

Laboratory work 6

1. Write a query that displays all flights of a specific airline.

The screenshot shows the pgAdmin 4 interface. The left sidebar has an 'Object Explorer' tree with nodes like 'Triggers', 'airport', 'baggage', etc., under 'flights'. The main area is a SQL editor with the following query:

```
1 SELECT f.flight_id, f.sch_departure_time, f.sch_arrival_time,
2     f.departing_airport_id, f.arriving_airport_id,
3     f.act_departure_time, f.act_arrival_time
4 FROM flights f
5 JOIN airline a ON f.airline_id = a.airline_id
6 WHERE airline_name = 'FlyHigh';
```

The 'Data Output' tab is selected, showing a table structure with columns: flight_id, sch_departure_time, sch_arrival_time, departing_airport_id, arriving_airport_id, act_departure_time, act_arrival_time, created_at, and updated_at. Below the table, it says 'Total rows: 0 Query complete 00:00:00.082'.

2. Compose a query to obtain a list of all flights with the names of departure airports.

The screenshot shows the pgAdmin 4 interface. The left sidebar has an 'Object Explorer' tree with nodes like 'Rules', 'Triggers', 'baggage', etc., under 'flights'. The main area is a SQL editor with the following query:

```
1 SELECT f.flight_id, f.scheduled_departure,
2     f.departure_airport_id, a.airport_name AS departure_airport_name,
3     f.actual_departure, f.actual_arrival
4 FROM flights f
5 JOIN airport a ON f.departure_airport_id = a.airport_id
```

The 'Data Output' tab is selected, showing a table with columns: flight_id, scheduled_departure, departure_airport_id, departure_airport_name, actual_departure, and actual_arrival. The table contains 1000 rows of flight data with their respective departure airport names. Below the table, it says 'Showing rows: 1 to 1000 Page No: 1 of 1' and 'Total rows: 1000 Query complete 00:00:00.119'.

3. Create a query that finds all airlines that have no flights scheduled for the next month.

```

SELECT a.airline_id, a.airline_name
FROM airline a
LEFT JOIN flights f
    ON a.airline_id = f.airline_id
    AND f.scheduled_departure >= date_trunc('month', now()) + interval '1 month'
    AND f.scheduled_departure < date_trunc('month', now()) + interval '2 month'
WHERE f.flight_id IS NULL;

```

airline_id [PK] integer	airline_name character varying (50)
1	SSL
2	CLY
3	YHB
4	KKL
5	IVA
6	GYA
7	YLP

4. Create a query to display a list of passengers on a specific flight.

```

SELECT p.*
FROM passengers p
JOIN booking b ON p.passenger_id = b.passenger_id
JOIN booking_flight bf ON b.booking_id = bf.booking_id
WHERE bf.flight_id = 15;

```

passenger_id [PK] integer	first_name character varying (50)	last_name character varying (50)	date_of_birth date	gender character varying (50)	country_of_citizenship character varying (50)	country_of_residence character varying (50)
1	Maryellen	MacGilmartin	1975-02-25	Female	Finland	China

5. Write a query that calculates the average, total, maximum and minimum price of tickets for each flight

```

SELECT bf.flight_id,
       AVG(b.price) AS avg_price,
       SUM(b.price) AS total_price,
       MAX(b.price) AS max_price,
       MIN(b.price) AS min_price
  FROM booking_flight bf
 JOIN booking b ON bf.booking_id = b.booking_id
 GROUP BY bf.flight_id;

```

flight_id	avg_price	total_price	max_price	min_price
1	846	7070.450000000000	7070.45	7070.45
2	790	7988.950000000000	9851.86	6126.04
3	828	5974.540000000000	5974.54	5974.54
4	938	4635.860000000000	4635.86	4635.86
5	184	2856.200000000000	5712.40	3529.26
6	753	3153.710000000000	3153.71	3153.71
7	273	8929.200000000000	8929.20	8929.20

