

## Laboratory work 8

### VIEW.

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 tree displays a database structure with tables like 'airport', 'flights', and 'passengers'. The 'flights' table is selected, showing its 14 columns. In the center, the Query Editor window contains the following SQL code:

```
CREATE VIEW date_flight AS
SELECT * FROM flights
WHERE DATE(scheduled_departure) = '2025-01-01';
```

Below the query, the Messages tab shows the result of the execution:

CREATE VIEW  
Query returned successfully in 115 msec.

Total rows: 0 Query complete 00:00:00.115

A green message bar at the bottom right indicates: ✓ Query returned successfully in 115 msec. LF Ln 5, Col 1

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

Object Explorer

```

CREATE VIEW bookings_next_week AS
SELECT b.booking_id, b.created_at, f.flight_no, f.scheduled_departure
FROM booking b
JOIN booking_flight bf ON b.booking_id = bf.booking_id
JOIN flights f ON bf.flight_id = f.flight_id
WHERE EXTRACT(YEAR FROM f.scheduled_departure) = EXTRACT(YEAR FROM CURRENT_DATE)
    AND EXTRACT(WEEK FROM f.scheduled_departure) = EXTRACT(WEEK FROM CURRENT_DATE) + 1;

```

Total rows: Query complete 00:00:00.056

✓ Query returned successfully in 56 msec. LF Ln 2, Col 49

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

```

CREATE VIEW top5_routes AS
SELECT f.departure_airport_id, f.arrival_airport_id,
COUNT(*) AS total_bookings
FROM flights f
JOIN booking_flight bf ON f.flight_id = bf.flight_id
GROUP BY f.departure_airport_id, f.arrival_airport_id
ORDER BY total_bookings DESC
LIMIT 5;

```

Total rows: Query complete 00:00:00.095

✓ Query returned successfully in 95 msec. LF Ln 3, Col 1

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

The screenshot shows the pgAdmin 4 interface with the Object Explorer on the left and a query editor on the right.

**Object Explorer:**

- Expanded table: `flights`
  - Columns (14): `flight_id`, `flight_no`, `scheduled_departure`, `scheduled_arrival`, `departure_airport_id`, `arrival_airport_id`, `departing_gate`, `arriving_gate`, ...
- Expanded table: `booking`
  - Columns (7): `booking_id`, `passenger_id`, `booking_platform`, `created_at`, `update_at`, `status`, `price`
- Expanded table: `flights`
  - Columns (14): `flight_id`, `flight_no`, `scheduled_departure`, `scheduled_arrival`, `departure_airport_id`, `arrival_airport_id`, `departing_gate`, `arriving_gate`, ...

**Query Editor:**

```
CREATE VIEW airline_flights AS SELECT * FROM flights WHERE airline_id = 1;
```

Messages tab (bottom right):

- Total rows: Query complete 00:00:00.056
- LF Ln 1, Col 32
- ✓ Query returned successfully in 56 msec. X

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

The screenshot shows the pgAdmin 4 interface with the Object Explorer on the left and a query editor on the right.

**Object Explorer:**

- Expanded table: `flights`
  - Columns (14): `flight_id`, `flight_no`, `scheduled_departure`, `scheduled_arrival`, `departure_airport_id`, `arrival_airport_id`, `departing_gate`, `arriving_gate`, ...

**Query Editor:**

```
CREATE OR REPLACE VIEW airline_flights AS
SELECT * FROM flights WHERE airline_id = 1
AND scheduled_departure BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days';
```

Messages tab (bottom right):

- Total rows: Query complete 00:00:00.043
- LF Ln 3, Col 83
- ✓ Query returned successfully in 43 msec. X

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

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer pane displays a tree structure of database objects. In the center, the main pane shows a query editor with the following SQL code:

```
1 CREATE VIEW delayed_24h AS
2 SELECT flight_id, flight_no, scheduled_departure, actual_departure, actual_departure - scheduled_departure AS delay
3 FROM flights
4 WHERE actual_departure IS NOT NULL AND actual_departure > scheduled_departure + INTERVAL '24 hours';
```

The 'Messages' tab in the query editor shows the message "CREATE VIEW". Below the query editor, the status bar indicates "Query returned successfully in 50 msec." and "Total rows: 0". A green notification bar at the bottom right says "Query returned successfully in 50 msec. LF Ln 4, Col 1".

