

TASK 1

The screenshot shows the pgAdmin 4 interface for PostgreSQL 17. The left sidebar is the Object Explorer, displaying a tree structure of database objects. The 'Tables' node under 'flights' is selected, revealing 13 columns: flight_id, sch_departure_time, sch_arrival_time, departing_airport_id, arriving_airport_id, departing_gate, arriving_gate, airline_id, act_departure_time, act_arrival_time, created_at, updated_at, and flight_no. The main pane contains a SQL query window with the following code:

```
CREATE VIEW v_flights AS
SELECT
    flight_id,
    sch_departure_time,
    sch_arrival_time,
    departing_airport_id,
    arriving_airport_id,
    departing_gate,
    arriving_gate,
    airline_id,
    act_departure_time,
    act_arrival_time,
    created_at,
    updated_at,
    flight_no
FROM flights
WHERE DATE(sch_departure_time) = '2024-12-03'
```

The 'Data Output' tab at the bottom shows the message: "CREATE VIEW". Below it, it says "Query returned successfully in 234 msec." The status bar at the bottom indicates the query took 00:00.234 and was run at CRLF, Line 10, Column 13. The system tray at the bottom right shows the date and time as 13.11.2025 17:11.

TASK 2

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Airport/postgres@PostgreSQL 17* public.flights/Airport...

Query History

```
1 CREATE VIEW v_next_week_bookings AS
2   SELECT
3     b.booking_id
4   FROM booking b
5   JOIN flights f ON b.flight_id = f.flight_id
6   WHERE DATE(f.sch_departure_time) BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 day';
```

Scratch Pad

Columns (9)

- booking_id
- flight_id
- passenger_id
- booking_platform
- created_at
- updated_at
- status
- ticket_price
- ticket_discount

Constraints

Indexes

RLS Policies

Rules

Triggers

booking_flight

flights

Columns (13)

- flight_id
- sch_departure_time
- sch_arrival_time
- departing_airport_id
- arriving_airport_id
- departing_gate
- arriving_gate
- airline_id

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 50 msec.

Total rows: Query complete 00:00:00.050 CRLF Ln 6, Col 34

Поиск

17:19 47 KA3 13.11.2025

TASK 3

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

Airport/postgres@PostgreSQL 17* public.flights/Air...

Query History

Query Scratch Pad

CREATE VIEW v_top5_routes AS
SELECT f.departing_airport_id,
f.arriving_airport_id,
COUNT(b.booking_id) AS bookings
FROM booking b
JOIN flights f ON b.flight_id = f.flight_id
GROUP BY f.departing_airport_id, f.arriving_airport_id
ORDER BY bookings DESC
LIMIT 5;

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 62 msec.

Total rows: Query complete 00:00:00.062 CRLF Ln 10, Col 9

Поиск

17:26 47 KA3 13.11.2025

TASK 4

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying a tree structure of database objects. In the 'Tables (10)' section, the 'airline' table is selected, and its columns (airline_id, airline_code, airline_name, airline_country, created_at, updated_at) are visible. The main pane contains a query editor window titled 'Airport/postgres@PostgreSQL 17*' showing the SQL code for creating a view:

```
CREATE VIEW v_airline AS
SELECT
    f.flight_id,
    a.airline_name
FROM flights f
JOIN airline a ON f.airline_id = a.airline_id
WHERE a.airline_name = 'FlyFly_13'
```

Below the query editor is a 'Data Output' tab showing the message: 'CREATE VIEW'. Underneath it, it says 'Query returned successfully in 49 msec.' At the bottom of the pgAdmin window, the status bar shows 'Total rows: 0' and 'Query complete 00:00:00.049'.

The system tray at the bottom of the screen shows various icons for network, battery, and system status, along with the date and time '13.11.2025' and '17:30'.

TASK 5

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying a tree view of database objects. The main area is a query editor with the following SQL code:

```
1 CREATE OR REPLACE VIEW v_airline AS
2 SELECT
3     f.flight_id,
4     a.airline_name
5 FROM flights f
6 JOIN airline a ON f.airline_id = a.airline_id
7 WHERE a.airline_name = 'FlyFly_13'
8     AND DATE(f.sch_departure_time) BETWEEN CURRENT_DATE AND CURRENT_DATE + INTERVAL '7 days'
9
```

The query was executed successfully, as indicated by the messages tab:

CREATE VIEW
Query returned successfully in 54 msec.

At the bottom, the status bar shows "Total rows: 0" and "Query complete 00:00:00.054". The bottom right corner shows the system tray with various icons and the date/time "13.11.2025".

