

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

- File** | Object Tools Edit View Window Help
- Two tabs are open: **Welcome** and **postgres/postgres@PostgreSQL 17\***.
- The main area displays a SQL query in the **Query** tab:

```
1 create view daily_dep as
2 select *
3 from flights
4 where date(sch_departure_time)='2025-10-22'
```
- The **Scratch Pad** tab is also visible.
- Data Output** tab is selected, showing the results of the query:
  - CREATE VIEW
  - Query returned successfully in 117 msec.
- Bottom status bar: Total rows: 0, Query complete 00:00:00.117, CRLF, Ln 4, Col 45.

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

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

No limit

Query History

Scratch Pad ×

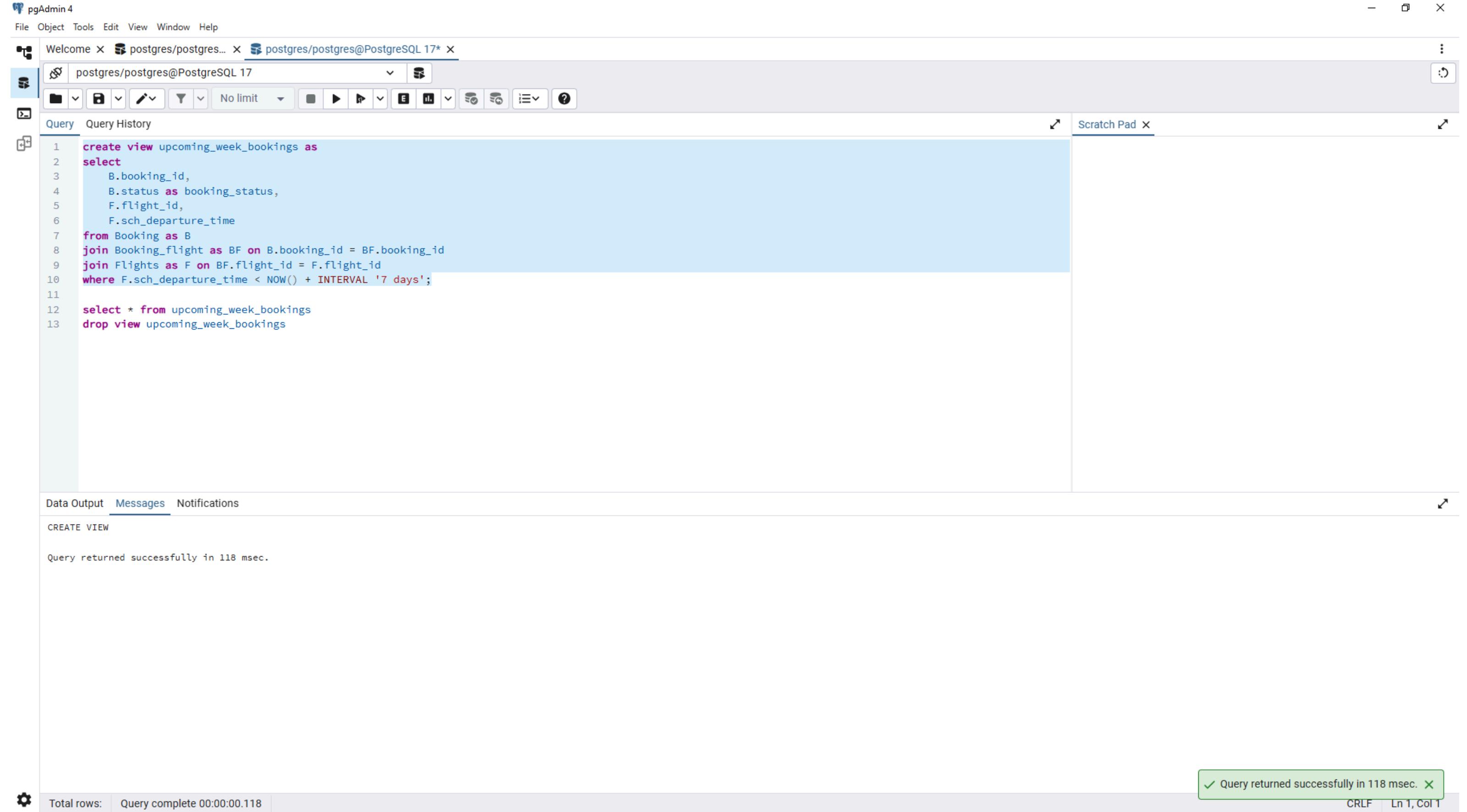
```
1 create view upcoming_week_bookings as
2 select
3     B.booking_id,
4     B.status as booking_status,
5     F.flight_id,
6     F.sch_departure_time
7 from Booking as B
8 join Booking_flight as BF on B.booking_id = BF.booking_id
9 join Flights as F on BF.flight_id = F.flight_id
10 where F.sch_departure_time < NOW() + INTERVAL '7 days';
11
12 select * from upcoming_week_bookings
13 drop view upcoming_week_bookings
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 118 msec.

Total rows: Query complete 00:00:00.118 ✓ CRLF Ln 1, Col 1



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

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

No limit

Query History

Query Scratch Pad

```
1 create view top_5_popular_routes as
2 select
3     F.departing_airport_id,
4     F.arriving_airport_id,
5     count(BF.booking_id) as total_bookings
6 from Booking_flight as BF
7 join Flights as F on BF.flight_id = F.flight_id
8 group by F.departing_airport_id, F.arriving_airport_id
9 order by total_bookings desc
10 limit 5;
11 select * from top_5_popular_routes
12 drop view top_5_popular_routes
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 105 msec.