## 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 interface with the Object Explorer on the left and the Query tool on the right.

**Object Explorer:**

- Expanded tree under **bookings**:
  - Columns (7)**: airport\_id, airport\_name, country, state, city, created\_at, update\_at
  - Constraints**
  - Indexes**
  - RLS Policies**
  - Rules**
  - Triggers**
  - baggage**
  - baggage\_check**
  - boarding\_pass**
  - booking** (selected):
    - Columns (7)**: booking\_id, passenger\_id, booking\_platform, created\_at, update\_at, status, price
    - Constraints**
    - Indexes**
    - RLS Policies**
    - Rules**
    - Triggers**
    - booking\_flight**
    - flights** (selected):
      - Columns (14)**: flight\_id, flight\_no, scheduled\_departure, scheduled\_arrival, departure\_airport\_id, arrival\_airport\_id, departing\_gate, arriving\_gate, ...

**Query Editor:**

```

CREATE VIEW lt_passengers AS
SELECT DISTINCT p.first_name || ' ' || p.last_name AS full_name, p.country_of_citizenship AS country
FROM passengers p
JOIN booking b ON p.passenger_id = b.passenger_id
WHERE b.booking_platform = 'Leffler-Thompson';

```

**Data Output:**

CREATE VIEW

Query returned successfully in 52 msec.

Total rows: 0 Query complete 00:00:00.052

LF Ln 5, Col 11

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

The screenshot shows the pgAdmin interface with the Object Explorer on the left and the Query tool on the right.

**Object Explorer:**

- Expanded tree under **bookings**:
  - Columns (7)**: airport\_id, airport\_name, country, state, city, created\_at, update\_at
  - Constraints**
  - Indexes**
  - RLS Policies**
  - Rules**
  - Triggers**
  - baggage**
  - baggage\_check**
  - boarding\_pass**
  - booking** (selected):
    - Columns (7)**: booking\_id, passenger\_id, booking\_platform, created\_at, update\_at, status, price
    - Constraints**
    - Indexes**
    - RLS Policies**
    - Rules**
    - Triggers**
    - booking\_flight**
    - flights** (selected):
      - Columns (14)**: flight\_id, flight\_no, scheduled\_departure, scheduled\_arrival, departure\_airport\_id, arrival\_airport\_id, departing\_gate, arriving\_gate, ...

**Query Editor:**

```

CREATE VIEW top10_countries AS
SELECT a.country, COUNT(*) AS visits
FROM flights f
JOIN airport a ON f.arrival_airport_id = a.airport_id
GROUP BY a.country ORDER BY visits DESC
LIMIT 10;

```

**Data Output:**

CREATE VIEW

Query returned successfully in 55 msec.

Total rows: 0 Query complete 00:00:00.055

LF Ln 5, Col 20

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

Object Explorer

```

CREATE OR REPLACE VIEW airline_flights AS
SELECT f.flight_id, f.flight_no, f.airline_id, f.scheduled_departure, f.departure_airport_id, f.arrival_airport_id,
da.airport_name AS departure_airport_name, aa.airport_name AS arrival_airport_name
FROM flights f
JOIN airport da ON f.departure_airport_id = da.airport_id
JOIN airport aa ON f.arrival_airport_id = aa.airport_id;

```

Total rows: Query complete 00:00:00.097

✓ Query returned successfully in 97 msec. X

LF Ln 3, Col 44

Object Explorer

```

SELECT * FROM airline_flights;

