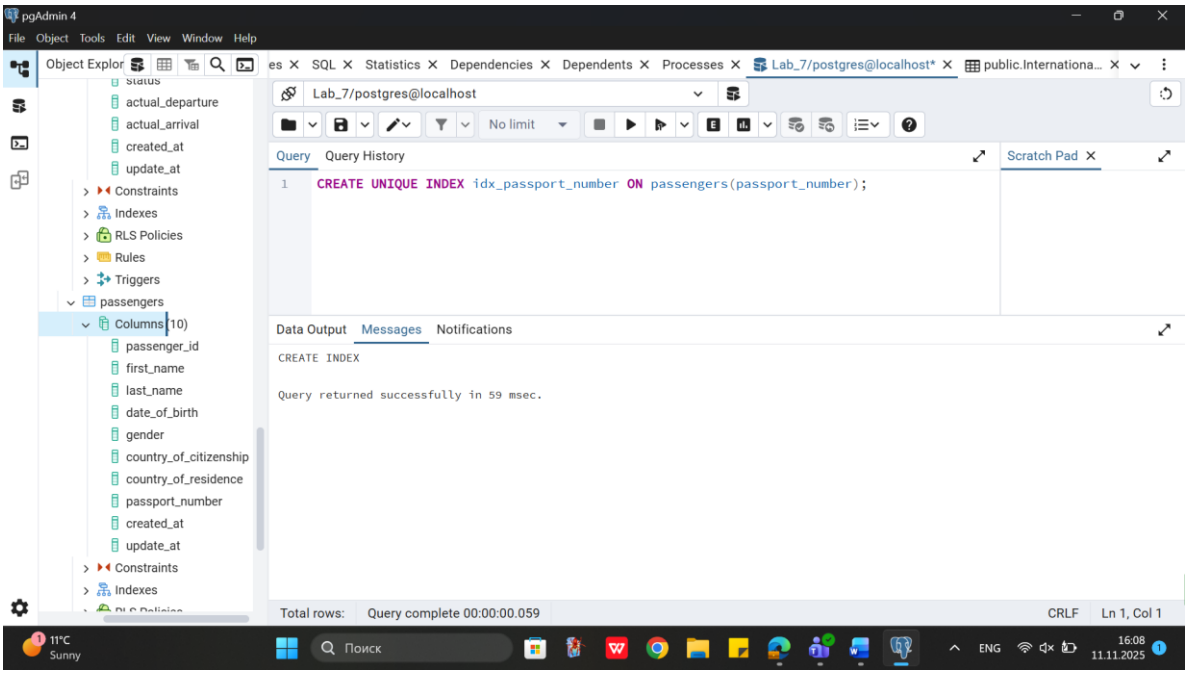


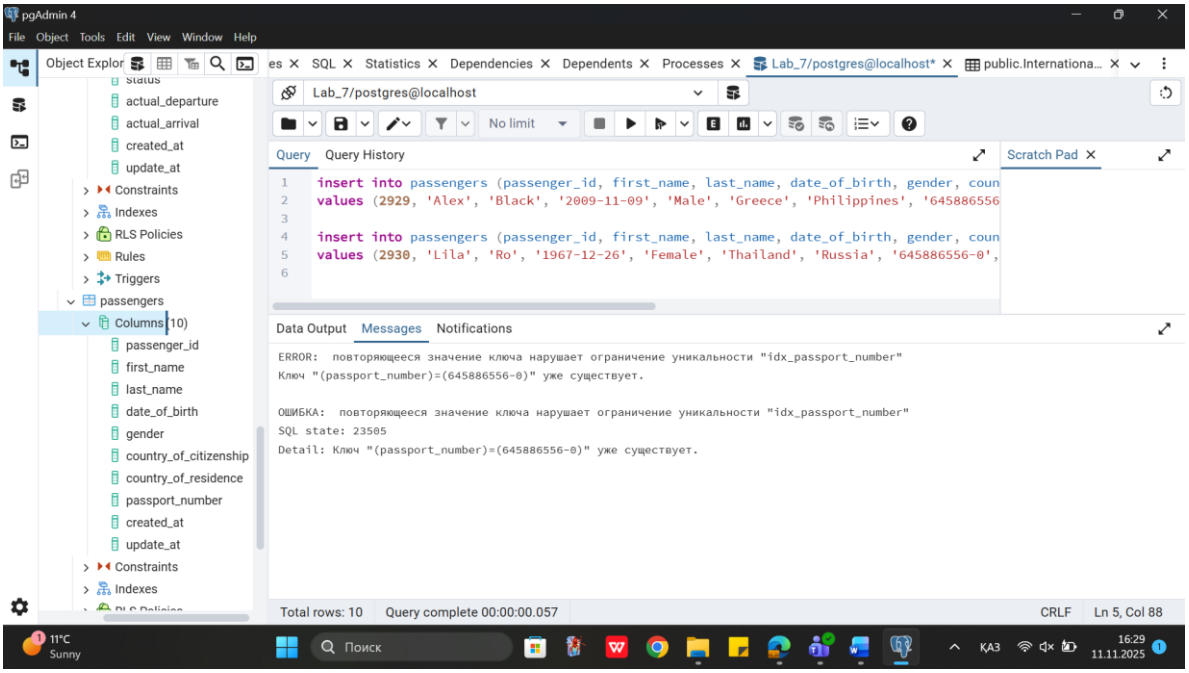
After the indexes were created, the query execution time decreased. Before creating the index, PostgreSQL used Seq Scan, which takes longer. After creating the index, PostgreSQL started using Index Scan, which selects only the necessary rows by indexes. This reduces the search time, especially for large tables.