Total rows: Query complete 00:00:00.105 ✓ CRLF Ln 1, Col 1

The screenshot shows the pgAdmin 4 interface with a query editor window. The query creates a view named 'top\_5\_popular\_routes' that selects flight routes based on the number of bookings. It joins the 'Booking\_flight' and 'Flights' tables, groups by departure and arrival airports, orders by total bookings in descending order, and limits the results to 5. After executing the query, a message in the 'Messages' tab indicates success with a duration of 105 msec. A status bar at the bottom shows the total rows affected and the completion time.

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

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

No limit

Query History

Query Scratch Pad

```
1 create view all_flights_with_airline_name as
2 select
3     F.flight_id,
4     F.sch_departure_time,
5     F.departing_airport_id,
6     F.arriving_airport_id,
7     A.airline_name
8 from Flights as F
9 join Airline as A on F.airline_id = A.airline_id;
10
11 select * from all_flights_with_airline_name
12 drop view all_flights_with_airline_name
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 97 msec.

Total rows: Query complete 00:00:00.097 ✓ CRLF Ln 1, Col 1

The screenshot shows the pgAdmin 4 interface with a query editor window. The query creates a view named 'all\_flights\_with\_airline\_name' that joins the 'Flights' and 'Airline' tables based on their 'airline\_id'. It then selects all columns from the joined table. Finally, it drops the view. The 'Messages' tab shows a success message: 'CREATE VIEW' and 'Query returned successfully in 97 msec.'. A green status bar at the bottom right indicates the query was completed successfully in 0.097 seconds. The top menu bar includes File, Object, Tools, Edit, View, Window, and Help.

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

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

No limit

Query History

Query Scratch Pad

```
1 create or replace view specific_airline_upcoming_flights as
2 select
3     F.flight_id,
4     F.sch_departure_time,
5     F.sch_arrival_time,
6     A.airline_name
7 from Flights as F
8 join Airline as A on F.airline_id = A.airline_id
9 where A.airline_name = 'Air_afd85e' and F.sch_departure_time < NOW() + INTERVAL '7 days';
10
11 select * from specific_airline_upcoming_flights
12 drop view specific_airline_upcoming_flights
13
14
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 117 msec.