6. Create a query that shows all flights flying to a specific country by combining flights, airports and airline, and using the condition on the country name.

```

SELECT f.flight_id, f.flight_no, a.airline_name, arr.airport_name
  FROM flights f
  JOIN airline a ON f.airline_id = a.airline_id
  JOIN airport arr ON f.arrival_airport_id = arr.airport_id
 WHERE arr.country = 'Philippines';

```

flight_id	flight_no	airline_name	airport_name
1	US-CT	GYA	Ocean Falls Seaplane Base
2	AU-TAS	SOZ	Darchula Airport
3	ML-2	DUC	Ocean Falls Seaplane Base
4	US-OR	IVA	Industrial Airpark
5	US-CA	CLY	Ocean Falls Seaplane Base
6	AU-NT	IPC	Industrial Airpark
7	TZ-18	BLD	Darchula Airport
8	US-OH	HOU	Industrial Airpark
9	CY-02	NQX	Darchula Airport
10	GY-NF	NGL	Industrial Airpark

7. Display a list of minor passengers and their arrival destination.

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with the **passenger** table selected. The table has 10 columns: passenger_id, first_name, last_name, date_of_birth, gender, country_of_citizenship, country_of_residence, passport_number, created_at, and update_at.
- Query Editor:** Contains the following SQL query:

```
1 SELECT p.passenger_id,
2        p.first_name || ' ' || p.last_name AS full_name,
3        arr.airport_name AS arrival_airport
4 FROM passengers p
5 JOIN booking b ON p.passenger_id = b.passenger_id
6 JOIN booking_flight bf ON b.booking_id = bf.booking_id
7 JOIN flights f ON bf.flight_id = f.flight_id
8 JOIN airport arr ON f.arrival_airport_id = arr.airport_id
9 WHERE EXTRACT(YEAR FROM AGE(now(), p.date_of_birth)) < 18;
```
- Data Output:** Displays the results of the query in a table format. The table has three columns: passenger_id, full_name, and arrival_airport. The results show 37 rows of data.
- System Tray:** Shows battery status (AW01 -1.00%), system icons, and a timestamp (05.11.2025 00:04).

8. Display the passenger's full name, passport number, and the passenger's current time of arrival at the destination.

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with the **flights** table selected. The table has 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.
- Query Editor:** Contains the following SQL query:

```
1 SELECT p.passport_number,
2        p.first_name || ' ' || p.last_name AS full_name,
3        f.actual_arrival
4 FROM passengers p
5 JOIN booking b ON p.passenger_id = b.passenger_id
6 JOIN booking_flight bf ON b.booking_id = bf.booking_id
7 JOIN flights f ON bf.flight_id = f.flight_id
```
- Data Output:** Displays the results of the query in a table format. The table has three columns: passport_number, full_name, and actual_arrival. The results show 1000 rows of data.
- System Tray:** Shows weather (4°C Cloudy), system icons, and a timestamp (05.11.2025 00:14).

9. Print a list of flights where the airline's home country and origin country are the same. Group them by the airport country.

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer pane, which lists various database objects like Procedures, Sequences, Tables (including airline and airport), Constraints, Indexes, RLS Policies, Rules, Triggers, baggage, baggage_check, boarding_pass, and booking. The booking object is expanded to show its columns: booking_id, city, created_at, state, update_at, and airport_id. The main window contains a SQL tab with the following query:

```
1 SELECT airp.country AS origin_country,
2        COUNT(*) AS flights_count
3 FROM flights f
4 JOIN airline a ON f.airline_id = a.airline_id
5 JOIN airport airp ON f.departure_airport_id = airp.airport_id
6 WHERE a.airline_country = airp.country
7 GROUP BY airp.country;
8
```

Below the query is a Data Output tab showing the results:

origin_country	flights_count
Indonesia	6
Slovenia	2
Greece	1
Russia	5
China	50
Brazil	5
Poland	2

Total rows: 8 Query complete 00:00:00.131

At the bottom right, there are system status icons for battery level (2%, 8°C), network, and date/time (00:32, 05.11.2025).