5. Use EXPLAIN ANALYZE to check index usage in a query filtering by departure_airport and arrival_airport.



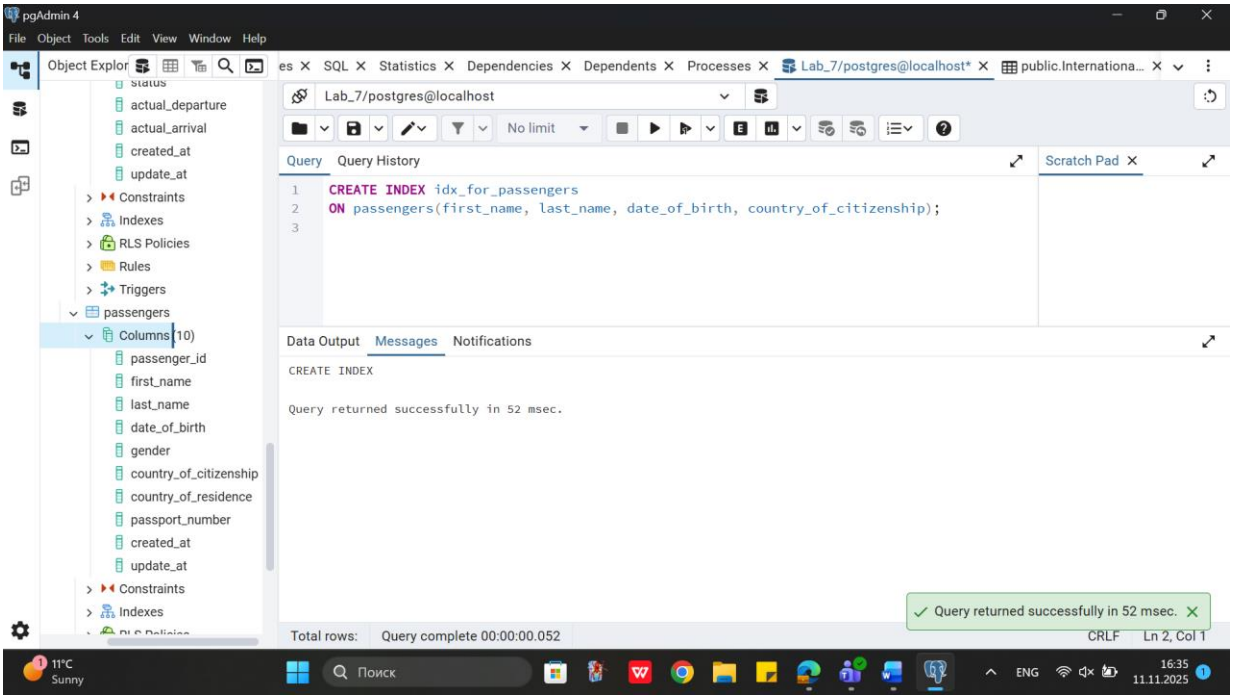
6. Create a unique index for the passport_number of the Passengers table. Check if the index was created or not. Insert into the table two new passengers. Explain in your own words what is going on in the output?