Total rows: Query complete 00:00:00.117 ✓ CRLF Ln 1, Col 1

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

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

No limit

Query History

Scratch Pad ×

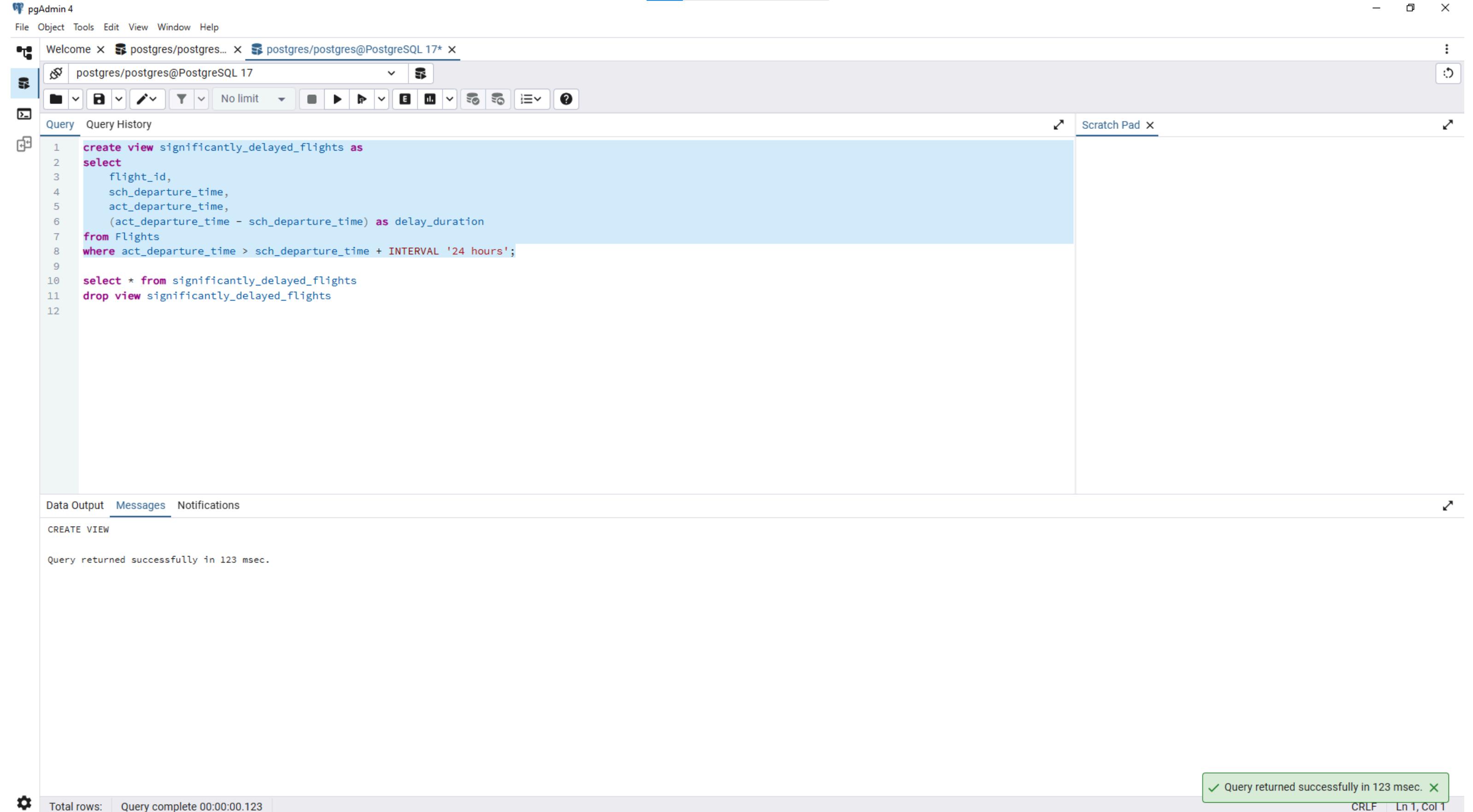
```
1 create view significantly_delayed_flights as
2 select
3     flight_id,
4     sch_departure_time,
5     act_departure_time,
6     (act_departure_time - sch_departure_time) as delay_duration
7 from Flights
8 where act_departure_time > sch_departure_time + INTERVAL '24 hours';
9
10 select * from significantly_delayed_flights
11 drop view significantly_delayed_flights
12
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 123 msec.

Total rows: Query complete 00:00:00.123 ✓ CRLF Ln 1, Col 1



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

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

No limit

Query History

Scratch Pad ×

```
1 create view leffler_thompson_passengers as
2 select
3     P.passenger_id,
4     P.first_name || ' ' || P.last_name as full_name,
5     P.country_of_citizenship
6 from Passengers as P
7 join Booking as B on P.passenger_id = B.passenger_id
8 where B.booking_platform = 'Platform_108377'
9
10 select * from leffler_thompson_passengers
11 drop view leffler_thompson_passengers
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 148 msec.

Total rows: Query complete 00:00:00.148

✓ Query returned successfully in 148 msec. ×

CRLF Ln 1, Col 1

A screenshot of the pgAdmin 4 interface. The main window shows a SQL query being run against a PostgreSQL 17 database. The query creates a view named 'leffler\_thompson\_passengers' that selects passenger IDs, first names, last names, and countries of citizenship from the 'Passengers' table, joining it with the 'Booking' table to filter by the 'Platform\_108377' booking platform. The query is successfully executed, returning 148 msec. The status bar at the bottom indicates the total rows processed and the completion time.

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.

pgAdmin 4

File Object Tools Edit View Window Help

Welcome × postgres/postgres@PostgreSQL 17\* ×

postgres/postgres@PostgreSQL 17

No limit

Query History

Query Scratch Pad

```
1 create view top_10_visited_countries as
2 select
3     A.country,
4     count(F.flight_id) as total_arrivals
5 from Flights as F
6 join Airport as A on F.arriving_airport_id = A.airport_id
7 group by A.country
8 order by total_arrivals desc
9 LIMIT 1;
10 select * from top_10_visited_countries
11 drop view top_10_visited_countries
```

Data Output Messages Notifications

CREATE VIEW

Query returned successfully in 120 msec.

Total rows: Query complete 00:00:00.120 ✓ CRLF Ln 1, Col 1

The screenshot shows the pgAdmin 4 interface with a query editor window. The query creates a view named 'top\_10\_visited\_countries' that selects the country and total arrivals from the 'Flights' and 'Airport' tables, ordered by total arrivals in descending order, with a limit of 1. It then selects all rows from the view and finally drops the view. Below the query editor, the 'Messages' tab is active, displaying the message 'CREATE VIEW' followed by 'Query returned successfully in 120 msec.'. A green status bar at the bottom right indicates the query was completed successfully in 120 msec. The bottom navigation bar shows 'Total rows: Query complete 00:00:00.120' and file paths 'CRLF Ln 1, Col 1'.

Create a view that shows top 10 most visited countries.