

## Laboratory work 8

### VIEW.

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

The screenshot shows a PostgreSQL IDE interface. On the left, a tree view displays the database schema, including tables like 'airport', 'baggage', 'boarding', 'booking', 'flights', and 'passeng'. The 'flights' table is selected, and its columns are visible: flight\_id, flight\_no, scheduled\_departure, scheduled\_arrival, departure\_airport\_id, arrival\_airport\_id, airline\_id, status, actual\_departure, actual\_arrival, created\_at, and update\_at. The main query editor displays the following SQL code:

```
1 CREATE VIEW flights_by_date AS
2 SELECT flight_id,
3        flight_no,
4        scheduled_departure,
5        scheduled_arrival,
6        departure_airport_id,
7        arrival_airport_id,
8        airline_id,
9        status,
10       actual_departure,
11       actual_arrival,
12       created_at,
13       update_at
14 FROM flights;
```

Below the query editor, the 'Data Output' tab shows the message: 'CREATE VIEW' and 'Query returned successfully in 54 msec.' A status bar at the bottom indicates 'Total rows: Query complete 00:00:00.054' and 'LF Ln 14, Col 14'.

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

The screenshot shows a database management tool interface with a sidebar on the left containing a tree view of database objects. The main window displays a SQL query in the 'Query' tab. The query is as follows:

```

1 CREATE VIEW bookings_next_week AS
2 SELECT bf.booking_id,
3        bf.flight_id,
4        f.scheduled_departure,
5        f.scheduled_arrival,
6        f.departure_airport_id,
7        f.arrival_airport_id
8 FROM booking_flight bf JOIN flights f ON bf.flight_id = f.flight_id
9 WHERE f.scheduled_departure BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '1 week'

```

Below the query editor, the 'Data Output' tab shows the message: 'CREATE VIEW' and 'Query returned successfully in 39 msec.' A status bar at the bottom indicates 'Total rows: Query complete 00:00:00.039'.

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

The screenshot shows the same database management tool interface. The 'Query' tab now displays a different SQL query:

```

1 CREATE VIEW top5_routes AS
2 SELECT
3     departure_airport_id,
4     arrival_airport_id,
5     COUNT(*) AS total_bookings
6 FROM booking_flight bf
7 JOIN flights f ON bf.flight_id = f.flight_id
8 GROUP BY departure_airport_id, arrival_airport_id
9 ORDER BY total_bookings DESC
10 LIMIT 5;

```

The 'Data Output' tab shows the message: 'CREATE VIEW' and 'Query returned successfully in 53 msec.' The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.053'.

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

The screenshot shows the PostgreSQL IDE interface. On the left, the 'Schemas (1)' tree is expanded to 'public', and the 'airline' table is selected. The 'airline' table has columns: airline\_id, airline\_code, airline\_name, airline\_coun, and created\_at. The main query editor contains the following SQL code:

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

The 'Data Output' tab shows the message: 'CREATE VIEW' and 'Query returned successfully in 74 msec.' A green notification box at the bottom right states: '✓ Query returned successfully in 74 msec. X'. The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.074' and 'LF Ln 4, Col 13'.

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 PostgreSQL IDE interface. The 'airline' table is still selected in the left pane. The main query editor contains the following SQL code:

```
1 CREATE VIEW flights_by_airlines AS
2 SELECT f.*, a.airline_name
3 FROM flights f
4 JOIN airline a ON f.airline_id = a.airline_id
5 WHERE a.airline_name = 'IPC'
6 AND f.scheduled_departure BETWEEN CURRENT_DATE
7 AND CURRENT_DATE + INTERVAL '7 days';
```

The 'Data Output' tab shows the message: 'CREATE VIEW' and 'Query returned successfully in 37 msec.' A green notification box at the bottom right states: '✓ Query returned successfully in 37 msec. X'. The status bar at the bottom indicates 'Total rows: Query complete 00:00:00.037' and 'LF Ln 4, Col 13'.

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

The screenshot shows the PostgreSQL FastAPI interface with the 'public' schema selected. The 'airline' table is expanded, showing columns 'airri' and 'crec'. The SQL editor contains the following query:

```
1 CREATE VIEW delayed_24h AS
2 SELECT *
3 FROM flights
4 WHERE actual_departure IS NOT NULL
5       AND actual_departure > scheduled_departure + INTERVAL '24 hours';
```

The 'Data Output' tab shows the message: 'CREATE VIEW' and 'Query returned successfully in 43 msec.' A green status bar at the bottom right confirms: 'Query returned successfully in 43 msec.'

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 PostgreSQL FastAPI interface with the 'public' schema selected. The 'airline' table is expanded, showing columns 'airri' and 'crec'. The SQL editor contains the following query:

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

The 'Data Output' tab shows the message: 'CREATE VIEW' and 'Query returned successfully in 41 msec.' A green status bar at the bottom right confirms: 'Query returned successfully in 41 msec.'

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

The screenshot shows a database management tool interface. The left sidebar displays a tree view of database objects, including 'airport' and 'flights'. The main query editor contains the following SQL code:

```

1 CREATE VIEW top_10_visited_countries AS
2 SELECT
3     a.country AS arrival_country,
4     COUNT(*) AS flights_count,
5     CURRENT_DATE AS created_at,
6     CURRENT_DATE AS update_at
7 FROM flights f
8 JOIN airport a
9     ON f.arrival_airport_id = a.airport_id
10 GROUP BY a.country
11 ORDER BY flights_count DESC
12 LIMIT 10;

```

The 'Data Output' tab shows the message: 'CREATE VIEW' and 'Query returned successfully in 56 msec.' A status bar at the bottom indicates 'Total rows: Query complete 00:00:00.056'.

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

The screenshot shows the same database management tool interface. The left sidebar shows the 'flights' table selected. The main query editor contains the following SQL code:

```

1 CREATE OR REPLACE VIEW flights_ipc_simple AS
2 SELECT
3     flight_id,
4     flight_no,
5     airline_id,
6     scheduled_departure,
7     scheduled_arrival,
8     CURRENT_DATE AS created_at,
9     CURRENT_DATE AS update_at
10 FROM flights
11 WHERE airline_id = 1;

```

The 'Data Output' tab shows the message: 'CREATE VIEW' and 'Query returned successfully in 46 msec.' A status bar at the bottom indicates 'Total rows: Query complete 00:00:00.046'.

- Drop all existing views.

Objec

database\_subj/postgres@Local PostgreSQL

Query

Query History

Scratch Pad

1 DROP VIEW IF EXISTS

2 flights\_by\_date,

3 bookings\_next\_week,

4 top5\_routes,

5 flights\_by\_airline,

6 delayed\_24h,

7 leffler\_passengers,

8 top10\_countries;

Messages

Notifications

NOTICE: view "top10\_countries" does not exist, skipping

DROP VIEW

Query returned successfully in 41 msec.

Total rows: Query complete 00:00:00.041

LF Ln 8, Col 21

upd

Const

airli

unic

Index

RLS P

Rules

Trigge

airport

Column

airp

airp

cou

stat

city

cree

upd

Const

Index

RLS P

Rules

Trigge

baggage

baggage

boarding

booking

booking

flights

Column

fligt

fligt

sch

sch

dep

arri

dep