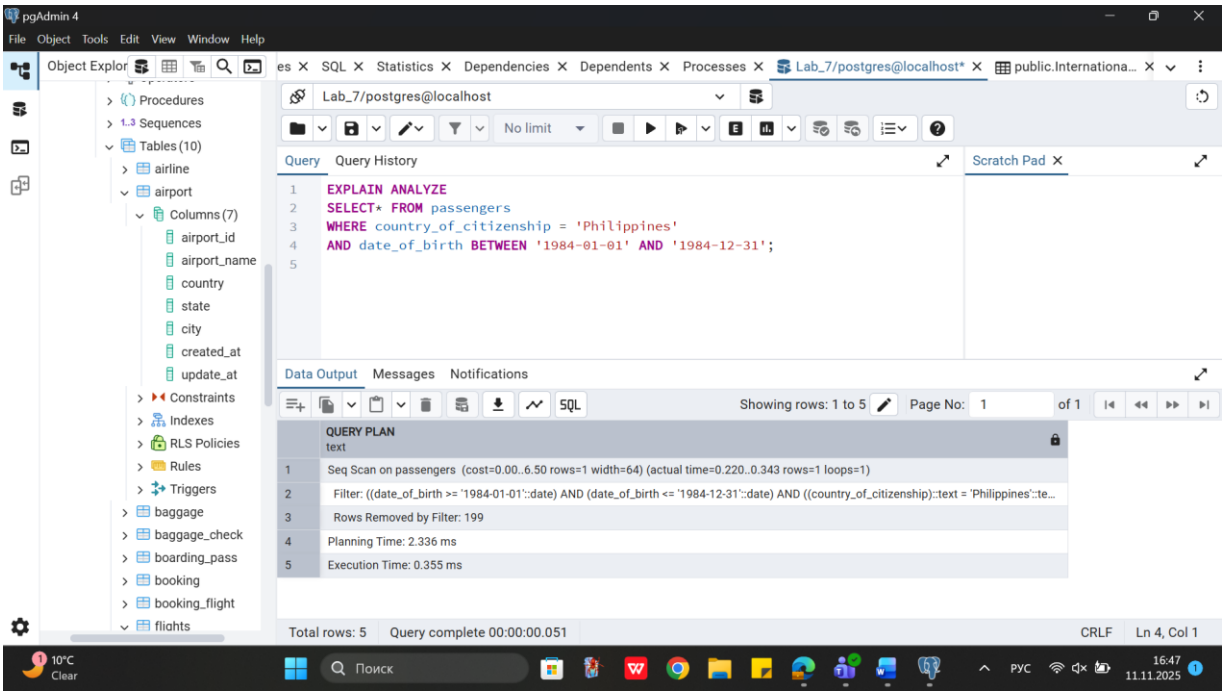




When trying to insert a passenger with an existing passport number, PostgreSQL returned an error. This proves that the unique idx_passport_number index has been successfully created and is working correctly. It prevents duplicate passport numbers in the passengers table.

7. Create an index for the Passengers table. Use for that first name, last name, date of birth and country of citizenship. Then, write a SQL query to find a passenger who was born in Philippines and was born in 1984 and check if the query uses indexes or not. Give the explanation of the results.





A composite index of 4 fields has been create. But in the query, filtering was performed on only two of these fields. The index was not used because the search did not start from the first index field.

8. Write a SQL query to list indexes for table Passengers. After delete the created indexes.