TASK 6

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer pane, which lists various database objects like tables, constraints, indexes, and triggers. In the center is the Query Editor pane, showing a SQL script to create a view named v_delayed_flights. The script selects columns from the flights table where the actual departure time minus the scheduled departure time is greater than 24 hours. Below the Query Editor is the Data Output pane, which displays the successful creation of the view and a message indicating the query ran in 77 msec. The bottom status bar shows the total rows processed and the completion time.

```
CREATE OR REPLACE VIEW v_delayed_flights AS
SELECT
    flight_id,
    departing_airport_id,
    arriving_airport_id,
    departing_gate,
    arriving_gate,
    airline_id,
    created_at,
    updated_at,
    flight_no,
    act_departure_time - sch_departure_time AS delay
FROM flights
WHERE act_departure_time - sch_departure_time > INTERVAL '24 hours';
```

Total rows: Query complete 00:00:00.077 CRLF Ln 15, Col 1

17:38 47 ENG 13.11.2025

TASK 7

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- > RLS Policies
- > Rules
- > Triggers
- > airport
- > baggage
- > baggage_check
- > boarding_pass
- > booking
- > Columns(9)
 - booking_id
 - flight_id
 - passenger_id
 - booking_platform
 - created_at
 - updated_at
 - status
 - ticket_price
 - ticket_discount
- > Constraints
- > Indexes
- > RLS Policies
- > Rules
- > Triggers
- > booking_flight
- > flights
- > Columns(13)
 - flight_id
 - sch_departure_time
 - sch_arrival_time
 - departing_airport_id

Airport/postgres@PostgreSQL 17* public.passengers... public.booking/Air...

Query History

```
1 CREATE VIEW v_passengers AS
2 SELECT
3     p.first_name ||' '|| last_name AS Full_Name,
4     p.country_of_citizenship,
5     b.booking_platform
6 FROM passengers p
7 JOIN booking b ON b.passenger_id = p.passenger_id
```

Scratch Pad

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 60 msec.

Total rows: Query complete 00:00:00.060 CRLF Ln 6, Col 16

Поиск

17:44 48 ENG 13.11.2025

TASK 8

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer Tables (10) Columns (7) Constraints Indexes RLS Policies Rules Triggers baggage baggage_check boarding_pass booking Columns (9) Constraints

Airport/postgres@PostgreSQL 17* public.passengers... public.booking/Air...

Airport/postgres@PostgreSQL 17 No limit E II

Query Query History Scratch Pad

```
1 CREATE VIEW v_top_10_visited_country AS
2 SELECT
3     a.country,
4     COUNT(b.booking_id) AS Number_of_visits
5 FROM flights f
6 JOIN booking b ON b.flight_id = f.flight_id
7 JOIN airport a ON f.arriving_airport_id = a.airport_id
8 GROUP BY a.country
9 ORDER BY Number_of_visits DESC
10 LIMIT 10;
11
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 53 msec.

Total rows: Query complete 00:00:00.053 CRLF Ln 4, Col 17

Поиск 17:52 48 ENG 13.11.2025

TASK 9

The screenshot displays two pgAdmin 4 sessions side-by-side, illustrating a database transformation.

Object Explorer:

- BEFORE Session:** Shows the structure of the 'public.booking' schema, including tables like 'baggage', 'baggage_check', 'boarding_pass', and 'booking'. The 'booking' table contains 9 columns: booking_id, flight_id, passenger_id, booking_platform, created_at, updated_at, status, ticket_price, and ticket_discount.
- AFTER Session:** Shows the same schema structure, but the 'ticket_price' column in the 'booking' table has been renamed to 'ticket_discount'.

Data Output:

- BEFORE Session:** Displays the contents of the 'v_passenger' view. The 'ticket_price' column is present in the results.
- AFTER Session:** Displays the same data, but the 'ticket_price' column is now labeled 'ticket_discount'.

Session Details:

- BEFORE Session:** Total rows: 200, Query complete 00:00:00.135.
- AFTER Session:** Total rows: 200, Query complete 00:00:00.065.

BEFORE

AFTER

TASK 10

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer Airport/postgres@PostgreSQL 17*

Query Scratch Pad

```
1 DROP VIEW IF EXISTS v_flights;
2 DROP VIEW IF EXISTS v_next_week_bookings;
3 DROP VIEW IF EXISTS v_top5_routes;
4 DROP VIEW IF EXISTS v_airline;
5 DROP VIEW IF EXISTS v_delayed_flights;
6 DROP VIEW IF EXISTS v_passengers;
7 DROP VIEW IF EXISTS v_top_10_visited_country;
```

Data Output Messages Notifications

DROP VIEW

Query returned successfully in 64 msec.

Total rows: Query complete 00:00:00.064

CRLF Ln 10, Col 1

18:00 ENG 48 13.11.2025

The screenshot shows the pgAdmin 4 interface. The left sidebar is the Object Explorer, displaying a tree structure of database objects. Under 'Tables (10)', there are ten tables: airline, airport, baggage, baggage_check, boarding_pass, booking, flight, passenger, route, and user. The 'airport' table is expanded, showing its seven columns: airport_id, airport_name, country, state, city, created_at, and updated_at. The 'booking' table is also expanded, showing its nine columns: booking_id, flight_id, passenger_id, booking_platform, booking_source, booking_type, booking_status, booking_date, and booking_time. The top right pane is the 'Query' editor, which contains a multi-line SQL script. The script consists of several 'DROP VIEW IF EXISTS' statements, each targeting a view named after one of the expanded tables. The bottom right pane is the 'Messages' tab, which displays the message 'Query returned successfully in 64 msec.' The bottom status bar shows the total rows processed (0), the query completion time (00:00:00.064), and the current line and column (Ln 10, Col 1). The bottom right corner of the status bar shows the system time (18:00) and date (13.11.2025).