

Laboratory work 8

1. Create a view to show details of all flights that are departing on a specific date.

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' displays the 'flights' table with 14 columns: flight_id, flight_no, scheduled_departure, scheduled_arrival, departure_airport_id, arrival_airport_id, departing_gate, arriving_gate, airline_id, status, actual_departure, actual_arrival, created_at, and update_at. The 'scheduled_departure' column is selected. In the center, the 'Query' editor contains the following SQL code:

```
1 CREATE VIEW all_view AS
2 SELECT*FROM flights WHERE scheduled_departure::date = CURRENT_DATE;
```

The 'Messages' tab at the bottom shows the message: 'CREATE VIEW' and 'Query returned successfully in 58 msec.' A green notification bar at the bottom right confirms: 'Query returned successfully in 58 msec.'

2. Create a view that shows bookings for flights scheduled to depart within the next week.

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' displays the 'public' schema with various objects. In the center, the 'Query' editor contains the following SQL code:

```
1 CREATE VIEW show_booking AS
2 SELECT
3     f.scheduled_departure,
4     b.booking_id,
5     (p.last_name||' '|| p.first_name) AS full_name,
6     f.flight_id
7 FROM flights f
8 JOIN booking_flight bo ON f.flight_id = bo.flight_id
9 JOIN booking b ON b.booking_id = bo.booking_id
10 JOIN passengers p ON p.passenger_id = b.passenger_id
11 WHERE f.scheduled_departure BETWEEN NOW() AND NOW() + INTERVAL '7 days';
```

The 'Messages' tab at the bottom shows the message: 'CREATE VIEW' and 'Query returned successfully in 50 msec.' A green notification bar at the bottom right confirms: 'Query returned successfully in 50 msec.'

3. Create a view to show the top 5 most popular flight routes based on the number of bookings.

The screenshot shows the pgAdmin 4 interface. On the left, the 'flights' table is selected under 'Columns (14)'. The 'Query' tab is active, displaying the following SQL code:

```
1 CREATE VIEW popular_routes AS
2 SELECT
3     f.departure_airport_id,
4     f.arrival_airport_id,
5     count(b.booking_id) AS total
6 FROM flights f
7 JOIN booking_flight bo ON f.flight_id = bo.flight_id
8 JOIN booking b ON b.booking_id = bo.booking_id
9 GROUP BY f.departure_airport_id, f.arrival_airport_id
10 ORDER BY total DESC LIMIT 5;
```

The 'Data Output' tab shows the results of the query:

departure_airport_id	arrival_airport_id	total
10	7	18
4	10	15
13	4	14
6	16	14
14	7	11

Total rows: 5. Query complete 00:00:00.051.

4. Create a view that lists all flights for a specific airline.

The screenshot shows the pgAdmin 4 interface. On the left, the 'airline' table is selected under 'Columns (7)'. The 'Query' tab is active, displaying the following SQL code:

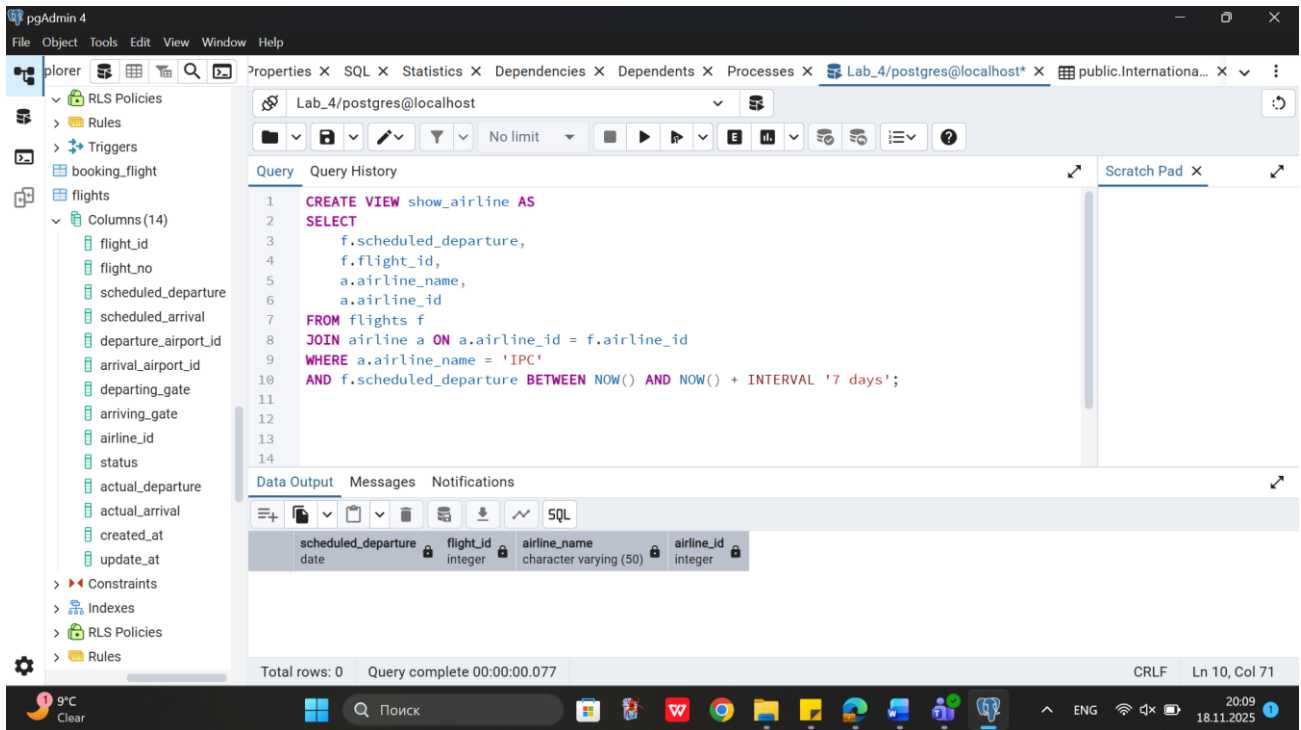
```
1 CREATE VIEW specific_airline AS
2 SELECT a.airline_name, a.airline_id, f.flight_id
3 FROM flights f
4 JOIN airline a ON a.airline_id = f.airline_id
5 WHERE a.airline_name = 'IPC';
6
7 SELECT * FROM specific_airline;
```

The 'Data Output' tab shows the results of the query:

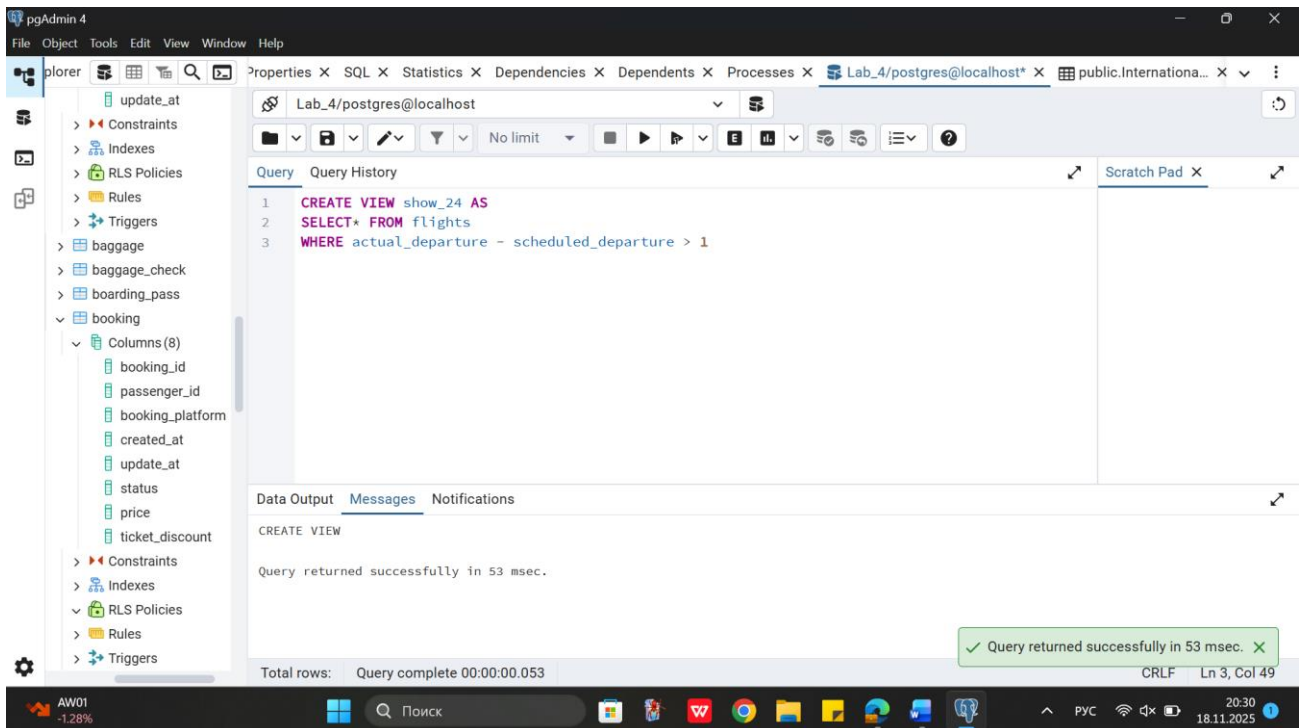
airline_name	airline_id	flight_id
IPC	1	13
IPC	1	33
IPC	1	36
IPC	1	73
IPC	1	78
IPC	1	143
IPC	1	170

Total rows: 32. Query complete 00:00:00.056.

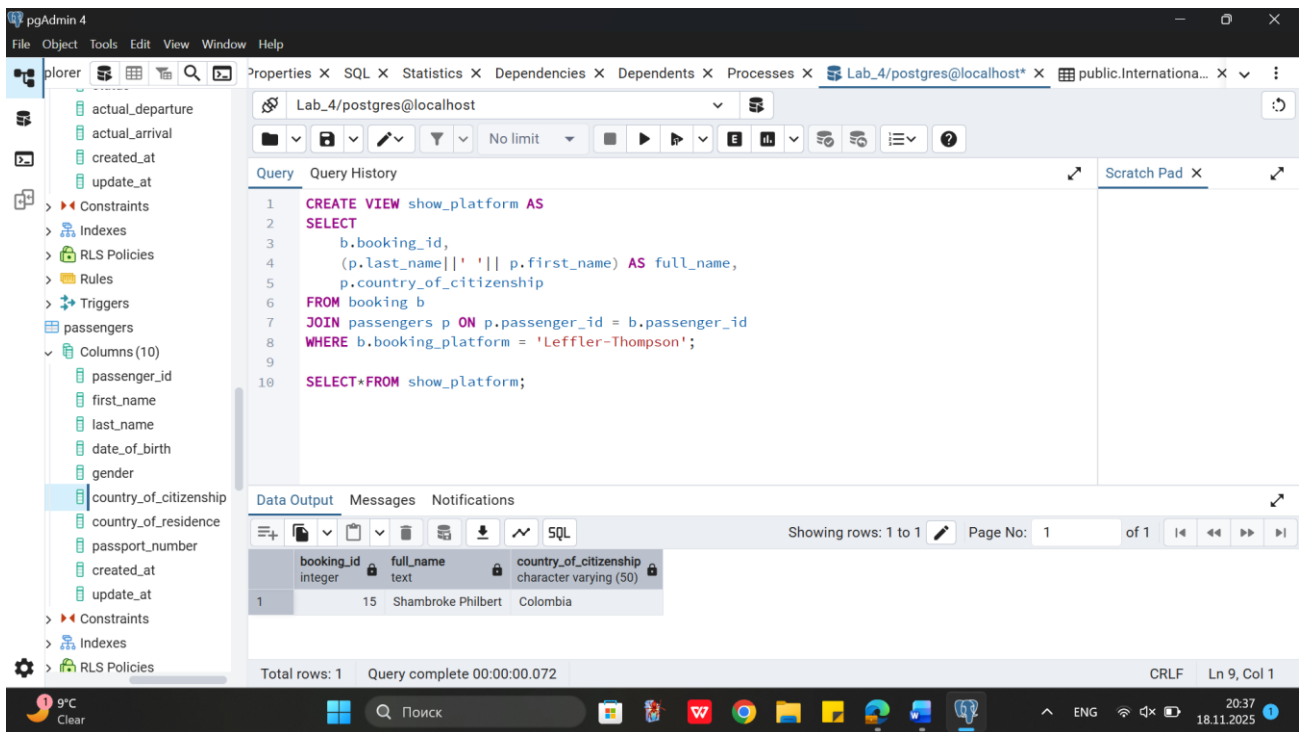
5. Modify the view created in task 4 to show only flights departing within the next 7 days for a specific airline