```

Data Output Messages Notifications

Showing rows: 1 to 993 Page No: 1 of 1

flight_id	flight_no	airline_id	scheduled_departure	departure_airport_id	arrival_airport_id	departure_airport_name	arrival_airport_name
1	US-CT	39	2024-01-22	12	15	Elorza Airport	Ocean Falls Seaplane Base
2	US-NM	34	2023-07-21	13	16	Figari Sud-Corse Airport	Zephyrhills Municipal Airport
3	FI-OL	34	2023-03-29	18	12	Darchula Airport	Elorza Airport
4	RU-KR	33	2024-01-02	3	19	Lime Acres Finsch Mine Airport	Longana Airport
5	RO-DJ	14	2023-07-03	6	2	Hana Airport	Alert Bay Airport
6	CA-SK	34	2023-07-07	18	1	Darchula Airport	Akunaq Heliport
7	AU-TAS	10	2023-10-12	15	18	Ocean Falls Seaplane Base	Darchula Airport
8	US-AZ	12	2023-07-29	13	6	Figari Sud-Corse Airport	Hana Airport
9	IN-OR	13	2023-05-18	6	9	Hana Airport	Pitalito Airport
10	AU-NT	29	2023-11-25	19	16	Longana Airport	Zephyrhills Municipal Airport
11	TH-57	46	2023-03-28	5	5	Delta County Airport	Delta County Airport
12	CA-NL	27	2023-03-22	12	13	Elorza Airport	Figari Sud-Corse Airport
13	BR-PE	1	2024-01-16	13	20	Figari Sud-Corse Airport	Mellila Airport
14	US-IN	8	2023-07-24	18			
15	MI-9	16	2023-11-07	1			

Total rows: 993 Query complete 00:00:00.084

✓ Successfully run. Total query runtime: 84 msec. 993 rows affected. X

LF Ln 1, Col 31

10. Drop all existing views.

Object Explorer

Object Explorer

Columns (7)

- airport\_id
- airport\_name
- country
- state
- city
- created\_at
- update\_at

Constraints

Indexes

RLS Policies

Rules

Triggers

baggage

baggage\_check

boarding\_pass

booking

Columns (7)

- booking\_id
- passenger\_id
- booking\_platform
- created\_at
- update\_at
- status
- price

Constraints

Indexes

RLS Policies

Rules

Triggers

booking\_flight

flights

Columns (14)

- flight\_id
- flight\_no
- scheduled\_departure
- scheduled\_arrival
- departure\_airport\_id
- arrival\_airport\_id
- departing\_gate
- arriving\_gate
- ...

Query History

```
1 DROP VIEW IF EXISTS date_flight;
2 DROP VIEW IF EXISTS bookings_next_week;
3 DROP VIEW IF EXISTS top5_routes;
4 DROP VIEW IF EXISTS airline_flights;
5 DROP VIEW IF EXISTS delayed_24h;
6 DROP VIEW IF EXISTS lt_passengers;
7 DROP VIEW IF EXISTS top10_countries;
```

Data Output

Messages

Notifications

DROP VIEW

Query returned successfully in 54 msec.

Total rows: Query complete 00:00:00.054

✓ Query returned successfully in 54 msec. X

LF Ln 7, Col 37

The screenshot shows a PostgreSQL database interface. On the left, the Object Explorer pane lists various database objects: 'booking' (with 7 columns: airport\_id, airport\_name, country, state, city, created\_at, update\_at), 'flights' (with 14 columns: flight\_id, flight\_no, scheduled\_departure, scheduled\_arrival, departure\_airport\_id, arrival\_airport\_id, departing\_gate, arriving\_gate, etc.), and other tables like 'baggage', 'baggage\_check', 'boarding\_pass', and 'booking\_platform'. The 'booking' table is currently selected. On the right, the main area contains a query editor with the following SQL code:

```
1 DROP VIEW IF EXISTS date_flight;
2 DROP VIEW IF EXISTS bookings_next_week;
3 DROP VIEW IF EXISTS top5_routes;
4 DROP VIEW IF EXISTS airline_flights;
5 DROP VIEW IF EXISTS delayed_24h;
6 DROP VIEW IF EXISTS lt_passengers;
7 DROP VIEW IF EXISTS top10_countries;
```

Below the query editor, the 'Data Output' tab is active, showing the message "DROP VIEW" and the note "Query returned successfully in 54 msec.". At the bottom of the interface, status information includes "Total rows: Query complete 00:00:00.054" and a green notification bar stating "✓ Query returned successfully in 54 msec. X".