6. Create a view to show flights that are delayed by more than 24 hours.



7. Create a view in which you can display the full name and country of origin of passengers who made bookings on Leffler-Thompson platform. Then show the list of that passengers.



The screenshot shows the pgAdmin 4 interface. On the left, the 'passengers' table is expanded, showing columns like 'passenger_id', 'first_name', 'last_name', 'date_of_birth', 'gender', 'country_of_citizenship', 'country_of_residence', 'passport_number', 'created_at', and 'update_at'. The 'country_of_citizenship' column is selected. In the center, the SQL editor shows the following query:

```

1 CREATE VIEW show_platform AS
2 SELECT
3     b.booking_id,
4     (p.last_name || ' ' || p.first_name) AS full_name,
5     p.country_of_citizenship
6 FROM booking b
7 JOIN passengers p ON p.passenger_id = b.passenger_id
8 WHERE b.booking_platform = 'Leffler-Thompson';
9
10 SELECT*FROM show_platform;

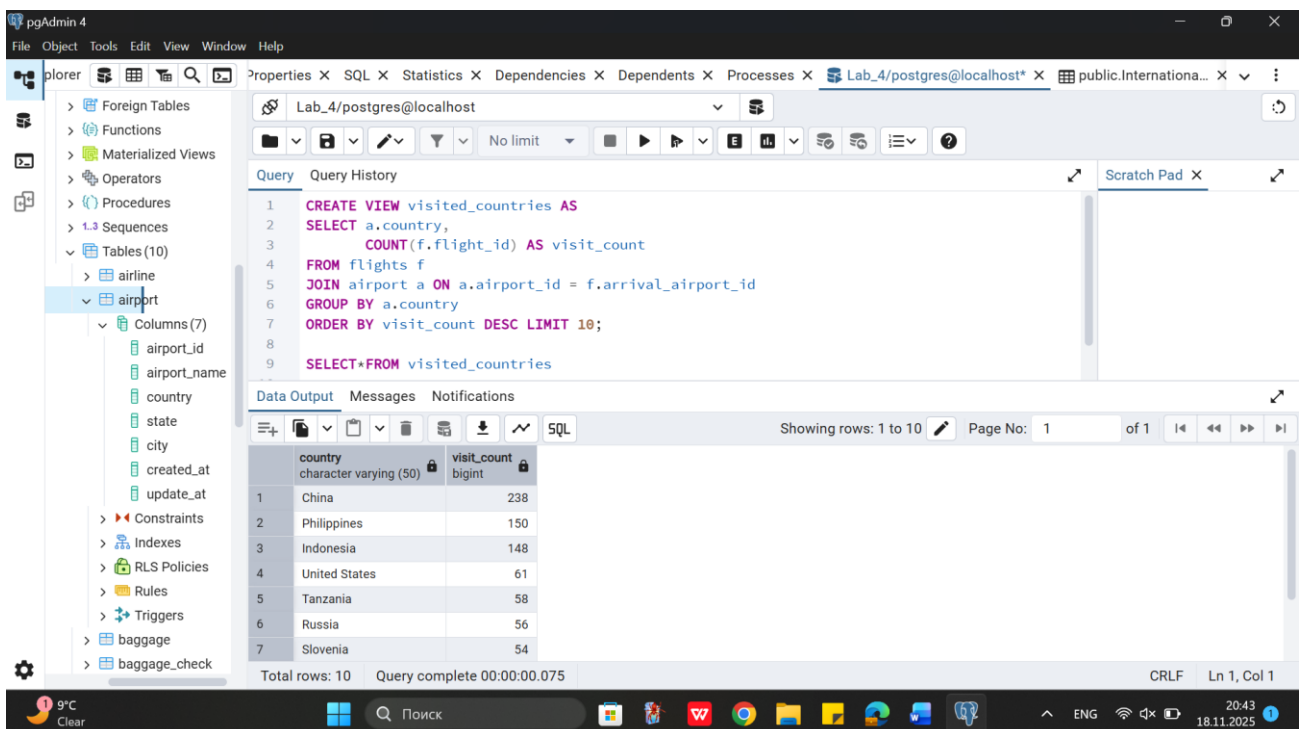
```

The 'Data Output' tab at the bottom shows the results of the query:

booking_id	full_name	country_of_citizenship
15	Shambroke Philbert	Colombia

Total rows: 1. Query complete 00:00:00.072.

8. Create a view that shows top 10 most visited countries.



The screenshot shows the pgAdmin 4 interface. On the left, the 'airport' table is expanded, showing columns like 'airport_id', 'airport_name', 'country', 'state', 'city', 'created_at', and 'update_at'. The 'country' column is selected. In the center, the SQL editor shows the following query:

```

1 CREATE VIEW visited_countries AS
2 SELECT a.country,
3        COUNT(f.flight_id) AS visit_count
4 FROM flights f
5 JOIN airport a ON a.airport_id = f.arrival_airport_id
6 GROUP BY a.country
7 ORDER BY visit_count DESC LIMIT 10;
8
9 SELECT*FROM visited_countries

```

The 'Data Output' tab at the bottom shows the results of the query:

country	visit_count
China	238
Philippines	150
Indonesia	148
United States	61
Tanzania	58
Russia	56
Slovenia	54

Total rows: 10. Query complete 00:00:00.075.

9. Update any of the created views by adding new information in the view table.
Show results.

The screenshot shows the pgAdmin 4 interface. On the left, the 'Columns' tab for the 'flights' table is selected, showing columns like 'flight_id', 'flight_no', 'scheduled_departure', etc. The main query editor contains the following SQL code:

```
1 CREATE OR REPLACE VIEW visited_countries AS
2 SELECT a.country,
3        COUNT(f.flight_id) AS visit_count,
4        COUNT(b.booking_id) AS visit_count_bo
5 FROM flights f
6 JOIN airport a ON a.airport_id = f.arrival_airport_id
7 JOIN booking_flight b ON b.flight_id = f.flight_id
8 GROUP BY a.country
9 ORDER BY visit_count DESC LIMIT 10;
10
11 SELECT*FROM visited_countries
```

The 'Data Output' tab shows the results of the query:

	country character varying (50)	visit_count bigint	visit_count_bo bigint
1	China	232	232
2	Indonesia	167	167
3	Philippines	137	137
4	Tanzania	78	78
5	United States	62	62

Total rows: 10 Query complete 00:00:00.095

10. Drop all existing views.

The screenshot shows the pgAdmin 4 interface. The main query editor contains the following SQL code:

```
1 DROP VIEW IF EXISTS
2 visited_countries,
3 show_platform,
4 show_24,
5 show_airline,
6 specific_airline,
7 popular_routes,
8 show_booking,
9 all_view
10 CASCADE;
```

The 'Messages' tab shows the result of the query:

DROP VIEW

Query returned successfully in 80 msec.

Total rows: Query complete 00:00:00.080